Algorithms for Crystallography in the Scanning Electron Microscope

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Department of Materials Science and Engineering

Zachary T. Varley

B.S., Materials Science and Engineering, University of Pennsylvania
M.S., Materials Science and Engineering, University of Pennsylvania

Carnegie Mellon University
Pittsburgh, PA

August, 2024

Abstract

This thesis explores new approaches to enhance the efficiency and accuracy of materials characterization in the scanning electron microscope (SEM) with particular emphasis on electron backscatter diffraction (EBSD). The work addresses two challenges in the field: accelerating data acquisition and improving the handling of orientation data acquired from polycrystals. Firstly, we introduce an unsupervised dynamic sampling approach based on a nearest neighbor heuristic for scanned microscopy modalities. Secondly, we present a variation on dictionary-based orientation indexing for EBSD using principal component analysis (PCA) and numerical quantization which accelerates and improves the noise robustness of the method. Thirdly, we cover two potential tools for multimodal image registration in the SEM using approaches rooted in information theory. Lastly, we develop decision tree algorithms for the efficient reduction of crystallographic orientations to the fundamental zone of their respective Laue classes. By addressing key algorithmic obstacles in SEM-based microstructure characterization, this thesis aims to alleviate some of the challenges of handling such materials science data in this era of big data.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support and guidance of numerous individuals and institutions. First and foremost, I would like to express my deepest gratitude to my advisors, Dr. Marc De Graef and Dr. Gregory Rohrer, for their support, invaluable advice, and continuous encouragement throughout my PhD journey. Their expertise and insight have been instrumental in shaping this work.

I am also immensely grateful to the other members of my dissertation committee, Dr. Anthony Rollett, Dr. Amanda Krause, and Dr. Michael Uchic, for their time, effort, and thoughtful feedback, which have greatly improved the quality of this research.

Stimulating interactions with other researchers Dr. Megna Shah, Dr. Sean Donegan, Dr. Mike Chapman, and Dr. Greg Sparks of the Air Force Research Laboratory in Dayton, Ohio were particularly fruitful.

I extend my sincere thanks to my colleagues and friends in the MSE department at CMU for their support and camaraderie. In particular thanks to Dr. Elena Pascal, Dr. Chaoyi Zhu, Dr. Joseph Tessmer, Dr. Ke-Wei Jin, Dr. Maxwell Li, Dr. Michael Kitcher, Dr. Vincent Monardo, Marcus Ochsendorf, Chenxi Yu, Toby Francis, Harry Dong, Crestienne Dechaine, Hao Zhu, and Zehua Liu for their assistance and for making my time here memorable.

I would like to acknowledge the financial support provided by the Data-Driven Discovery of Optimized Multifunctional Material Systems (D3OM2S) Center of Excellence, Air Force agreement number FA8650-19-2-5209; and by the Materials Characterization Facility at Carnegie Mellon University supported by grant MCF-677785, without which this research would not have been possible.

To my family, thank you for your unwavering love, patience, and encouragement. My

parents have always been my pillars of strength and inspiration. Thank you for believing in me and for your endless support. My three brothers Jacob, Matthew, and Thomas have always given me candid and heartfelt guidance and advice.

Finally, I am profoundly grateful to my wife, Safa, for her love, understanding, and sacrifices.

Zachary Varley
Pittsburgh, PA

List of Publications Resulting from this Work

The following papers related to the work presented in this thesis have been published, submitted, or are in preparation:

1. Zachary Varley, Gregory S. Rohrer, and Marc De Graef. "Dynamic sampling for scanned microscopy modalities." Materials Characterization (2024): 113865.

2. Zachary Varley, Gregory S. Rohrer, and Marc De Graef. "Accelerating Dictionary Indexing of Electron Backscatter Diffraction Patterns with PCA and Quantization" [Submitted]

3. Zachary Varley, Gregory S. Rohrer, and Marc De Graef. "On Fitting Detector Pose in EBSD" [in preparation]

TABLE OF CONTENTS

Abstract . . . . **i**

Acknowledgements . . . . **i**

List of Publications Resulting from this Work . . . . **iv**

List of Tables . . . . **viii**

List of Figures . . . . **ix**

List of Acronyms . . . . **xiii**

List of Symbols . . . . **xv**

Glossary . . . . **xvii**

I. Introduction 1
1.1 Dynamic Sampling and Compressed Sensing 2
1.2 Multimodal Image Registration 3
1.3 Fundamental Zone Reductions 3
1.4 Hypotheses 4
1.5 Thesis Outline 4
II. Background 6
2.1 Orientation and Texture 6
2.1.1 SO(3) 6
2.1.2 Orientation Representations 7
2.1.3 Uniform Sampling of Orientation Space 13
2.2 Electron Back-scatter Diffraction in the SEM 14
2.2.1 Experimental Geometry and Setup 15
2.2.2 Electron Diffraction Forward Model 16
2.2.3 Automation of EBSD Indexing 17
2.3 EBSD Orientation Determination 18
2.3.1 Hough Transform Indexing 19
2.3.2 Dictionary Indexing 20
2.3.3 Spherical Harmonic Indexing 21
2.4 More on the Homochoric Inverse 23
III. Dynamic Sampling of Scanned Modalities 28
3.1 Introduction 28
3.2 Sampling Methodology and Test Datasets 29

3.2.1 Motivation 29
3.2.2 Scoring and Selection 30
3.2.3 Implementation 31
3.2.4 Image Completion 33
3.2.5 Performance Evaluation 34
3.2.6 Test Datasets 34
3.2.7 Image Complexity 35
3.2.8 Microstructure Comparison 35
3.3 Results and Discussion 36
3.3.1 Example Samplings across Datasets 36
3.3.2 Comparison with SLADS 39
3.3.3 User Parameter Dependency 44
3.3.4 Synthetic Runtime Results 46
3.3.5 Sampling Performance vs. Image Complexity 47
3.4 Conclusion and Summary 48
IV. PCA Dictionary Indexing 50
4.1 Introduction 50
4.2 Methods 52
4.2.1 PCA Calculation 52
4.2.2 K-Nearest Neighbors and Dynamic Numerical Quantization 54
4.3 Test Dataset 55
4.4 Results and Discussion 56
4.4.1 PCA Components 56
4.4.2 Pattern Matching Speed and Hardware Utilization 57
4.4.3 Disorientation Distributions 59
4.5 Conclusion 67
4.6 Data Availability 67
V. Multimodal SEM Image Registration Tools 68
5.1 Difference of Local Shannon Entropy Key Points 68
5.1.1 Rotation and Scaling Invariance 68
5.1.2 Registration Results 70
5.2 Edgeworth Cross Mutual Information Function 73
5.2.1 Theory 74
5.2.2 Preliminary Results 80
VI. Decision Tree Reductions for Laue Classes 82
6.1 Introduction and Motivation 82
6.2 Theory and Background 82
6.2.1 Generator Adjacency 83
6.2.2 Infeasible Generators 83
6.3 Computational Methods 91
6.3.1 Cyclic Groups: C2, C3, C4, C6 92
6.3.2 Dihedral Groups: D2, D3, D4, D6 94
6.3.3 Groups O and T 102
6.4 Reduction Benchmarks 107
VII. Conclusions and Future Work 108
7.1 Summary 108
7.2 Hypotheses Revisited 109
7.3 Future Outlook 110

A. Lie Algebras and Lie Groups 113
B. Polynomial Fits for the Homochoric Inverse 114
C. Laue Group Generator Adjacency Graphs 118

Bibliography **124**

LIST OF TABLES

2.1 Maximum exactly representable scalars, rounding thresholds to unity, and corresponding minimum representable angles for floating point quaternions. 10

3.1 Average compute time per pixel in μs (10 trials) across resolutions for random images. The devices used for this benchmark were a Nvidia T4 GPU and 4 Intel Skylake CPU cores. 33

4.1 PPS and TOPS on 8 CPU cores using INT8 quantization. 58
4.2 PPS and TOPS on a Nvidia T4 using FP16 precision. 59
4.3 PPS and TOPS on a Nvidia T4 using FP32 precision. 59
4.4 Confusion matrices for each dictionary size of the 1500 component PCA and Full resolution DI across dictionary size. 67

6.1 Generators for Laue group D6 with the zone infeasible under FZ-FZ composition bolded. 85
6.2 Laue group D6 generators that touch the RFZ, and their S^3 bounds with the identity. 86
6.3 Generators for Laue group O with zones infeasible under FZ-FZ composition bolded. 90
6.4 The 14 Laue group O generators that touch the RFZ, and their combined S^3 bounds with the identity. 91

B.1 Polynomial Fits for EBSDtorch 116
B.2 Mean and Max Absolute Errors for Various Polyfits for the Homochoric inverse using FP64 and FP32 Precision 117

C.1 Adjacency matrix for Laue group D4. 119
C.2 Adjacency matrix for Laue group D6. 120
C.3 Adjacency matrix for Laue group T. 121
C.4 Adjacency matrix for Laue group O. 122

LIST OF FIGURES

2.1 ZXZ Bunge angles rotation of reference frame (adapted from Figure 2.3 of [1]) 8

2.2 First figure of [2] showing exploded view of (a) cube and (b) ball subdivisions in 3D. 11

2.3 Typical EBSD geometry (reproduced from [3] - CC 3.0 License.) 15

2.4 Image reproduced from [4] showing the first ever EBSD pattern (adaptive histogram equalization applied to show more details). 18

2.5 Figure reproduced from [5] compares indexing methods at increasing noise levels: ”From left to right are patterns, indexing results of Hough, DI, spherical indexing (SI) with 87 bands, and EBSD-CNN.” 19

2.6 The Hough and Radon transform convert lines to points which facilitates calculating angles between diffraction bands. The vertical axis is the distance of the line from the center of the pattern and the angle tells the orientation of the line. 20

2.7 (left) original homochoric conversion and (right) the modified homochoric equation 24

2.8 Uniformly spaced input angles are converted to homochoric moduli and then inverted and the absolute error in the inversion in radians (logscale) is shown for both 32-bit and 64-bit floating numbers 25

2.9 In 32-bit precision, uniformly sampled (Shoemake approach) scaled axis-angle vectors are converted to homochoric vectors and then the angle is estimated from various methods. 25

2.10 In 64-bit precision, uniformly sampled (Shoemake approach) scaled axis-angle vectors are converted to homochoric vectors and then the angle is estimated from various methods. 26

2.11 Residuals of Chebyshev polynomial fits to inverse of equation 2.12 and modified inverse for both single and double precision. 27

3.1 (top left) binary map indicating disorientation angle over 2° (top right) sampling locations (bot left) ground truth (bot right) nearest neighbor imputed orientation map (O_h IPF coloring) 37

3.2 (top right) binary map indicating sampling locations (top left) raw map of the absolute difference between (bot left) ground truth and (bot right) imputed micrograph 38

3.3 (top right) binary map indicating sampling locations (top left) raw map of the absolute difference between (bot left) ground truth and (bot right) imputed micrograph 39

3.4 For a candidate acceptance factors of (a) 0.01 and (b) 0.1 the resultant sampling masks for 10%, 20%, and 40% when using UDS. 40
3.5 Direct comparison of the PSNR for reconstruction of the same example micrograph dubbed "microstructure" in reference [6]. 40
3.6 (a) dimensionless shape number Ω_3 empirical cumulative distribution functions compared over sampling approaches and sampling fraction (b) equivalent sphere diameter log-plots from the sample synthetic sampling experiment. 42
3.7 For uniform binning with 10, 100, and 1000 bins, the Hellinger distances are shown between the Omega3 distributions computed from sampling and the corresponding distribution from the original dataset. 43
3.8 Reconstructions at various sampling percents under the proposed dynamic sampling algorithm of the medial slice in the 3D MIDAS dataset. The ground truth orientation map and masks of the sampling locations are also shown. 43
3.9 Percent of pixels sampled up to 50.0% vs. PSNR plotted across K number of nearest neighbors and weighting. 45
3.10 Percent of pixels sampled up to 50.0% vs PSNR plotted across several user parameters 46
3.11 Across F and W, heatmaps of (a) time to 30% (log-scale legend) (b) PSNR at 30% sampled 47
3.12 Heatmap showing the average image complexity that led to a given PSNR at each percent sampled. Only mean values with more than 3 contributing data-points appear in the heatmap. UHCS and CLSM datasets are combined for both plots. 48

4.1 Example patterns from the (4,4) point on the region of interest in the camera gain dataset. The camera gain (in dB) is over each column. The middle and bottom rows show static and dynamic background subtraction preprocessing steps. 56
4.2 Representative principal components from the projection of the Nickel master pattern onto the virtual detector plane. Lower rows use a larger sampling of the orientation fundamental zone. 57
4.3 Representative principal components from the projection of a faux checkerboard master pattern onto the virtual detector plane. Lower rows use a larger sampling of the orientation fundamental zone. 57
4.4 A grid of disorientation angle ECDFs across indexing mode and dictionary size for Scan 10 alone. Each plot line is colored and styled according to the numeric precision and compute device utilized. 60
4.5 A grid of disorientation angle ECDFs across indexing mode and scan number with each plot line colored according to the size of the dictionary. 61
4.6 A grid of disorientation angle ECDFs across indexing mode and dictionary size with each plot line colored according to the size of the scan number. . 62
4.7 A grid of disorientation angle ECDFs across dictionary size and scan number with each plot line colored according to the indexing mode. 64
4.8 A grid of inverse pole figure (IPF) Z-axis orientation maps across scans and dictionary sizes at a fixed precision (FP32) and indexing method (1500 PCA components) 65

4.9 Comparison on Scan 10 between PCA with 1500 component and Full DI IPF Z-axis color maps. Top row legend: both methods exceed 3° disorientation to reference (black pixels), neither (blue) PCA 1500 only (red) Full DI only (green). 66

5.1 Inliers vs. centered Rotation Angle for IN100 / MIDAS / NIST for (left) disk local entropy DoLE and (right) box local entropy DoLE. The y-axis ticks and grid lines are shared. 69

5.2 Inliers vs. centered Rescaling Factor for IN100 / MIDAS / NIST for (left) disk local entropy DoLE and (right) box local entropy DoLE. The y-axis ticks and grid lines are shared. 70

5.3 (left) serial sectioning example pairs and (right) inlier matched image patches 71

5.4 Green and magenta color channels show the ISE/BSE and EBSD data registration results. Top to bottom rows show IN100 / MIDAS / NIST results respectively. Left to right, the initial unregistered images, DoLE key point registered, and DoLE + MIND refinement registration results are displayed. 72

5.5 DoLE key point registered, and DoLE + MIND refinement registration results on the NIST as-built sample alone. Red arrows highlight refinement corrections. 73

5.6 Cross mutual information surfaces indicating the estimated mutual information (normalized per image) between the example ISEI and EBSD scan with both the existing and new Edgeworth series formulation. Additive Gaussian noise has been applied with the labeled σ for each column. The ideal peak position indicating the correct offset is circled in red, and the actual locations are in blue. 81

6.1 Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the three other Voronoi cells flanking the cube shaped fundamental zone of D2. 95

6.2 Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the five other Voronoi cells flanking the hexagonal prism shaped fundamental zone of D3. 97

6.3 Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the seven other Voronoi cells surrounding the octagonal prism shaped fundamental zone of D4. 99

6.4 Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the eleven other Voronoi cells surrounding the dodecagonal prism shaped fundamental zone of D6. 100

6.5 Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the eleven other Voronoi cells surrounding the octahedron shaped fundamental zone of T. 103

6.6 Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the 23 other Voronoi cells surrounding the truncated cube shaped fundamental zone of O. 104

6.7 Bar plot showing the relative speedup of the tree based method compared to the best of either the angle and bounds methods for a given Laue group. Single hash indicates comparison with the angle method while double hash indicates the bounds method. 107

C.1 Adjacency graph for Laue group D4 (red edges indicate non-adjacent generators). 119
C.2 Adjacency graph for Laue group D6 (red edges indicate non-adjacent generators). 120
C.3 Adjacency graph for Laue group T (red edges indicate non-adjacent generators). 121
C.4 Adjacency graph for Laue group O. 123

LIST OF ACRONYMS

SEM	Scanning Electron Microscope
EBSD	Electron Backscatter Diffraction
EDS	Energy-Dispersive X-ray Spectroscopy
ECCI	Electron Channeling Contrast Imaging
SLADS	Supervised Learning Approach for Dynamic Sampling
K-sphere	Kikuchi Sphere
PSNR	Peak Signal-to-Noise Ratio
CMIF	Cross Mutual Information Function
ECMIF	Edgeworth Cross Mutual Information Function
BSE	Backscatter Electron
RFZ	Rodrigues Fundamental Zone
FZ	Fundamental Zone
PCA	Principal Component Analysis
PC	Pattern Center
HI	Hough Indexing
SI	Spherical Indexing
DI	Dictionary Indexing
GPU	Graphics Processing Unit
FFT	Fast Fourier Transform
ROI	Region of Interest
ERD	Expected Reduction in Distortion
k-NN	k-Nearest Neighbors
CUDA	Compute Unified Device Architecture

IPF	Inverse Pole Figure
UHCS	Ultra-High Carbon Steel
CLSM	Confocal Laser Scanning Microscopy
SE	Secondary Electron
LSM	Laser Scanning Microscope
UDS	Unsupervised Dynamic Sampling
EBSP	Electron Backscatter Pattern
SVD	Singular Value Decomposition
EVD	Eigenvalue Decomposition
LSH	Locality Sensitive Hashing
PQ	Product Quantization
TOPS	Trillions of Operations Per Second
PPS	Patterns Per Second
ECDF	Empirical Cumulative Distribution Function
NLPAR	Non-Local Pattern Averaging and Reindexing

LIST OF SYMBOLS

$\mathbf{SU(2)}$	Special Unitary Group 2
$\mathbf{SO(3)}$	Special Orthogonal Group 3
$\mathbb{S}^2$	2-Sphere
$\mathbb{S}^3$	3-sphere
$\mathbb{S}^3_-$	Southern real hemisphere of $\mathbb{S}^3$
$\mathbb{S}^3_+$	Northern real hemisphere of $\mathbb{S}^3$
$\mathbf{R}$	Rotation matrix
$\mathbf{I}$	Identity matrix
$\mathbf{K}$	Skew-symmetric matrix
$\mathfrak{so}(3)$	Lie algebra of SO(3)
$\mathbf{n}$	Axis of rotation
θ	Angle of rotation
$\varphi_1, \Phi, \varphi_2$	Bunge Euler angles
$\mathbb{H}$	Quaternions
x_0, x_1, x_2, x_3	Quaternion components
h_x, h_y, h_z	Homochoric vector
c_x, c_y, c_z	Cubochoric coordinates
ρ	Rodrigues-Frank vector
Y^ℓ_m	Spherical harmonics
$\hat{f}^\ell_m$	Spherical harmonic coefficients
P^ℓ_m	Associated Legendre polynomials
$D^\ell_{m,n}$	Wigner D-matrix
$d^\ell_{k,m}$	Wigner d-matrix

$\mathbf{X}$	Data matrix
$\mathbf{U}$	Left singular vectors matrix
$\mathbf{V}$	Right singular vectors matrix
$\boldsymbol{\Sigma}$	Singular values diagonal matrix
$\mathbf{C}$	Covariance matrix
$\mathbf{Q}$	Eigenvectors matrix
$\boldsymbol{\Lambda}$	Eigenvalues diagonal matrix
x_i	Dictionary pattern or PCA loading
q_i	Query pattern or PCA loading
p	Number of PCA components
K_O	Space group
$\mathrm{SO}(K_o)$	Admissible finite subgroup of SO(3)
$D(k_i)$	Dirichlet-Voronoi partition
θ_{D6}	Maximum angle of D6 fundamental zone
$\mathcal{L}$	Lagrangian

GLOSSARY

Axis-angle representation A method of describing rotations using a unit vector (axis) and an angle of rotation around that axis.

Backscattered electrons Electrons that are scattered back from a sample in an SEM, used in EBSD.

Bloch-wave ansatz A mathematical approach used in dynamical scattering simulations for EBSD.

Bunge Euler angles A set of three angles (φ_1, Φ, φ_2) used to describe crystallographic orientations.

Cathodoluminescence Light emission from a material when bombarded by electrons, often observed in EBSD of ceramics and semiconductors.

Clifford algebra An algebraic structure generalizing complex numbers and quaternions, relevant in some rotation representations.

Cross Mutual Information Function (CMIF) A technique for computing mutual information between two images over various spatial shifts.

Crystallite (see grain) A small, single crystal within a polycrystalline material.

Cubochoric coordinates A coordinate system that provides a volume-preserving bijection between a cube and a ball, useful for uniform sampling of orientation space.

Dictionary indexing An EBSD indexing method that compares experimental patterns to a pre-computed dictionary of simulated patterns.

Difference of Local Entropy (DoLE) A method for detecting key points in images based on local entropy differences.

Dirichlet-Voronoi partition A partitioning of space based on distance to a specified set of points, relevant in orientation space analysis.

Dynamical scattering A model of electron diffraction that accounts for multiple scattering events.

EBSD (Electron Backscatter Diffraction) A materials characterization technique that provides information about crystal orientations, phases, and defects.

EDS (Energy-Dispersive X-ray Spectroscopy) An analytical technique used in conjunction with EBSD for elemental analysis.

Edgeworth Cross Mutual Information Function (ECMIF) An extension of CMIF using Edgeworth series expansions to estimate entropy.

Edgeworth series A method of representing a probability distribution as a series, used here to approximate entropy.

Euler space The three-dimensional space defined by Euler angles, used to represent orientations.

Euler-Rodrigues formula A mathematical expression relating rotation matrices to axis-angle representation.

Forward model A simulation of the physical process that produces EBSD patterns from a given crystal orientation.

Friedel's law A principle in diffraction stating that the intensity of a diffracted beam is the same for reflections (hkl) and (-h-k-l).

Fundamental zone The smallest region in orientation space that contains all unique orientations for a given symmetry.

Grain A small, single crystal within a polycrystalline material.

Grain Reference Orientation Deviation (GROD) A measure of local misorientation within a grain relative to its average orientation.

Homochoric inverse The process of converting from homochoric coordinates back to axis-angle representation.

Homochoric representation An equal-volume mapping from the hemisphere of the 3-sphere to a 3D ball, used in orientation analysis.

Homography A transformation used in image processing to map points between two planes.

Hough transform An image processing technique used to detect lines, applied in EBSD for identifying Kikuchi bands.

Ion-induced Secondary Electron (ISE) Electrons emitted from a sample when bombarded with ions, used in some microscopy techniques.

Jacobi polynomials A class of orthogonal polynomials used in the calculation of Wigner d-matrices.

K-sphere (Kikuchi sphere) A reference signal on the 2-sphere used in EBSD simulations and analysis.

Kernel Average Misorientation (KAM) A measure of local misorientation between a central point and its neighbors.

Kepler's equation An equation in orbital mechanics, related to the homochoric inverse problem in orientation analysis.

Kikuchi bands Bands in an EBSD pattern formed by diffraction from crystal planes.

Klein-Gordon equation A relativistic wave equation used in the theoretical framework of EBSD simulations.

Laue group A point group combined with inversion symmetry, relevant in crystallographic texture analysis.

Lie algebra The tangent space at the identity of a Lie group, denoted as $\mathfrak{so}(3)$ for rotations.

Marsaglia sampling A method for generating random points uniformly distributed on a sphere.

Master pattern A simulated diffraction pattern representing the complete sphere of diffraction for a given phase and experimental conditions.

Modality Independent Neighborhood Descriptor (MIND) A method for multimodal image registration that is robust to intensity variations.

Monte Carlo simulation A computational method used to model electron-matter interactions in EBSD.

Orientation Distribution Function (ODF) A function describing the frequency of occurrence of particular orientations in a polycrystalline sample.

Padé approximant A rational function approximation of a power series, used in some numerical methods for orientation calculations.

Pattern center The point of tangency between a sphere emanating from where the electron beam strikes the sample and the EBSD camera sensor.

Point group A group of symmetry operations that can be performed on an infinitely repeating lattice of crystal unit cells without changing the lattice, given that the origin stays fixed.

Polycrystal A material composed of many crystallites or grains, each with potentially different orientations.

Quaternions A number system extending complex numbers, that can be used to represent rotations in 3D space.

Radon transform An integral transform related to the Hough transform, used in some EBSD indexing methods.

Riemannian manifold A smooth manifold with a defined metric, such as SO(3) for rotations.

Rodrigues space The space of Rodrigues-Frank vectors, useful for visualizing orientations and misorientations.

Rodrigues-Frank vector A representation of rotations that scales the axis of rotation by the tangent of half the rotation angle.

Shannon entropy A measure of the amount of information contained in a piece of data, used here in image analysis.

Shoemake sampling A method for uniform random sampling of orientations in SO(3).

Skew-symmetric matrix A square matrix whose transpose equals its negative, related to the representation of rotations.

SO(3) The special orthogonal group in three dimensions, representing all possible rotations in 3D space.

Space group A group of symmetry operations, including translational operations, that can be performed on an infinitely repeating lattice of crystal unit cells without changing the lattice.

Spherical coordinates A coordinate system for representing points on a sphere, often used in EBSD analysis.

Spherical harmonic indexing An EBSD indexing method using spherical harmonics to compute cross-correlations in the frequency domain.

Spherical harmonic transform (SHT) A technique for representing functions on a sphere using spherical harmonics.

Super-Fibonacci approach A method for creating low-discrepancy sequences over SO(3) for uniform orientation sampling.

Texture The distribution of crystallographic orientations in a polycrystalline material.

Wigner D-matrix Matrix elements of the rotation operator which are needed to rotate spherical harmonics.

Wigner little-d matrix A component of the Wigner D-matrix, used in spherical harmonic calculations for rotations.

Zone axis A direction parallel to the intersection of two (or more) families of lattices planes, visible as intersections of Kikuchi bands in EBSD patterns.

CHAPTER I

Introduction

Within materials science and engineering, the rapid advancement of characterization techniques has led to new frontiers in understanding and exploiting the complex relationships relating materials processing, structure and properties. Over the past century, a variety of experimental techniques and methods have been crafted to peer at the intricate details of materials, revealing insights that once were beyond reach. With this explosion in the capabilities and variety of analysis techniques has come a wealth of data, growing both in size and complexity. This presents a formidable challenge to materials engineers who are tasked with distilling beneficial knowledge from this vast information landscape.

At the forefront of this challenge is the field of electron microscopy, and in particular the Scanning Electron Microscope (SEM). It possesses the ability to capture high-resolution images from material surfaces using a variety of signals, having become an indispensable tool for characterizing the morphology, composition, and crystallographic texture of materials from the nanoscale to microscale. For example, one such technique, electron backscatter diffraction (EBSD) enables the mapping of crystalline phase as well as lattice orientation and strain. It has helped to revolutionize our understanding of microstructure-property relationships in poly-crystalline ceramics and metals. However, the power of characterization in the SEM comes at a cost; a single EBSD scan can easily produce a map consisting of millions of data points representing position, orientation, and phase, while the underlying EBSD patterns yielding such a map require billions of bytes to store. Moreover, the rise of in-situ and dynamic studies investigating the evolution of material structure and properties then adds an additional temporal dimension, while serial sectioning studies add an additional spatial dimension. The task of handling and analyzing these large datasets is compounded by the fact that SEM-based characterization usually involves multiple modalities. For example, a typical study might combine EBSD with energy-dispersive X-ray spectroscopy (EDS) to map out the chemistry profile, or with electron channeling contrast imaging (ECCI) to visualize defects and strain fields. Collating these multimodal datasets is a difficult task requiring expertise as the interface of materials science and data science.

In the face of these challenges, materials engineers have increasingly turned to advanced computational methods, particularly computer vision and machine learning to accelerate and automate SEM data analysis, especially EBSD data. The goal of this thesis is to provide a handful of new tools to aid the materials scientist in two aspects: 1) more rapidly acquire scans in the SEM 2) better handle SEM imaging modalities and orientation data.

1.1 Dynamic Sampling and Compressed Sensing

One avenue to the first goal is the development of algorithms for dynamic sampling and sparse data collection. By selecting a subset of raster points to visit during imaging, dynamic sampling can significantly reduce the resources required for data acquisition without drastic compromise to accuracy. This is particularly valuable in reducing the duration of high-cost experiments, such as 3D serial sectioning experiments that currently require several weeks of continuous polishing and EBSD scans. It is also useful in the context of high-throughput experiments meant to screen a large number of materials compositions and/or processing conditions.

Another path to the improvement of characterization throughput is the deliberate rapid acquisition of noisy images combined with fast and noise-robust analysis. This principle is presented in the context of orientation mapping in via EBSD pattern acquisition. The leveraging of physics-based forward models to solve the orientation indexing is the common theme in this approach to alleviating the time cost of data collection. Conventionally, these expensive modelling based approaches to orientation indexing have been considered at the far end of the speed-accuracy Pareto frontier. By applying compressed sensing using principal component analysis, orientation indexing can be accelerated while maintaining noise-robustness qualities.

1.2 Multimodal Image Registration

Several contributions to the second goal of this thesis are presented beginning with image registration. Multimodal image registration in the SEM resides in its own niche among image registration problems. Often the non-linear distortions present between SEM imaging modalities do not reach the magnitude of those found in multimodal medical imaging. Further, unlike histological multimodal imaging registration (e.g. fluorescence microscopy images registered unto corresponding light microscopy images), the local image texture and image gradients in SEM modalities often share significant information. This situates multimodal SEM image registration closest to multimodal remote sensing registration problems such as the alignment of near infrared (NIR) satellite images with their visible (VIS) spectrum counterparts. However, prior knowledge of the relative vantage of satellites in remote sensing make those registration problems comparatively simpler.

To this end, we propose two new approaches to registration for SEM imaging modalities. The first tool is a multimodal key-point detection method based on the information content in images at various scales. This tool is meant to rapidly estimate the registration parameters between two images without the need to traverse over an optimization landscape that so often contains false local optima. The second tool is an improvement on a drop in replacement for cross-correlation meant to handle multimodal data. The cross mutual information function (CMIF) measures the shared information content between two images as a function of discrete pixel shifts of the moving image (although generalizable to other groups via non-commutative harmonics). This function is expensive to evaluate using conventional binning approaches, motivating the development of a generalized Edgeworth series based approximation, dubbed the Edgeworth CMIF (ECMIF).

1.3 Fundamental Zone Reductions

Once EBSD orientation maps, or other orientation data, have been acquired, many different analysis techniques can be used to extract a variety of downstream summaries such as grain size and shape distributions, maps of preferred texture, and more. Crystallographic orientations require careful treatment as they very often possess discrete rotation symmetries, and intermediate results such as misorientations between adjacent points in an EBSD map must be reduced to the Rodrigues fundamental zone (RFZ) before continuing with the larger task at hand. We propose a new decision-tree approach to this reduction to help alleviate the computation burden in tasks where this underlying reduction problem dominates the overall runtime.

1.4 Hypotheses

Pursuantly, the specific hypotheses that I propose are:

- Rasterization-based scans, (e.g. BSE micrographs or EBSD orientation maps), acquired in the electron microscope can be subsampled using dynamic sampling, and then infilled, yielding at twofold reduction in scan time while maintaining an error rate (grayscale pixel values different by 10% relative deviation or misorientations exceeding 3°) not exceeding 1%.

- EBSD patterns can be indexed using compressed sensing via principal component analysis and physics-based forward models to index patterns one order of magnitude faster than conventional pattern matching without compressed sensing, with at least identical indexing accuracy.

- Orientations may be reduced to crystallographic orientation fundamental zones 50% faster than either existing brute force approaches which are based on enumerating equivalent orientations or their angles.

1.5 Thesis Outline

In this thesis, we explore several avenues to improve the accuracy and throughput of characterization in the SEM, with an emphasis on backscattered electron (BSE) micrographs and electron backscatter diffraction (EBSD) orientation maps.

Chapter II covers the essential crystallography background, including orientation and texture representations, the principles and physics of EBSD, as well as existing techniques for EBSD-based orientation determination.

The main body of the thesis is partitioned into four primary chapters. Chapter III details the proposed methodology for dynamic sampling of rasterization-based scans in the SEM to accelerate data acquisition, as hypothesized. It covers the sampling approach, scoring, selection criteria, image completion strategies, implementation details, and performance analysis.

Chapter IV focuses on the use of compressed sensing via principal component analysis (PCA) for noise-robust high speed EBSD pattern indexing. The PCA dictionary indexing (PCADI) is introduced, including the PCA calculation, k-nearest neighbors search, dynamic quantization, and results bench-marking its indexing speed and hardware utilization and evaluating its disorientation distributions on a test Nickel dataset.

Chapter V presents two aforementioned novel tools proposed for multimodal image registration for micrographs from the SEM, addressing the second hypothesis. These registration tools are evaluated using representative EBSD data, but the results are currently preliminary and registration algorithm development is ongoing.

Chapter VI presents decision tree algorithms for the reduction of crystallographic orientations to their fundamental zone equivalents. These methods are evaluated on uniform samplings of orientation space, and on orientation distributions arising from the misorientation between pairs of uniformly drawn fundamental zone orientations. The third hypothesis will be assessed individually in each of these subcases.

Finally, chapter VII summarizes the key findings of the thesis, revisiting the hypotheses and discussing potential future research directions.

CHAPTER II

Background

2.1 Orientation and Texture

The orientation of individual crystallites, or grains, and their collective distributions in a material, exert a profound impact on physical, mechanical, and functional properties of polycrystals. Quantifying orientations and their distributions is essential for establishing structure-property relationships in order to optimize materials [1,7]. For a polycrystal, the distribution of crystal orientations relative to a sample reference frame constitutes the bulk texture.

To analyze orientations, we require a consistent mathematical framework to measure, compare, and manipulate them. The three-dimensional (3D) rotation group, denoted SO(3) provides the fundamental setting for the rotation of crystallites in 3D space relative to a given reference frame. For a thorough overview of crystallographic orientations and texture analysis, especially in the context of group theory, see [8].

2.1.1 SO(3)

The special orthogonal group SO(3) is a Riemannian sub-manifold of 3×3 real valued matrices, denoted $\mathrm{M}_3(\mathbb{R})$. Explicitly, defined with the set:

$$\mathrm{SO}(3) = \{\mathrm{R} \in \mathrm{M}_3(\mathbb{R}) \mid \mathrm{R}^{\mathrm{T}}\mathrm{R} = \mathrm{I} \text{ and } \det(\mathrm{R}) = +1\} \tag{2.1}$$

This Lie group (see Appendix A) is non-abelian, and each of its elements is represented by at minimum three values, although an over-parameterization, or an embedding into a space which is defined in more than three variables can confer numerous benefits to analysis.

2.1.2 Orientation Representations

Axis-angle and $\mathfrak{so}(3)$: The intuitive representation of an element R of SO(3) is simply the unique 3×3 matrix which defines it. Eigendecomposition helps to characterize the action of a matrix upon the vectors that it transforms. For the special case of the neutral identity element, I, the eigenvalues are all equal to unity, while for a rotation of π radians, two of those eigenvalues must be -1, and in the general case, one eigenvalue is unity while the other two are of the form $e^{\pm i\theta}$ where $0 \leq \theta \leq \pi$. There are no unique eigenvectors for the three identical eigenvalues of one in the case of the identity rotation, while for the general case the eigenvalue $\lambda = 1$ is associated with an eigenvector $\mathbf{n}$, that is normalized to a unit vector $\hat{\mathbf{n}}$, the axis of rotation. This direction vector consists of direction cosines, with respect to the three coordinate vectors, (e_x, e_y, e_z). By definition, vectors in 3D space along this direction vector $\hat{\mathbf{n}}$ are unperturbed by R, while others undergo a rotation by a certain angle. The trace, equal to the sum of the eigenvalues of a matrix is invariant to change of basis, yielding:

$$\mathrm{tr}(\mathrm{R}) = 1 + e^{i\theta} + e^{-i\theta} = 1 + 2\cos\theta \tag{2.2}$$

These two pieces identify a rotation $\mathrm{R} = (\hat{\mathbf{n}}, \theta) = (n_x\theta, n_y\theta, n_z\theta) = (\theta_x, \theta_y, \theta_z)$ as an angle-rescaled axis of rotation in a more compact and immediately apparent representation than the 3×3 matrix. One can transform back from $(\hat{\mathbf{n}}, \theta)$ to R via the relationship independently discovered by Euler and Rodrigues [9]:

$$\mathrm{R} = \exp \mathrm{K} = \mathrm{I} + \mathrm{K}\sin\theta + \mathrm{K}^2(1 - \cos\theta), \text{ where } \mathrm{K} = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \tag{2.3}$$

This formulation in equation 2.3 identifies rotations with the family of real 3×3 skew-symmetric matrices, a vector space $\mathrm{Sk}(3, \mathbb{R})$, and the Lie algebra, $\mathfrak{so}(3)$, of the Lie group SO(3). Matrix exponentiation is the operation which takes vectors in the tangent space $\mathfrak{so}(3)$ to the elements in the group SO(3). This tangent space representation is useful for optimization of orientations under a given metric especially in the context of deep learning, although careful integration with existing codebases and frameworks that offer "automatic differentiation" is a necessity as the computation of the Jacobian of the exponential and logarithmic mappings between $\mathfrak{so}(3)$ and SO(3) from forward pass equations can be slower and less numerically stable than explicitly implementing the left Jacobian for the chain rule [10–12]. This treatment of Lie algebras and groups in the context of optimization has been extended to 3D Euclidean motion and similarity transforms, but the underlying relationship between Jacobian and Lie derivatives remains the same (see Ch. 11 of [13]).

Bunge Euler Angles: In crystallography, one of the most commonly used representations of a rotation R is Bunge Euler angles. Euler angles define a rotation as three sequential rotations around the coordinate axes. A Bunge Euler angle triplet usually written $(\varphi_1, \Phi, \varphi_2)$ denotes an intrinsic "ZXZ" ordering, as shown in Figure 2.1 below.

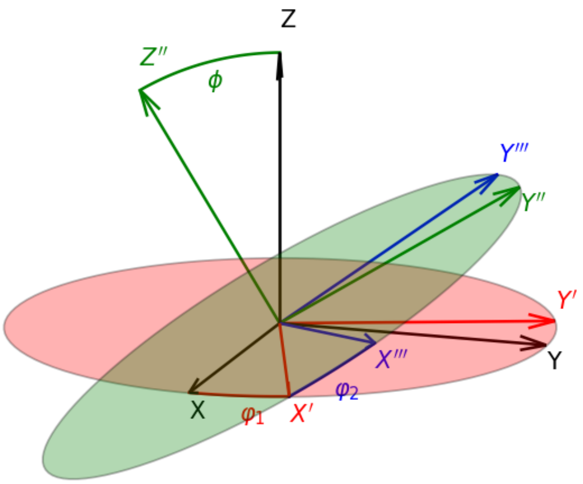

Figure 2.1: ZXZ Bunge angles rotation of reference frame (adapted from Figure 2.3 of [1])

Euler angles suffer from discontinuities and the degenerate identity element is represented by a line in Euler space:

$$(\varphi_1, \Phi, \varphi_2) = (\varphi_1, 0, 2\pi - \varphi_1) \tag{2.4}$$

However, despite these shortcomings, Euler space is periodic along each dimension as equation 2.5 shows:

$$(\varphi_1, \Phi, \varphi_2) = (\varphi_1 + 2\pi, \Phi, \varphi_2) = (\varphi_1, \Phi + \pi, \varphi_2) = (\varphi_1, \Phi, \varphi_2 + 2\pi) \tag{2.5}$$

and this has led to highly effective numerical routines based on the fast Fourier transform for convolutions and more generally harmonic analysis over the space of rotations as we will see in subsection 2.3.3.

Positive Scalar Unit Quaternions: Quaternions, denoted $\mathbb{H}$ are a skew field in $\mathbb{R}^4$: they are abelian under addition, possess an identity operation, and are distributive [8]. Intuitively, quaternions extend complex numbers, and are in turn are extended by Clifford algebras [14]. A quaternion has a scalar part and three imaginary parts:

$$x \in \mathbb{H} = (x_0, x_1, x_2, x_3) = x_0 + ix_1 + jx_2 + kx_3 \tag{2.6}$$

and the units $i, j,$ and k adhere to:

$$\begin{aligned} i \cdot i = j \cdot j = k \cdot k &= -1 \\ i \cdot j = k \quad j \cdot k = i \quad k \cdot i &= j \\ j \cdot i = -k \quad k \cdot j = -i \quad i \cdot k &= -j \end{aligned} \tag{2.7}$$

Quaternions multiply according to the following relationship:

$$\begin{aligned} xy = (&x_0y_0 - x_1y_1 - x_2y_2 - x_3y_3, \\ &x_0y_1 + x_1y_0 + x_2y_3 - x_3y_2, \\ &x_0y_2 + x_2y_0 + x_3y_1 - x_1y_3, \\ &x_0y_3 + x_3y_0 + x_1y_2 - x_2y_1) \end{aligned} \tag{2.8}$$

Quaternions that have unit norm form a double covering of the rotation group:

$$x_0^2 + x_1^2 + x_2^2 + x_3^2 = 1 \tag{2.9}$$

Each quaternion may be arbitrarily negated and it still represents the same rotation as seen in the fact that each component x_i is paired with another component, nullifying the effect of negating x (but note that the two versions can be visualized as forward versus backward rotations to arrive at the same result):

$$\boldsymbol{R}_x = \begin{pmatrix} 2x_0^2 - 1 + 2x_1^2 & 2x_1x_2 - 2x_0x_3 & 2x_1x_3 + 2x_0x_2 \\ 2x_1x_2 + 2x_0x_3 & 2x_0^2 - 1 + 2x_2^2 & 2x_2x_3 - 2x_0x_1 \\ 2x_1x_3 - 2x_0x_2 & 2x_2x_3 + 2x_0x_1 & 2x_0^2 - 1 + 2x_3^2 \end{pmatrix} \tag{2.10}$$

and a unit quaternion can be related to the axis angle formulation as follows:

$$x = (\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\hat{n}) \tag{2.11}$$

Quaternions are useful for efficiently computing the compositions of rotations, and have thus received widespread adoption by computer graphics programmers. By using 4-tuples instead of 3 × 3 rotation matrices composition only requires 16 multiplications and 12 additions instead of 27 multiplications and 18 additions [15]. However, an oft-overlooked drawback of this representation when using standard libraries is that 16, 32, and 64-bit IEEE 754 floating point numbers (each with 10, 23, and 52-bit mantissas respectively), can struggle to store the cosine of half the angle of rotation at small angles [16]. This is because the cosine of half of a near-zero angle is a value very slightly below unity. Table 2.1 shows the apparent accuracy limitations present even for 64-bit floating point representations of the scalar quaternion part near the identity.

Type	Smallest Exact	Round Threshold	Min Angle (°)
FP16	$1 - 2^{-10} = 1 - 9.77 \times 10^{-4}$	$1 - 2^{-11} = 1 - 4.88 \times 10^{-4}$	3.58
FP32	$1 - 2^{-23} = 1 - 1.19 \times 10^{-7}$	$1 - 2^{-24} = 1 - 5.96 \times 10^{-8}$	3.956×10^{-2}
FP64	$1 - 2^{-52} = 1 - 2.22 \times 10^{-16}$	$1 - 2^{-53} = 1 - 1.11 \times 10^{-16}$	1.708×10^{-6}

Table 2.1: Maximum exactly representable scalars, rounding thresholds to unity, and corresponding minimum representable angles for floating point quaternions.

For applications in crystallography a minimum angle of 3.956×10^{-2} degrees provided by 32-bit IEEE floats is often sufficiently below the noise level in estimations of orientation especially for the characterization of texture. In applications where small angles are important such as refining orientations, using axis angle or rotation matrix representations could be a necessary trade of speed in exchange for numerical precision.

Homochoric Representation: The Lambert equal area projection maps the $\mathbb{S}^2$ in 3D space to the 2D disk. The homochoric mapping is an equal-volume generalization of the Lambert equal area projection. It maps the hemisphere of the 3-sphere in 4D space to the 3D ball. For a given axis-angle pair $(\hat{n}, \omega)$:

$$h = \hat{n}\left[\frac{3}{4}(\omega - \sin\omega)\right]^{\frac{1}{3}} \tag{2.12}$$

Cubochoric Representation: Cubochoric coordinates provide a volume-preserving bijection between the three dimensional ball and the three dimensional cube, which in turn provides an equal volume mapping from the cube to the positive scalar quaternion hemisphere via the aforementioned homochoric inversions provided in (B.1) and (B.2), and the conversion given by equation 2.11 [2]. This representation is useful for uniform sampling of orientation space, and the sampling of rotations of crystals possessing (discrete) rotational symmetry. Figure 2.2 shows how the cube is divided into six pyramids which are mapped to divisions of the ball.

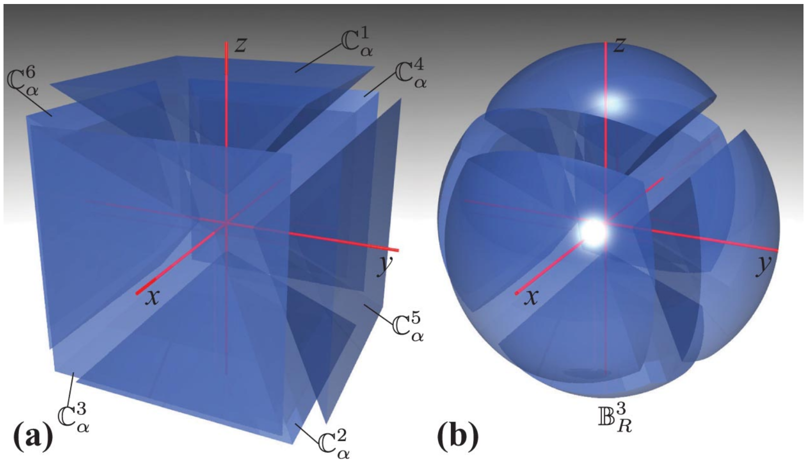

Figure 2.2: First figure of [2] showing exploded view of (a) cube and (b) ball subdivisions in 3D.

Equation 2.13 presents new compact formulae for converting from cubochoric coordinate magnitudes to homochoric coordinate magnitudes for $0 < c_x \leq c_y \leq c_z \leq \frac{\pi^{(\frac{2}{3})}}{2}$:

$$\begin{aligned}
|h_x| &= \frac{\sqrt[12]{2}\cdot\sqrt[3]{3}c_y\sqrt{\frac{\sqrt{2}c_y^2\left(-2\sqrt{2}\cos\left(\frac{\pi c_x}{12c_y}\right)+3\right)+4c_z^2\left(\cos\left(\frac{\pi c_x}{12c_y}\right)-\sqrt{2}\right)}{\cos\left(\frac{\pi c_x}{12c_y}\right)-\sqrt{2}}}\sin\left(\frac{\pi c_x}{12c_y}\right)}{\sqrt[3]{\pi}c_z\sqrt{-\cos\left(\frac{\pi c_x}{12c_y}\right)+\sqrt{2}}}\\
|h_y| &= \frac{2^{\frac{7}{12}}\cdot\sqrt[3]{3}c_y\sqrt{\frac{\sqrt{2}c_y^2\left(-2\sqrt{2}\cos\left(\frac{\pi c_x}{12c_y}\right)+3\right)+4c_z^2\left(\cos\left(\frac{\pi c_x}{12c_y}\right)-\sqrt{2}\right)}{\cos\left(\frac{\pi c_x}{12c_y}\right)-\sqrt{2}}}\left(\sqrt{2}\cos\left(\frac{\pi c_x}{12c_y}\right)-1\right)}{2\sqrt[3]{\pi}c_z\sqrt{-\cos\left(\frac{\pi c_x}{12c_y}\right)+\sqrt{2}}}\\
|h_z| &= \frac{2^{\frac{5}{6}}\cdot\sqrt[3]{3}c_y^2\left(-2\sqrt{2}\cos\left(\frac{\pi c_x}{12c_y}\right)+3\right)+2\cdot\sqrt[3]{6}c_z^2\left(\cos\left(\frac{\pi c_x}{12c_y}\right)-\sqrt{2}\right)}{2\sqrt[3]{\pi}c_z\left(\cos\left(\frac{\pi c_x}{12c_y}\right)-\sqrt{2}\right)}
\end{aligned} \tag{2.13}$$

and the inverse from homochoric for $0 < h_x \leq h_y \leq h_z$:

$$\begin{aligned}
|c_x| &= \frac{2\cdot 6^{\frac{2}{3}}\sqrt{h_x^2+h_y\left(2h_y+\sqrt{h_x^2+2h_y^2}\right)}\sqrt[4]{h_x^2+h_y^2+h_z^2}\,\mathrm{acos}\left(\frac{\sqrt{2}\left(h_x^2+h_y\sqrt{h_x^2+2h_y^2}\right)}{2\left(h_x^2+h_y^2\right)}\right)}{\pi^{\frac{2}{3}}\sqrt{h_z+\sqrt{h_x^2+h_y^2+h_z^2}}}\\
|c_y| &= \frac{6^{\frac{2}{3}}\sqrt[3]{\pi}\sqrt{h_x^2+h_y\left(2h_y+\sqrt{h_x^2+2h_y^2}\right)}\sqrt[4]{h_x^2+h_y^2+h_z^2}}{6\sqrt{h_z+\sqrt{h_x^2+h_y^2+h_z^2}}}\\
|c_z| &= \frac{6^{\frac{2}{3}}\sqrt[3]{\pi}\sqrt{h_x^2+h_y^2+h_z^2}}{6}
\end{aligned} \tag{2.14}$$

and for any triplet where it is not true that $0 < c_x \leq c_y \leq c_z$ for (2.13) or $0 < h_x \leq h_y \leq h_z$ for (2.14), one may take the absolute value and then reorder the triplet in ascending order. After evaluating (2.13) or (2.14) the original ordering must be restored followed by copying the sign of the original coordinates. The semi-edge of the cubochoric cube is $\frac{\pi^{(\frac{2}{3})}}{2}$ and the radius of the homochoric ball is $(\frac{3\pi}{4})^{\frac{1}{3}}$ giving each a volume of π^2 which is the same as the volume of the Northern hemisphere of the unit 3-sphere.

Rodrigues-Frank Representation: The Rodrigues-Frank vector scales the axis of rotation according to the tangent of the half angle [17]:

$$R = \hat{n} \tan \frac{\omega}{2} = (\rho_1, \rho_2, \rho_3) \tag{2.15}$$

This representation is highly advantageous for computations involving the various equivalent rotations of an entity possessing discrete rotational symmetry, as the boundary planes between these symmetry operators in Rodrigues-Frank space are planar. Rodrigues-Frank vectors are composed according to the following relationship:

$$\rho_b \rho_a = \frac{\rho_a + \rho_b - (\rho_a \times \rho_b)}{\rho_a \cdot \rho_b} \tag{2.16}$$

for a composition of ρ_a followed by ρ_b.

Other Representations There are numerous other representations of rotations that deserve mention. Just as the real valued 3 × 3 matrices describe rotations, the special unitary group, SU(2) of 2 × 2 complex valued matrices with determinant of unity may equivalently describe rotations. In another approach, rotations may be mapped to the Clifford torus, and this representation can be used to map crystallographic texture to RGB images [18]. Lastly, there are several embeddings and representations meant to alleviate the effect of symmetry when computing statistics or predicting orientations of objects possessing discrete rotational symmetries. The first method uses generalized spherical harmonics to represent the orientation of crystallites, while the other constructs the embeddings directly from the components [19–21].

2.1.3 Uniform Sampling of Orientation Space

There are several methods to uniformly sample orientation space. Taking four randomly drawn values in the range $[-1, 1]$ and normalizing them to form a unit quaternions does not uniformly draw samples from orientation space. The first approach to uniform sampling is that of Marsaglia, where two point pairs are drawn at random until each pair of coordinates lie within the unit disk in the 2D plane, and thereafter the norm of the second pair is rescaled to form a unit quaternion [22]. This approach requires rejection of points outside of the unit disk, leading to higher computational cost. The method of Shoemake avoids this added cost by reformulating the sampling as follows [23]:

$$q_S = (\sqrt{1-u_1}\sin u_2, \sqrt{1-u_1}\cos u_2, \sqrt{u_1}\sin u_3, \sqrt{u_1}\cos u_3) \tag{2.17}$$

where u_1 is uniformly drawn from $[0,1]$ while both u_2 and u_3 are uniformly drawn from $[0,2\pi]$.

The methods of Marsaglia and Shoemake both independently draw samples which may have "collisions". A low discrepancy sequence or grid over SO(3) is often desirable in a variety of contexts such as numerical integration or Monte Carlo simulation. Alexa's super-Fibonacci approach analyzes the volume-preserving mapping from the cylinder in 3D and SO(3) to formulate a low discrepancy sampling grid over SO(3) using irrational numbers [24]. The original publication proposes the golden ratio and super-golden ratio:

$$q_A = (\sqrt{t}\sin\frac{2\pi s}{\phi}, \sqrt{t}\cos\frac{2\pi s}{\phi}, \sqrt{1-t}\sin\frac{2\pi s}{\psi}, \sqrt{1-t}\cos\frac{2\pi s}{\psi}) \tag{2.18}$$

where $s = i + \frac{1}{2}$, and $t = \frac{s}{N}$ for $i \in [0, ..., N-1]$ for a number of sampling points N. ϕ is the golden ratio and ψ is the super golden ratio. The cubochoric approach maps uniform grids in a 3D cube unto SO(3) using (2.13). The cubochoric and super-Fibonacci grid construction approaches have comparable performance in terms of distribution quality and speed of computation [18].

2.2 Electron Back-scatter Diffraction in the SEM

Electron backscatter diffraction (EBSD) is a powerful technique utilized in materials characterization for crystalline and polycrystalline materials. EBSD leverages the interaction of an electron beam with a polished sample surface to provide a wealth of information about the local crystallographic information including orientation, phase, and defects. At the core of EBSD is the interaction between a high-energy electron beam (typically 1 to 40 keV) in a scanning electron microscope and the sample. The beam is focused down to a spot size on the order of a few nanometers. The interaction volume is at most 5 μm deep, rendering the SEM a surface characterizing analysis technique [25]. EBSD is often combined with other experimental signals. When the electron beam strikes the sample surface and interacts with the atoms to produce characteristic X-rays, which can be analyzed using energy-dispersive spectroscopy (EDS). In ceramics and semiconductors, cathodoluminescence light signals arise when secondary electrons scatter and excite valence electrons into the conduction band and subsequently combine with a hole.

2.2.1 Experimental Geometry and Setup

The typical EBSD geometry is shown in figure 2.3:

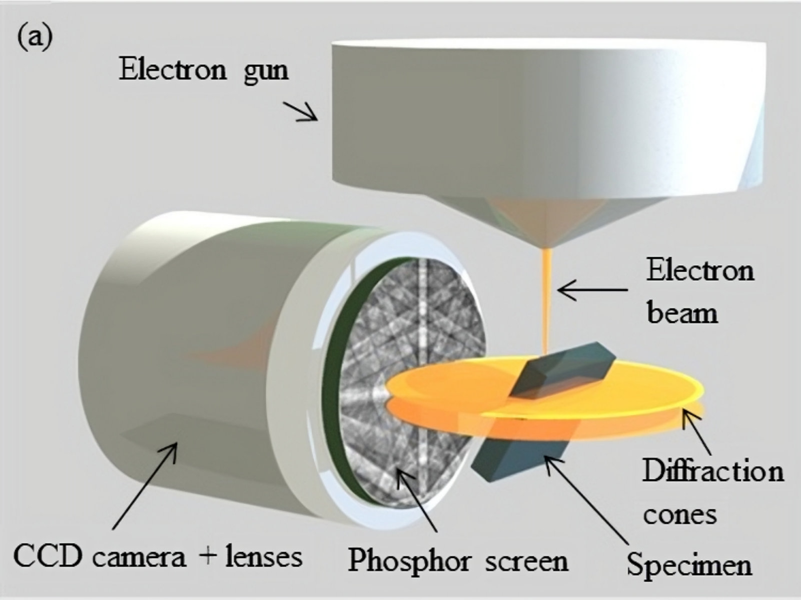

Figure 2.3: Typical EBSD geometry (reproduced from [3] - CC 3.0 License.)

In EBSD, the sample normal is typically tilted down by 70° towards the direction of the detector. This is done to maximize proportion of the backscatter electrons that are directed towards the detector. The cost of this is a variety of distortions that can be introduced [26]. The geometry of the detector is generally fit after the fact using software on a per-pattern basis, as a triplet of coordinates: the pattern center. The pattern center describes the location in the image plane of the point of tangency with the Kikuchi sphere as well as the distance from this point to the point of beam incidence and recent efforts have begun to use global optimization to fit these values [27].

2.2.2 Electron Diffraction Forward Model

In order to extract the most amount of information out of experimental data it is advantageous to have a simulation of the physics involved in electron diffraction. This subsection explains how a two step approach is used to simulate the backscatter electron signal in an SEM.

Monte Carlo Simulation of Backscatter Electrons

The simulation of electron-matter interactions has often been cast in the language of Monte Carlo modeling [28–31]. The forward model utilized in the present work bases its Monte Carlo portion largely on the book by David C. Joy [32]. The Monte Carlo model makes the simplification that during each step a stochastic but guaranteed removal of energy is applied. This significant "continuous" slowing approximation makes the simulation more computationally viable by in essence fitting the end result effect of discrete interaction events. The inputs to the Monte Carlo simulation are the atomic number, atomic mass, accelerating voltage, and the tilt of the sample. The output is used to approximate a 4D probability distribution over outgoing direction (described by polar and azimuthal ϕ and ψ), energy, and escape depth. These outputs are then used to perform a dynamical scattering simulation.

Dynamical Scattering Simulation

Following the Monte Carlo simulation, the Klein-Gordon equation is made tractable with a high-energy approximation and a Bloch-wave ansatz leading to an eigenvalue problem:

$$\mathcal{A}\mathbf{C}^{(j)} = 2\pi\lambda k_n \Gamma^{(j)}\mathbf{C}^{(j)}, \tag{2.19}$$

where $\mathbf{C}^{(j)}$ are the Bloch-wave coefficients, $\Gamma^{(j)}$ are the complex eigenvalues, and k_n is the normal component of the incidence wavevector. The solution can be expressed as:

$$\Psi(\mathbf{r}) = \sum_j \alpha^{(j)} \sum_{\mathbf{g}} C_{\mathbf{g}}^{(j)} \exp\left(2\pi\mathrm{i}\left(\mathbf{k}_0 + \Gamma^{(j)}\mathbf{n} + \mathbf{g}\right)\cdot\mathbf{r}\right) \tag{2.20}$$

for Bloch-wave excitation amplitudes $\alpha^{(j)}$, where $\mathbf{g}$ is the lattice plane, $\mathbf{r}$ is the position, $\mathbf{k}_0$ is the exit direction (for BSE electrons). This solution is substituted into the form of the backscattering probability equation for a direction $\mathbf{k}$:

$$P(\mathbf{k}) \simeq \frac{1}{z_0} Z^2 \exp(-M) \sum_{i=1}^{N} \int_0^{z_0} \mathrm{d}z \, |\Psi_{\mathbf{k}}(\mathbf{r}_i; z)|^2 \tag{2.21}$$

yielding a probability over $\mathbf{k}$ and energy E:

$$\begin{aligned} P(\mathbf{k}, E) &\simeq \sum_{\mathbf{g}} \sum_{\mathbf{h}} S_{\mathbf{gh}} L_{\mathbf{gh}} \\ S_{\mathbf{gh}} &= \sum_{n} \sum_{i \in S_n} Z_n^2 \exp\left(-M_{\mathbf{h}-\mathbf{g}}^{(n)}\right) \exp\left(2\pi \mathrm{i}(\mathbf{h}-\mathbf{g}) \cdot \mathbf{r}_i\right) \\ L_{\mathbf{gh}} &= \sum_{j} \sum_{k} C_{\mathbf{g}}^{(j)*} \alpha^{(j)*} \mathcal{I}_{jk}(E) \alpha^{(k)} C_{\mathbf{h}}^{(k)} \end{aligned} \tag{2.22}$$

where $S_{\mathbf{gh}}$ is similar to a structure factor and $L_{\mathbf{gh}}$ comes from solving equation 2.19. The end result of this forward model for electron backscatter diffraction is a probability distribution over energy and $\mathbb{S}^2$ of backscattering electrons exiting the sample.

2.2.3 Automation of EBSD Indexing

The first experimental electron diffraction patterns were observed in 1928 by Nishikawa and Kikuchi from the surface of a calcite sample, as well as ”cleavage faces of mica, topaz, zincblende and a natural face of quartz” [4]. Figure 2.4 shows the first EBSD pattern collected for calcite:

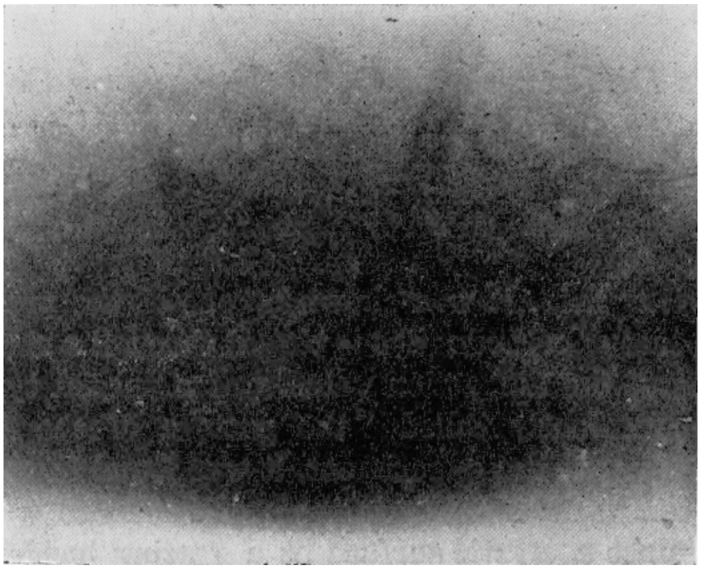

Figure 2.4: Image reproduced from [4] showing the first ever EBSD pattern (adaptive histogram equalization applied to show more details).

These images were indexed by manual inspection without computers. Since then, many methods have been developed to automatically index Kikuchi patterns, particularly the backscatter mode. The first procedure to introduce fully automatic indexing using the computer was developed in 1992 using the Hough transform and Burns algorithm [33, 34]. The second technique published in 1993 utilized the Hough transform and butterfly mask [35]. In 1997, the first software-directed and SEM-interactive automatic indexing method brought dynamic focusing and calibration support for the SEM and used the Hough-Radon transform without a butterfly mask to find the pattern center (PC) and at least 3 bands for orientation determination. [36]. The Hough and Radon transforms are closely related. The Hough transform fills the parameter space (angle and distance for lines) bins, input pixel by input pixel, based on membership in a given bin, while the Radon transform calculates each parameter space value one by one with the relevant input pixels [37].

2.3 EBSD Orientation Determination

In the years since the advent of Hough indexing, numerous methods including spherical indexing, dictionary indexing, deep-learning based methods, and hybrid deep learning with dictionary approaches have been developed to improve indexing speed as well as accuracy in the face of shot noise in the SEM [5, 38–40]. Figure 2.5 shows that dictionary indexing overtakes other indexing methods in noise-robustness, and similar outcomes are also seen in the later developed hybrid EBSDDI-CNN approach of [40]. Importantly, spherical indexing with higher band count would likely improve the noise robustness towards the level of dictionary indexing, but a GPU implementation is not currently publicly available, and runtime becomes a major deterrent from adoption at high bandwidth. One goal of this work is to retain the noise robustness of dictionary indexing while lowering its runtime to be near that of Hough indexing (thousands or tens of thousands of patterns per second on a consumer grade GPU).

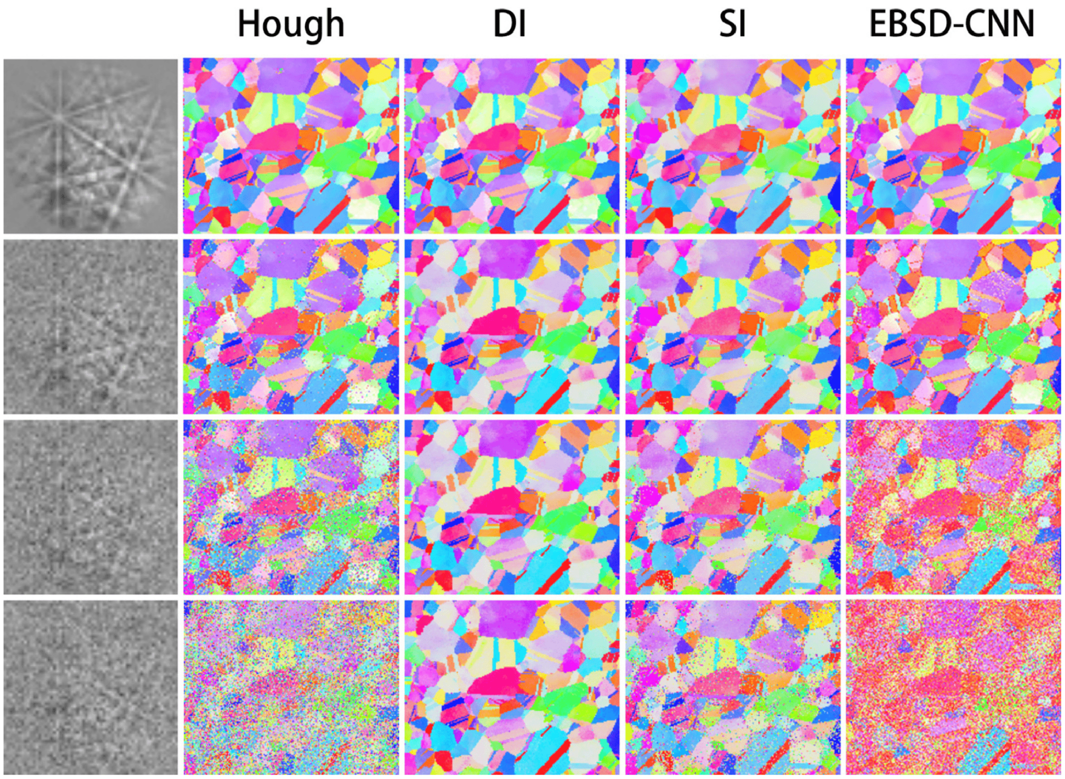

Figure 2.5: Figure reproduced from [5] compares indexing methods at increasing noise levels: ”From left to right are patterns, indexing results of Hough, DI, spherical indexing (SI) with 87 bands, and EBSD-CNN.”

2.3.1 Hough Transform Indexing

The Radon and Hough transform are closely related to each other and they both transform lines into points in their output space [37]. This transformed space is useful for EBSD orientation indexing as Kikuchi bands may be identified and used to infer the orientation of the crystalline lattice. Figure 2.6 shows the sinogram resulting from the radon transform of an example EBSD pattern. The zone axis at the intersection of each of the colored lines traces a sine wave in the radon transform space. By using convolution filters or other analysis methods the Kikuchi bands may be identified and then the relative angles between them can be used to vote on a particular crystalline orientation. This approach forms the basis of Radon and/or Hough transform based EBSD orientation indexing.

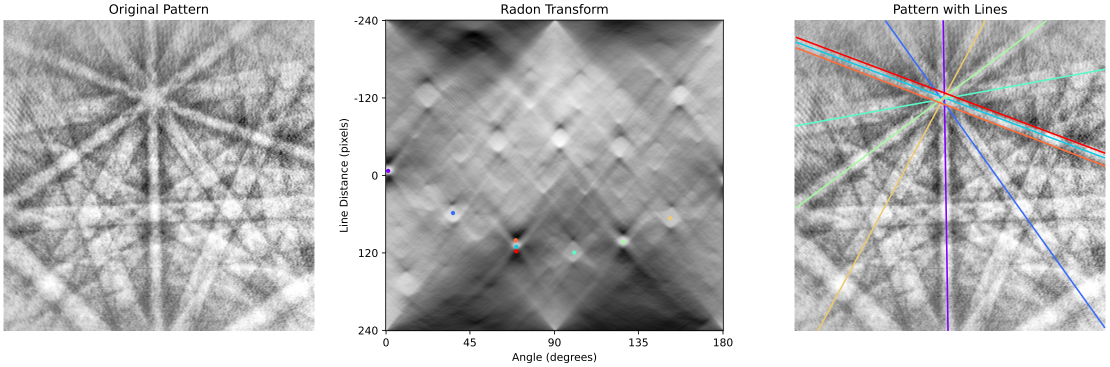

Figure 2.6: The Hough and Radon transform convert lines to points which facilitates calculating angles between diffraction bands. The vertical axis is the distance of the line from the center of the pattern and the angle tells the orientation of the line.

2.3.2 Dictionary Indexing

Dictionary indexing uses the physics-based forward model of EBSD to simulate the signal on $\mathbb{S}^2$ (often called the Kikuchi or “K” sphere). From this reference signal called the master pattern, a titular dictionary of possible idealized experimental observations can be constructed. Each experimental observation may then be systematically compared to all of these possible dictionary entries in order to procure a set of orientations that have simulated patterns closest to that of the experimental observation. The most common metrics for these comparisons are the normalized dot product, mutual information, and normalized cross correlation. This approach leverages uniform grids over SO(3), particularly cubochoric coordinates to sample orientation space. These orientations are then used to rotate the K-sphere before an image is interpolated from the master pattern. The master pattern is stored using an equal-area bijection between the circle and square, which readily permits bilinear, bicubic or higher order interpolation with its equal volume pixels.

2.3.3 Spherical Harmonic Indexing

Spherical harmonic indexing uses the mathematics of harmonics on $\mathbb{S}^2$ and on SO(3) to compute the normalized cross correlation between a master pattern and an experimental pattern over the rotations in the spectral domain where convolution is equivalent to element wise products [38,41–44]. A function on $\mathbb{S}^2$ can be expressed as a series expansion of spherical harmonics Y_m^ℓ and spherical harmonic coefficients $\hat{f}_m^\ell$ via the spherical harmonic transform (SHT):

$$f(\theta, \phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{+\ell} \hat{f}_m^\ell Y_m^\ell(\theta, \phi) \tag{2.23}$$

where the coefficients are defined as:

$$\hat{f}_m^\ell = \int_0^\pi \mathrm{d}\theta \sin\theta \int_0^{2\pi} \mathrm{d}\phi f(\theta, \phi) \overline{Y_m^\ell(\theta, \phi)} \tag{2.24}$$

where the spherical harmonics $Y_m^\ell(\theta, \phi)$ have their physics-convention (as opposed to the geographic convention) definition (θ is the polar declination and ϕ is the azimuthal angle):

$$Y_m^\ell(\theta, \phi) = \sqrt{\frac{(2\ell+1)!(\ell-1)!}{4(\ell+m)!}} P_m^\ell(\cos(\theta)) \mathrm{e}^{im\phi} \tag{2.25}$$

with $P_m^\ell(x)$ the associate Legendre polynomial of degree ℓ and order m, with coefficients determined by:

$$\hat{f}_m^\ell \approx \sum_k f_k w_k \overline{Y_m^\ell\left(\theta_k, \phi_k\right)} \tag{2.26}$$

There are variety of ways to compute the spherical harmonic transform, including different sampling grids. Once the spherical harmonic coefficients for a signal on $\mathbb{S}^2$ have been determined, the spherical cross correlation (SCC) over SO(3) can be computed via the inverse Fast Fourier transform. In the spectral domain, the SCC can be expressed as a simple triple summation over the indices of the harmonics for an element wise product modified by the entries of the unitary matrix in an irreducible representation of SO(3): the Wigner-D matrix.

$$(f \star g)(R) = \sum_{\ell,m,n} \hat{f}^{\ell}_{m} \overline{\hat{g}^{\ell}_{n}} D^{\ell}_{m,n}(R) \tag{2.27}$$

where a deliberately chosen "ZYZ" (and not "ZXZ") intrinsic Euler angle convention with angles (α, β, γ) yields real valued Wigner D coefficients:

$$D^{\ell}_{m,n}(\alpha, \beta, \gamma) = d^{\ell}_{m,n}(\beta)\mathrm{e}^{\mathrm{i}n\alpha}\mathrm{e}^{\mathrm{i}m\gamma}, \tag{2.28}$$

with Wigner little-d given in terms of Jacobi polynomials $P^{m-n,m+n}_{\ell-m}(\cos(\beta))$ as:

$$d^{\ell}_{m,n}(\beta) = \sqrt{\frac{(\ell+m)!(\ell-m)!}{(\ell+n)!(\ell-n)!}} \cos\left(\frac{\beta}{2}\right)^{m+n} \sin\left(\frac{\beta}{2}\right)^{m-n} P^{m-n,m+n}_{\ell-m}(\cos(\beta)) \tag{2.29}$$

and these coefficients must be carefully computed and tabulated as at high degrees in ℓ, numerical precision becomes a significant concern. Researchers have often resorted to virtual extended numerical precision at a small cost to speed [45,46]. Others have developed FFT-based methods to evaluate these coefficients at high degrees ℓ and their numerical evaluation is an area of active research [47].

The key to spherical harmonics based indexing is decomposing equation (2.27) into two separate rotations and the inverse 3D Fourier transform:

$$(f \star g)(\alpha, \beta, \gamma) = \mathscr{F}^{-1}\left\{\sum_{\ell} \hat{f}^{\ell}_{m} \overline{\hat{g}^{\ell}_{n}} d^{\ell}_{m,k}\left(\frac{\pi}{2}\right) d^{\ell}_{k,n}\left(\frac{\pi}{2}\right)\right\}\left(\alpha + \frac{\pi}{2}, \beta + \pi, \gamma + \frac{\pi}{2}\right). \tag{2.30}$$

Conveniently the little-d matrix entries are only required for β of $\frac{\pi}{2}$ for this application which simplifies their recursive calculation. Further, there are systematic zeros in the spherical harmonic transform coefficients in equation (2.24) resulting from symmetry of the master pattern signal $f(\theta, \phi)$ that must be recognized to calculate equation (2.30) as rapidly as is possible. Firstly, the signal in EBSD is real valued on the sphere so $\hat{f}^{\ell}_{m}$ must have conjugate symmetry, and our 3D inverse FFT may be a half-sized real-valued one:

$$\hat{f}^{\ell}_{-m} = (-1)^{m} \overline{\hat{f}^{\ell}_{m}} \tag{2.31}$$

and a mirror within the xy plane, inversion symmetry, and n-fold rotations about the z-axis each reduced the number of possible non-zero coefficients by a factor of 2, 2, and N respectively.

2.4 More on the Homochoric Inverse

This section expands more on the homochoric orientation representation, specifically the inversion of equation (2.12), but these results do not warrant their own chapter in the dissertation.

The conversion of the homochoric vector back to the axis angle representation requires a numerical approximation via a polynomial fit or other approach. In fact, this inversion is closely related to the solution of Kepler's equation, $E - e \sin E = l$, which has seen several centuries of research [48]. The homochoric inversion is the cube root of a scaled version of Kepler's equation with eccentricity $e = 1$. While the inverse has an exact integral representation [49], for rapid evaluation, both EMsoft and Kikuchipy use a polynomial fit on the cosine of the half angle in terms of the squared homochoric modulus. The cosine of the half angle modification makes the Chebyshev interpolation nodes equally spaced for polynomial fitting, while the squared homochoric modulus is fitted as it is immediately available from the sum of the squared components (save computation by having the polynomial fit absorb the square root). The original and modified homochoric functions are plotted in Figure 2.7:

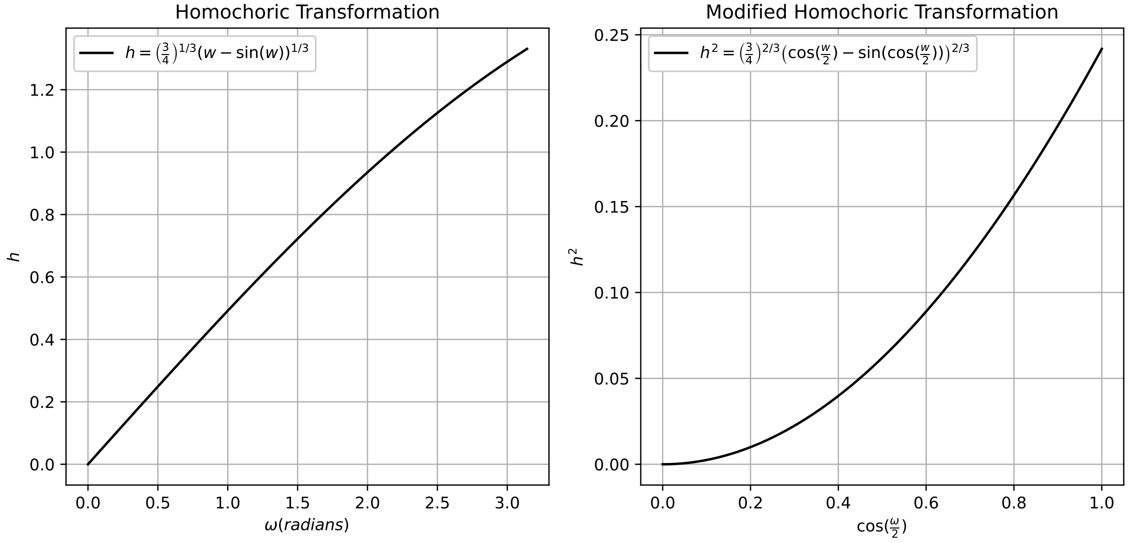

Figure 2.7: (left) original homochoric conversion and (right) the modified homochoric equation

Few sources explain the full details of numerical estimations to this particular inverse and we will address them here. Appendix B provides the polynomial fits taken from current libraries to demonstrate how the rotation angle can be estimated from the homochoric squared Euclidean norm. These polynomial fitting methods trade some accuracy compared to Newton's method in turn for a significant speed advantage, especially on GPU's which can do the reduction of the polynomial terms in parallel. Newton's method can use as initialization an invertible Padé approximant around $\omega = 0$ of equation 2.12 such as the following pair:

$$\begin{aligned} h \approx& \frac{\omega}{\frac{\omega^2}{30} + 2} \\ \omega \approx& \frac{15 \pm \sqrt{15}\sqrt{15 - 4h^2}}{h} \end{aligned} \tag{2.32}$$

and iterate for 3 to 4 steps to convergence. This estimate is crucial in evaluating the inverse at Chebyshev nodes for polynomial fitting. Figure 2.8 shows the error in the inversion estimate over the applicable angle range $[0, \pi]$ as compared to the results of iterative methods like Newton's or Halley's operating on the guess from equation 2.32 for 3 and 2 iterations respectively:

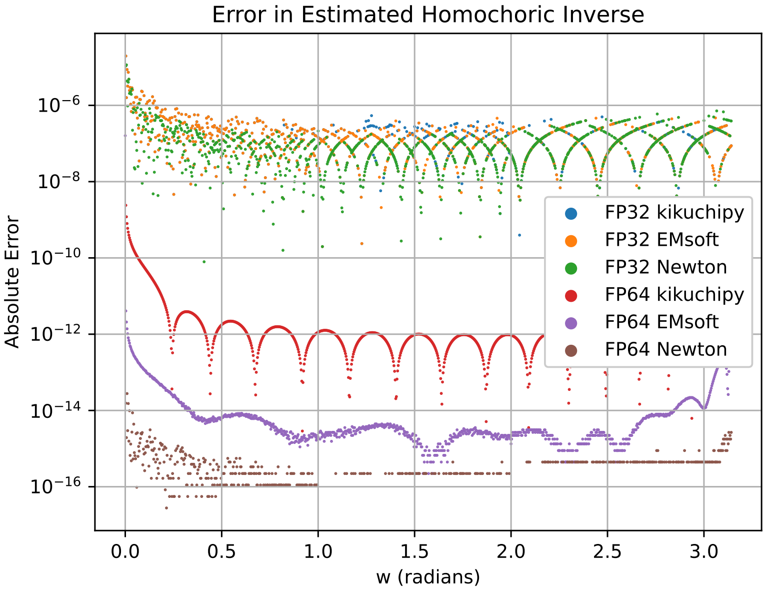

Figure 2.8: Uniformly spaced input angles are converted to homochoric moduli and then inverted and the absolute error in the inversion in radians (logscale) is shown for both 32-bit and 64-bit floating numbers

It is clear that the accuracy compromise is negligible, especially in the context of crystallography and texture. Storing the length of the homochoric vector as the Euclidean norm of its components can introduce further error into the estimates, and figure 2.9 shows the lack of any effect in single precision:

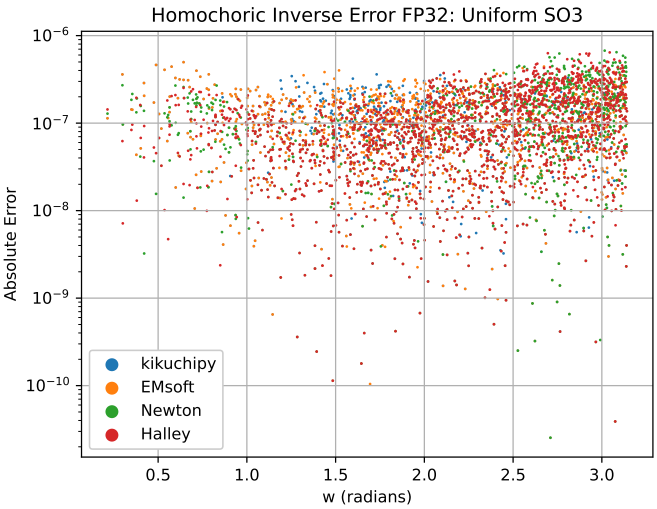

Figure 2.9: In 32-bit precision, uniformly sampled (Shoemake approach) scaled axis-angle vectors are converted to homochoric vectors and then the angle is estimated from various methods.

It's apparent that there is not a significant change in the accuracy from the case of figure 2.8. However the results are different in the case of double precision as seen in figure 2.10:

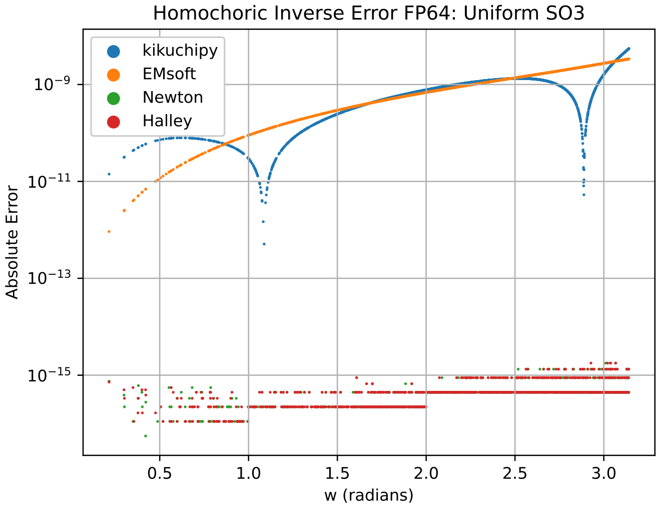

Figure 2.10: In 64-bit precision, uniformly sampled (Shoemake approach) scaled axis-angle vectors are converted to homochoric vectors and then the angle is estimated from various methods.

In practice, the accuracy of the Kikuchipy coefficients results in a comparable prediction as compared to EMsoft despite using 5 fewer coefficients. In general, Chebyshev polynomial fits to the original and modified homochoric inverse have residuals shown in figure 2.11:

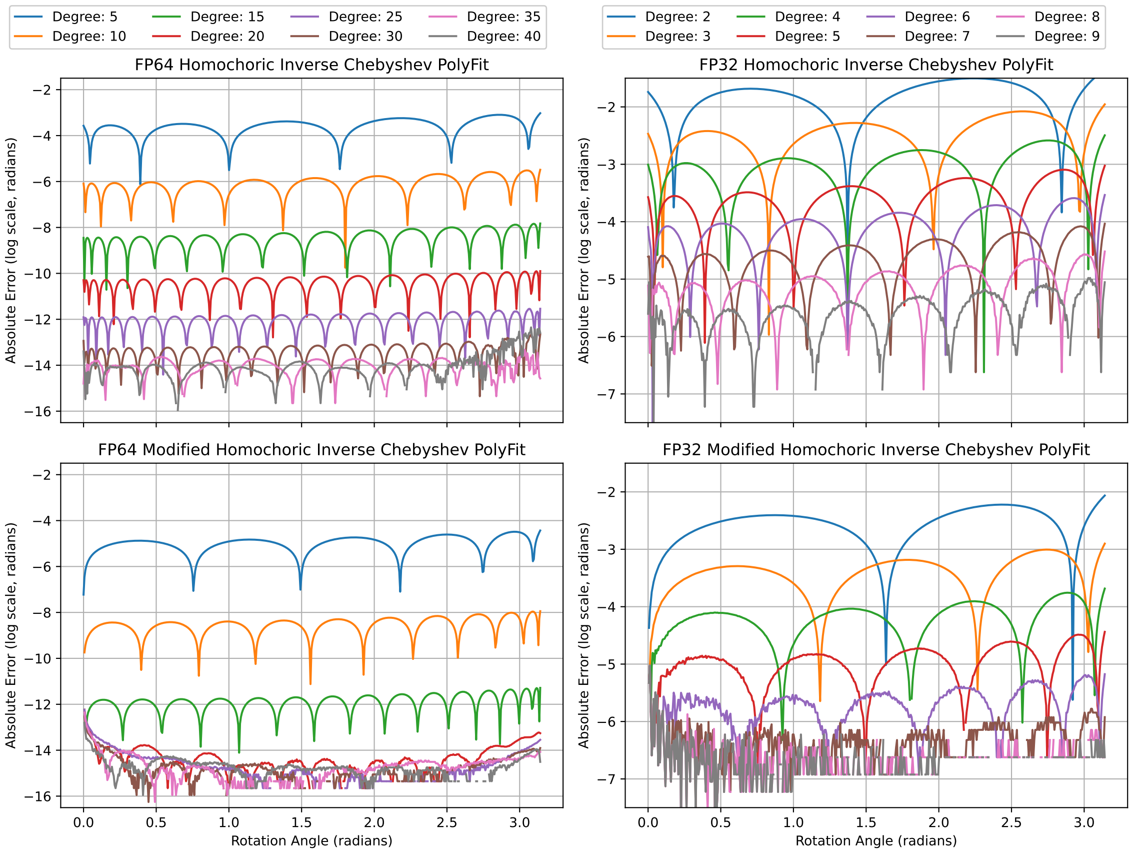

Figure 2.11: Residuals of Chebyshev polynomial fits to inverse of equation 2.12 and modified inverse for both single and double precision.

By using a modified homochoric inversion with equally spaced Chebyshev nodes with respect to ω, the absolute error over the entire domain $[0, \pi]$ is improved by more than two orders of magnitude as compared to the unmodified inversion when the degree exceeds 5 for double precision. For single precision a fit with 8 terms is sufficient to reach machine error. Conveniently, the highest error for polynomials with degree in excess of 15 is near zero radians rotation angle which is exceedingly rare under a uniform drawing of orientation space as figure 2.9 and 2.10 suggest; however this can become important when comparing similar rotations. The historical approach has been to use enough terms (20 in the case of EMsoft) that in practice render the error negligible.

CHAPTER III

Dynamic Sampling of Scanned Modalities

3.1 Introduction

Electron backscatter diffraction (EBSD) performed in a scanning electron microscope (SEM) offers a powerful and ubiquitous modality for the analysis of the microstructure of crystalline samples, generating rich and quantitative orientation data [50]. Chiefly, EBSD can probe a large mm-scale ($\leq 10^6$ μm^2) field of view at a resolution of 40–100 nanometers, well below common grain size ranges, allowing for grain-resolved data acquisition [51]. The EBSD modality makes use of the standard line scanning approach made available by SEM manufacturers; the diffraction patterns are acquired one line at a time, with user-defined step size, resulting in discrete data sets on a 2-D square or hexagonal grid of sampling points. Most SEMs, however, offer external scan controls, which enable the user to control the beam directly to create alternative sampling schemes. In this paper, we explore an unsupervised dynamic sampling scheme that can generate the same orientation data as the standard EBSD technique, but with a significantly reduced number of sampling points; our approach is not limited to EBSD and can be applied to any characterization technique that allows for point-wise sampling of a region of interest.

Previous research has leveraged the coarse structuring of EBSD data for data acquisition speed and accuracy; for instance, non-local pattern averaging and reindexing seeks to increase indexing noise-robustness by exploiting the spatial correlation of Kikuchi patterns through nonlocal averaging during postprocessing [52]. Rapid EBSD is another approach that relies on forescatter electron imaging to segment one serial section of the microstructure for static sparse EBSD sampling [53]. Other efforts, such as the “supervised learning approach for dynamic sampling" (SLADS), implement a supervised learning approach for the pixel-wise dynamic sampling of a region of interest (ROI); this is applicable to general image sampling problems, including EBSD applications [54]. SLADS trains a regression model to predict the utility of the next sample pixel based on a number of calculated features of the local neighborhood of already measured pixels; the pixels that are chosen for acquisition are determined by maximizing the reduction of an error metric, the "expected reduction in distortion" or ERD, i.e., one attempts to select as the next sample, the pixel that will provide the maximum amount of information.

To our knowledge, apart from the SLADS algorithm, no other immediately relevant work exists for pixel-wise dynamic sampling. SLADS' offline training approach requires data that is similar, including identical ROI dimensions, to the expected input for dynamic sampling. Additionally, because continuous SLADS infilling (i.e., inserting data for non-sampled points) does not necessarily ensure valid orientations, a discrete infilling mode was utilized. The discrete SLADS infilling is an inverse-distance weighted mode interpolation with a user-adjustable exponent for the weights. The infilling is not a trained part of the dynamic sampling model, unlike the work presented here. Having to use the discrete infilling mode means that SLADS can only be tested with synthetic data, wherein pixels take discrete single-channel (gray-scale) values, not crystal orientations, which generally require three parameters. Although the implementation of SLADS examined here for comparison has least squares linear regression as its core model, subsequent development of SLADS combined the approach with a neural network, resulting in SLADS-net [6]. It is important to recognize that the SLADS-net accuracy was shown to be significantly improved over the original SLADS, although the sampling run-time per pixel remained the same. The work presented here is an unsupervised dynamic sampling method, with no need for offline training.

3.2 Sampling Methodology and Test Datasets

3.2.1 Motivation

This section outlines a method by which 2D images may be dynamically sampled and then infilled. The main motivation in design and implementation choices was runtime, to permit usage in electron microscopy applications. The average time needed to choose measurements should be faster than the typical electron beam dwell times; however, this is not strictly required to achieve an advantage over a full grid scan-line-based raster. To make a dynamic pixel-wise sampling algorithm, measurements acquired as the sampling progresses should be used to inform subsequent measurement choices. We have pursued a different approach than the SLADS method which selects samples by maximal expected reduction in distortion (ERD). The present work relies on a paradigm of scoring potential measurements by a product of the distance to the nearest measurement and the variance of the nearest K measurements.

3.2.2 Scoring and Selection

To filter candidate measurements, a function FILTER described in pseudo-code in algorithm 1 is defined. FILTER takes a set of pixels and returns a set of winning candidates for measurement. It uses a value function $V(p)$ to determine this winning set. This value function $V(p)$ of a pixel p is defined as the product of its minimum distance to a measured neighbor and the standard deviation of all K neighbor values. The value $V(p)$ of a measured pixel p is arbitrarily low, as no pixel measurement will be made twice. The window function WINDOW(p) facilitates non-maximal suppression by returning a set of nearby pixels, within a square window of size W, excluding the pixel p itself. In this approach, pixels far from their closest measured neighbor, and possessing a local neighborhood of measured pixels with high variance are the most favorable measurements.

In algorithm 2, SAMPLE takes in an underlying image, an initial sampling SOBOL, a stopping fraction STOP, and a filtering function FILTER. At the start of data acquisition, no pixels have been observed, so the set of measured pixels M is empty, and the set U holds all pixels in the image. Firstly, a Sobol sequence, one possible low discrepancy quasi-random sequence, is used to draw an initial exploratory sampling [55,56]. This choice is more favorable than fully random sampling due to lower collision rates. These initial measurements are removed from the set of unmeasured pixels U and joined to the set of measured pixels M. Thereafter, the function FILTER continues to select pixels for measurement until a cutoff fraction STOP of the image is reached.

Algorithm 1
Scoring Function FILTER()

```
Input: unmeasured set: $U$, value function: $V$, fraction: $F$, window
  function: WINDOW
  $C^0 \leftarrow U$
  $W^0 \leftarrow \emptyset$
  while $\frac{|W|}{|C|} < F$ do
      for candidate pixel $c \in C$ do
          if $V(c) > V(n)\ \forall$ pixel $n \in \mathsf{WINDOW}(c)$ then
              $W^{i+1} \leftarrow W^i \cup \{c\}$
  return $W^i$
```

Algorithm 2
Sampling routine SAMPLE()

```
Input: image set: IMG, pixel set: SOBOL, stop percent: STOP,
  function: FILTER()
  $M^0 \leftarrow \emptyset$
  $U^0 \leftarrow \{$pixel p for p in IMG$\}$
  $M^{i+1} \leftarrow M^i \cup \mathsf{SOBOL}$
  $U^{i+1} \leftarrow U^i \setminus \mathsf{SOBOL}$
  while $\frac{|M|}{|M|+|U|} < \mathsf{STOP}$ do
      $M^{i+1} \leftarrow M^i \cup \mathsf{FILTER}(U)$
      $U^{i+1} \leftarrow U^i \setminus \mathsf{FILTER}(U)$
  return $M^i$
```

3.2.3 Implementation

Euclidean Distance Transform Implementation

The sampling algorithm is built upon PyTorch, a versatile machine learning library known for its extensive tensor operations on both GPU and CPU, including just-in-time compilation, and for its user-friendly interface and ease of installation [57]. One of the primary motivations for opting for PyTorch is its straightforward installation process, which typically requires just a few commands. For candidate scoring, our implementations assume a single-channel image or an image that can be mapped to a single channel. In cases where multichannel pixel values are involved, such as orientations in EBSD, these are filled in using their nearest measured values. The source code is available under a 3-clause BSD license, consistent with that of PyTorch at https://github.com/ZacharyVarley/DynamicSampling.

The first component of the score in the dynamic sampling strategy we propose is the Euclidean distance to the nearest measured point. While it is possible to employ data structures like k-dimensional trees (k-d trees) optimized for nearest neighbor searches in low dimensions, a more efficient alternative exists that exploits the discretized nature of the microscope grid. Crucially, our scoring mechanism doesn't necessitate identifying which specific measurement is the closest; we merely need the distance to the nearest measurement. Leveraging this insight, we utilize the Euclidean distance transformation (EDT). This approach allows us to calculate the distance to the nearest measured point for all unmeasured points in linear time, O(N) according to the number of image pixels, which significantly enhances the algorithm's efficiency. The repository "FastGeodis: Fast Generalised Geodesic Distance Transform" has convenience Python wrappers with implementations written in CUDA and OpenMP [58].

For the second part of the score, we use the windowed variance of the image, which can be effectively calculated using several Summed Area Tables (SATs). Specifically, SATs are computed for a mask of measurement locations, the sampled locations (with zeros where no measurements exist), and the square of the sampled locations (again with zeros where no measurements exist). These three SATs can then be used to calculate the variance within any window by the difference in the expectation and square expectation. Variance values for pixel windows that have fewer than three measurements within were set to unity so that only the Euclidean distance decides between them. SATs also have O(N) construction keeping the overall runtime linear in the number of pixels.

The pixel averaged runtime in μs of this implementation on both the CPU and GPU is given in the first two rows of Table 3.1 below. It shows that the runtime per pixel when sampling a random image is approximately constant on both devices, even as the resolution increases dramatically. Further, the average time for most samplings is well below a modest

10 μs dwell time, crucial to the adoption in SEMs for electron backscatter microscopy.

KeOps Implementation

While data structures and algorithm complexity analysis can indicate scaling, often it is important to benchmark against brute force approaches. Rapid brute-force k-nearest neighbor queries can be carried out with the library KeOps, which seamlessly integrates with PyTorch [59]. The KeOps Python bindings (PyKeOps) compile optimized CUDA kernels at runtime, and they are ideal for map-reduce type problems. Because the nearest k neighbors are returned, the window size for variance calculation is not needed, and it is replaced with the number of neighbors to consider. Table 3.1 below shows that the KeOps implementation permits dynamic sampling with dwell times below under 2 μs so long as the resolution does not exceed 1024x1024.

Due to the improved performance, the KeOps implementation is the main implementation analyzed in the results. Parameter W is the window size of the filtering operation for picking local optima in the scores, F is the top fraction of local optima to accept, K is the number of neighbors used to calculate variance, and S is the Sobol coordinate seed size. If S is a fraction, it refers to the fraction of the total number of pixels in the image, while if it is an integer above 1 it refers to the exact number of pixels (useful notation for small ROIs). Default values around W=3, F=0.25, K=3, and S=0.05 are suitable for most applications.

Implementation	Image Resolution				
	128x128	256x256	512x512	1024x1024	2048x2048
EDT GPU	8.58 ± 0.13	3.96 ± 0.06	2.42 ± 0.03	1.82 ± 0.02	1.72 ± 0.03
EDT CPU	11.24 ± 0.26	6.62 ± 0.09	5.33 ± 0.18	5.83 ± 0.21	8.75 ± 0.38
KeOps GPU	0.94 ± 0.02	0.60 ± 0.04	0.56 ± 0.01	1.90 ± 0.01	7.49 ± 0.09

Table 3.1: Average compute time per pixel in μs (10 trials) across resolutions for random images. The devices used for this benchmark were a Nvidia T4 GPU and 4 Intel Skylake CPU cores.

3.2.4 Image Completion

From the sampled pixels in the image, each unmeasured pixel is infilled as either the inverse distance weighted or unweighted mean of the values of its K-nearest neighbors. As averaging distinct crystalline orientations at grain boundaries falsely produces smoothed boundaries, a single nearest neighbor imputation was utilized for EBSD data. Disorientation angles were calculated using Orix, a Python library for crystal misorientations [60].

3.2.5 Performance Evaluation

For orientation data, the disorientation angle between predicted and ground truth crystalline orientations was used to examine imputation fidelity. For grayscale micrographs, the reconstruction peak signal-to-noise ratio (PSNR) in dB between a ground truth image and its predicted reconstruction is used to evaluate the quality of the reconstructed images. This metric is defined in equation (3.1) between images A and B in terms of their mean squared error (MSE) and the maximum possible pixel value (MAX):

$$\begin{aligned} \mathrm{PSNR}(A,B) &= 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}(A,B)}\right) \\ &= 20 \cdot \log_{10}\mathrm{MAX} - 10 \cdot \log_{10}\mathrm{MSE}(A,B) \end{aligned} \tag{3.1}$$

3.2.6 Test Datasets

We chose three test datasets from scanned image acquisition techniques, which conventionally raster across the ROI without sampling. By sampling an entire dataset, we want to show how image complexity can inform the ideal sampling time-accuracy trade-off.

The first test dataset is the MIDAS dataset provided by the Air Force Research Laboratories' Materials Directorate [61]. It is the same dataset used in Challenge 4 of the AFRL additive manufacturing Modeling Challenge Series [62–64]. It consists of 900 consecutive slices of EBSD, optical micrographs, and backscatter electron (BSE) micrographs from an IN625 Ni-based superalloy sample. Data acquisition required 40 successive days and generated 3 TB, originally motivating the development of a novel dynamic sampling approach to EBSD. The second dataset is a compilation of 961 secondary electron (SE) micrographs of ultra-high carbon steel (UHCS) [65]. The primary micro-constituents were spheroidite, carbide networks, and pearlite. All micrographs were provided at 645×484 pixels, after removing 38-pixel tall SEM banners at the bottom of each micrograph. The third test dataset is 230 images of the nodal heart cells from a Sprague-Dawley rat acquired via confocal laser scanning microscopy (CLSM) [66]. These micrographs were the result of a 204.8 x 204.8 μm scan with 0.2 μm step size. All images were resized to 605 x 564 pixels with bilinear interpolation before synthetic sampling experiments.

3.2.7 Image Complexity

Several metrics have been developed to evaluate the complexity of images, which can inform the difficulty of their samplings and imputations. In this work, the complexity measure Q is compared with image histogram entropy [67]. Q is a function of the mean 2×2 windowed variance, V, and the image down-scaling factor S, as given in eq. (3.2). In eq. (3.3), s and v are the logarithm (Q is invariant of base choice) of S and V respectively.

$$Q = \frac{1}{s_{\max} - s_{\min}} \int_{s_{\min}}^{s_{\max}} \left[1 - \frac{1}{4} \left(\frac{\mathrm{d}v}{\mathrm{d}s} \right)^2 \right]_+ \mathrm{d}s \tag{3.2}$$

$$\frac{\mathrm{d}v}{\mathrm{d}s} = \frac{S}{V} \frac{\mathrm{d}V}{\mathrm{d}S} \tag{3.3}$$

For this implementation of Q, all images were rescaled with bilinear interpolation to have their shortest length set to 512 pixels, and the images were further incrementally down-scaled by factors of $\frac{1}{\sqrt[4]{2}}$ down to 8 pixels, so that every fourth rescaling corresponded to a halving: $512, 431, 362, 304, 256 \ldots 8$.

The image entropy based on the pixel intensity histogram is defined as follows:

$$H(X) = - \sum_{x \in X} p(x) \log p(x) \tag{3.4}$$

where X contains the allowed intensity values in the image. All calculations used 256 grayscale levels. Both measures of image complexity are compared as metrics to gauge sampling and imputation difficulty.

3.2.8 Microstructure Comparison

In order to test any sampling method a metric for comparing microstructures must be established, and we have selected H, the Hellinger metric [68]. For two discrete probability distributions P and Q, the Hellinger distance can also be defined in terms of the Bhattacharyya similarity measure, $\beta(P, Q)$ as is given by Eq. 3.5 and Eq. 3.6:

$$\beta(P,Q) = \sum_{i=1}^{N} \sqrt{P(i)Q(i)} \tag{3.5}$$

$$H(P,Q) = \sqrt{1 - \beta(P,Q)} \tag{3.6}$$

This distance has been identified as suitable for comparing microstructure-derived probability distributions such as feature volumes and the affine moment invariant Ω_3, describing both grain size and shape respectively [69, 70].

3.3 Results and Discussion

3.3.1 Example Samplings across Datasets

MIDAS Slice Dataset

The performance of k-NN-based dynamic sampling is examined for an example EBSD serial section of the MIDAS dataset. This task represents the best-case scenario, as compared to the subsequent non-EBSD examples. Due to additional pixel-wise indexing costs, EBSD permits greater latitude in what constitutes a viable sampling dwell time as compared to BSE or CLSM. Converting the 626×610 inverse pole figure (IPF) color map to grayscale permitted k-NN-based dynamic sampling. This is not an injective mapping from orientations, although collision (two adjacent grains sharing indistinguishably-close grayscale values) did not occur in the entire image. Orientation values were imputed using the nearest measurements during post-processing. The sampling was simulated offline, with the ground truth orientation data as the source for pixel values. The user parameters set were W=3, F=0.5, K=3, and S=8192. Fig. 3.1 shows the result of the dynamic sampling after 33% of the pixels were sampled. For such coarse piece-wise constant images, k-NN dynamic sampling densely measures boundaries as shown in (a), and the errors in (b) are found mostly along grain boundaries. At 1 in 3 pixels sampled, the IPF maps of the ground truth and imputed orientation values in (c,d) are visually indistinguishable. For this task, k-NN-based dynamic sampling required an average of 2.1 μs wall-time per pixel to choose each measurement. These measurements were selected with an average batch size of $10,000$ pixels, yielding a mean batch time of approximately 21 ms. A batched sampling approach aids application by permitting several milliseconds of delay per batch for microscope IO control.

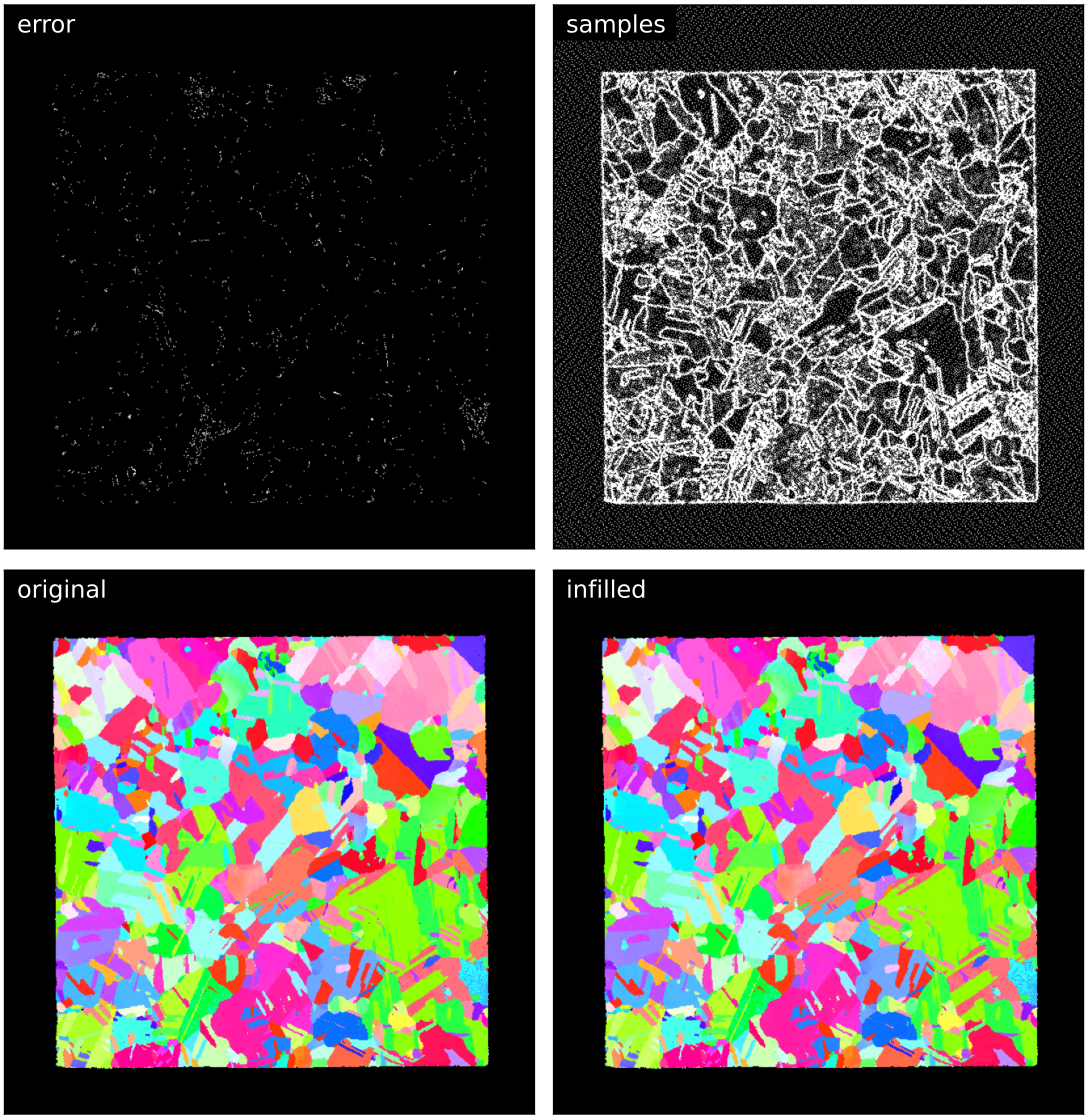

Figure 3.1: (top left) binary map indicating disorientation angle over 2° (top right) sampling locations (bot left) ground truth (bot right) nearest neighbor imputed orientation map (O_h IPF coloring)

UHCS Micrograph Dataset

Despite lacking the thick border that the MIDAS dataset images possess, the already single-channel UHCS dataset images were more easily sampled and imputed. Fig. 3.2 shows that at a slightly lower final sampling fraction of 25%, the algorithm is better able to reconstruct the example micrograph. Fig. 3.2 (b) shows a majority of the erroneous pixel values are found in missed micrograph features, one of which is circled in red. Upon close visual inspection, the rest of the error can be observed in the slightly blurred appearance of the micrograph in Fig. 3.2 (d). The user parameters were set to be W=3, F=0.5, K=3, and S=8192. Averaged over 100 runs, the algorithm runtime was 136 ± 2.75 ms for all 25% of the image or 1.74 μs per pixel. The averaged computation time per pixel is comparable to the dwell time used for imaging in the SEM.

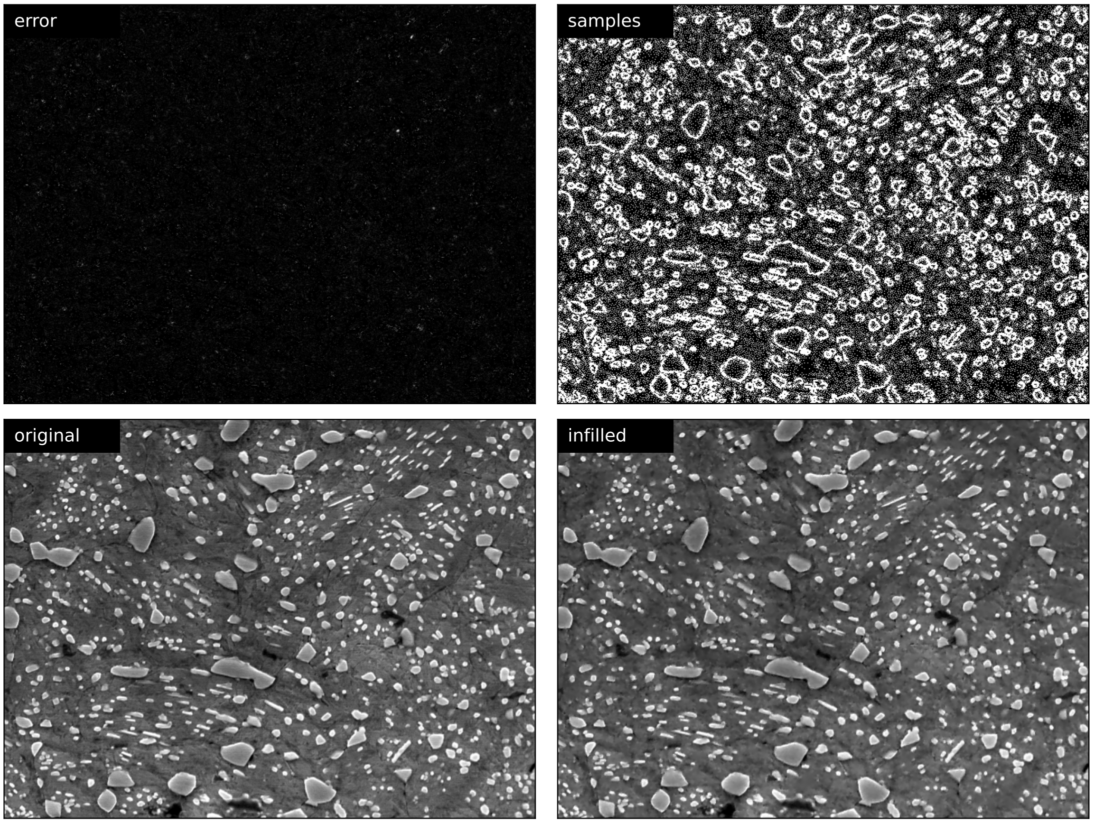

Figure 3.2: (top right) binary map indicating sampling locations (top left) raw map of the absolute difference between (bot left) ground truth and (bot right) imputed micrograph

CLSM Micrograph Dataset

For the CLSM dataset, the same user parameters were set to be W=3, F=0.5, K=3, and S=8192. Averaged over 100 runs, the algorithm runtime was similar at 108 ± 4.49 ms for all 25% of the image or 1.64 μs per pixel. As seen in Fig. 3.3 this micrograph sampling and imputation shows promising visual fidelity and a compatible runtime with the instrument at hand. According to its manual, the Zeiss laser scanning microscope (LSM) 5 Duo used to acquire these micrographs has a dwell time of 1.76 μs on a fast scan speed setting of 9, which is used to quickly adjust parameters before further imaging.

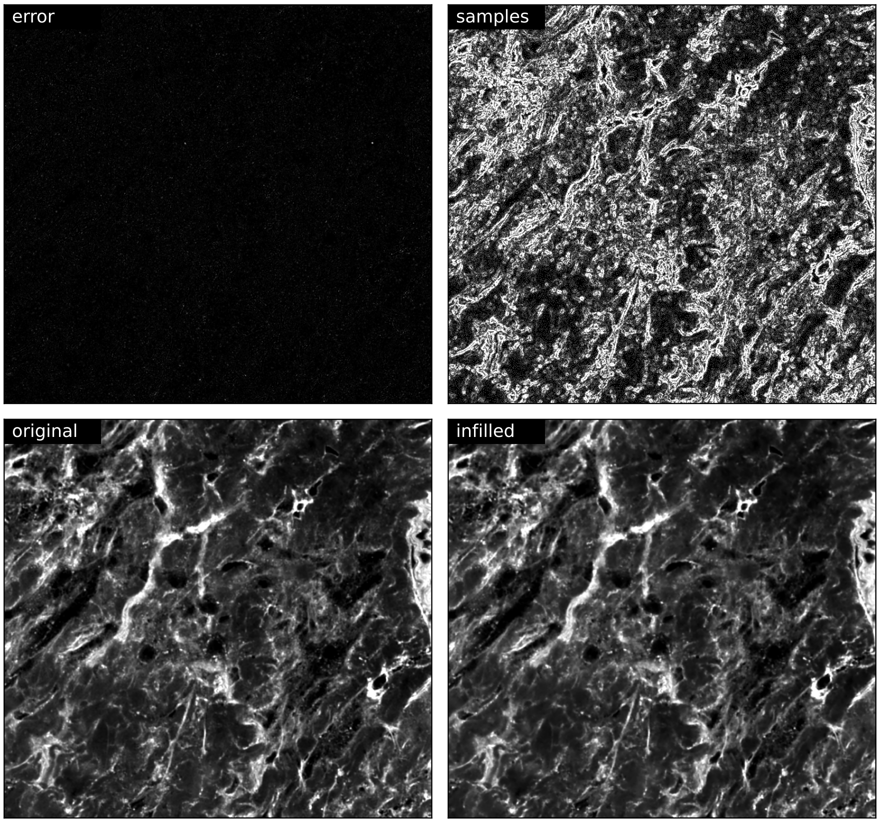

Figure 3.3: (top right) binary map indicating sampling locations (top left) raw map of the absolute difference between (bot left) ground truth and (bot right) imputed micrograph

3.3.2 Comparison with SLADS

Figure 3.4 shows the sampling locations chosen under two different parameters for the UDS. The acceptance factor of 0.01 is much more selective and requires more time to complete the sampling because each batch has fewer points. The end result of the sampling as measured by the PSNR in figure 3.5 shows that the proposed unsupervised dynamic sampling approach outperforms SLADS, barring 10% sampling, when a Sobol sequence is used to seed the initially measured batch of pixels. Overall, the PSNR differs by 2.0 at maximum between the three sampling runs, but UDS does so without any prior exposure training data, and purely based on a simple heuristic.

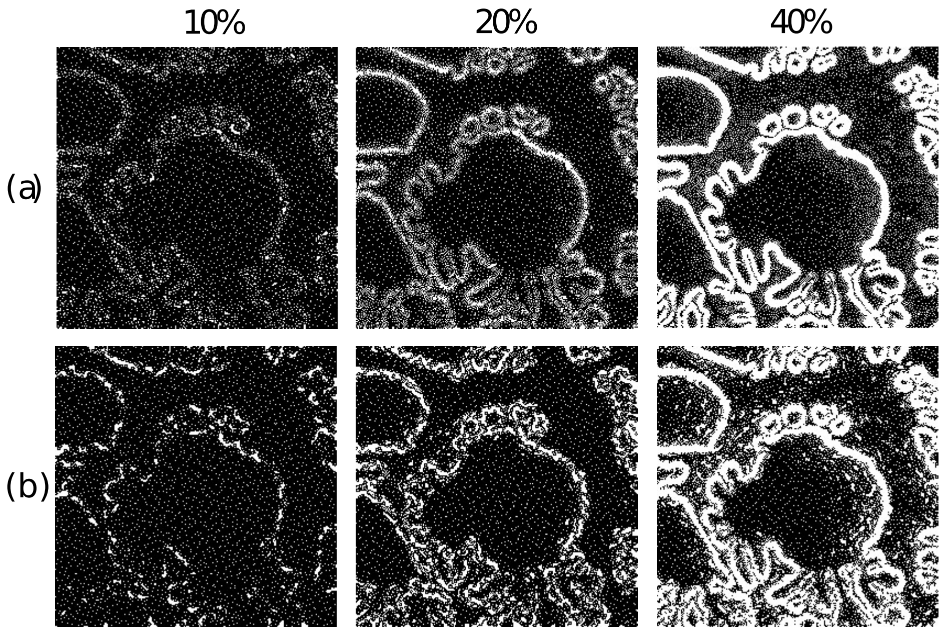

Figure 3.4: For a candidate acceptance factors of (a) 0.01 and (b) 0.1 the resultant sampling masks for 10%, 20%, and 40% when using UDS.

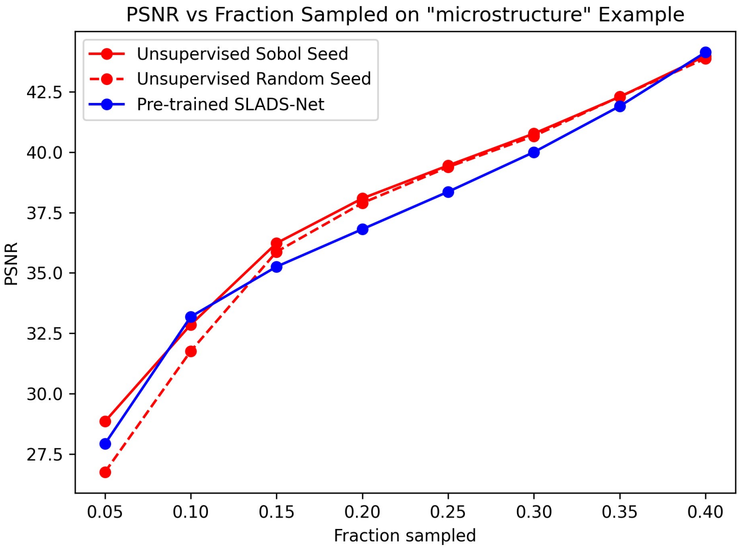

Figure 3.5: Direct comparison of the PSNR for reconstruction of the same example micrograph dubbed "microstructure" in reference [6].

MIDAS Volume

To further demonstrate the capabilities of UDS beyond application to 2D EBSD maps, UDS was applied to the entire MIDAS dataset volume. The dataset was trimmed to exclude pixels which only had BSE data, and to exclude incomplete slices. While the principles behind UDS can be extended to use 3D coordinates, we decided to apply UDS slice-by-slice because there is not a clear way to known the transformation which brings the coordinates from the previous slice into the reference frame of the next slice. In other words, running the dynamic sampling algorithm on a volume successfully registered with the complete dataset would misrepresent the current capabilities of UDS. For this simulated sampling, user parameters set were W=3, F=0.25, K=4, and S=0.05 (image fraction). The initial sampling was done with a 2D Sobol sequence. The individual EBSD slices were then infilled using nearest neighbor imputation. For this example the real component of the fundamental zone quaternion was use as the non-unique mapping of orientations to grayscale values. This process was also repeated for static Random and Sobol sampling.

Figure 3.8 shows the result of sampling to 20%, 30%, 40%, and 50%. The chosen locations indicated in the sampling masks demonstrate that UDS was able to quickly locate grain boundaries, and focus on exploring them. Few measurements were spent in data-lacking periphery regions where Kikuchi patterns were still acquired. Using DREAM3D [71], grains were segmented based on a misorientation threshold of 10 degrees and a minimum voxel count of 1000. These grains were then analyzed for their Ω_3 values and the equivalent sphere diameters (ESD) were calculated. Figure 3.7 and Figure 3.6 show the Hellinger distance between discretized versions of the Ω_3 distributions, and their corresponding empirical cumulative distribution functions (ECDF). Figure 3.6 also shows the grain size probability plots assuming a log-normal distribution of the ratio of ESD to mean ESD. The grain size distributions are essentially the same even with static random sampling at 20%. This implies that the grain size distributions might be easier to estimate from a sub-sampled volume than the Ω_3 distributions, although that might not extrapolate to other microstructures.

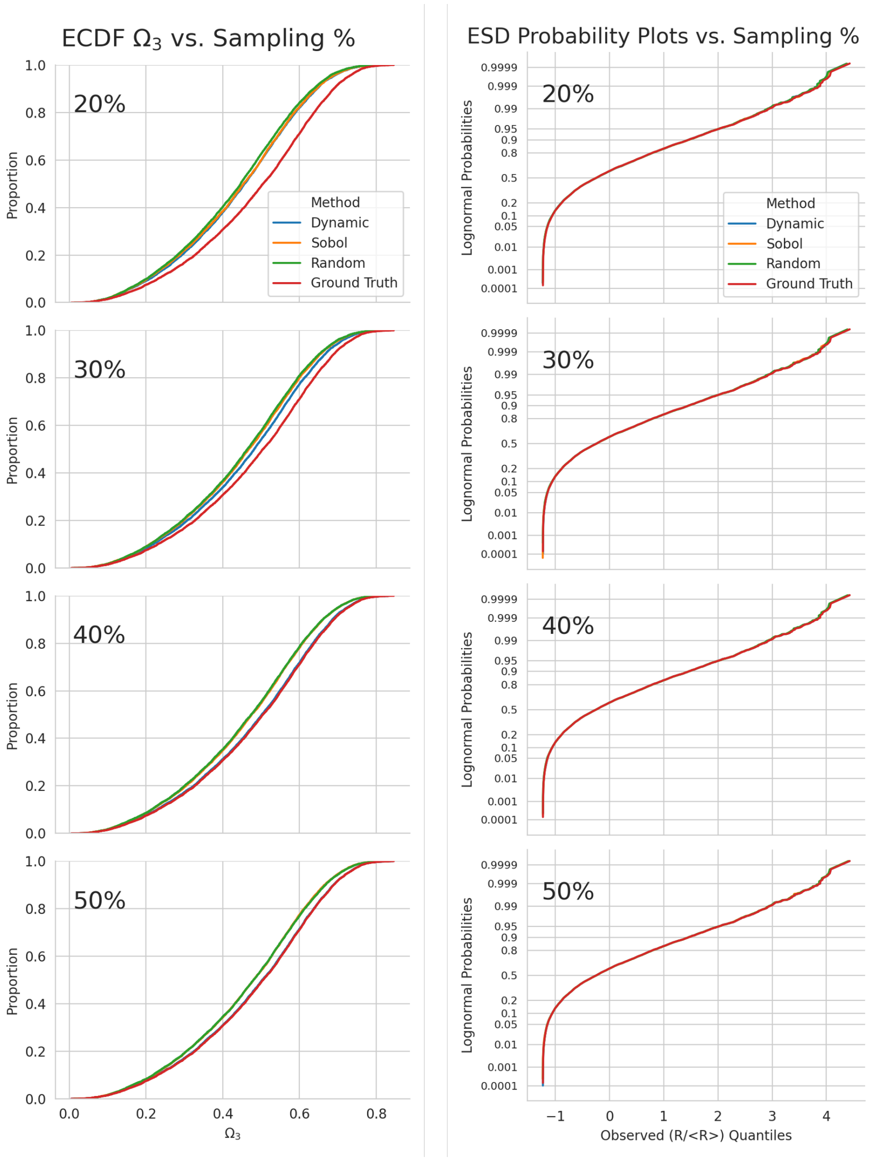

Figure 3.6: (a) dimensionless shape number Ω_3 empirical cumulative distribution functions compared over sampling approaches and sampling fraction (b) equivalent sphere diameter log-plots from the sample synthetic sampling experiment.

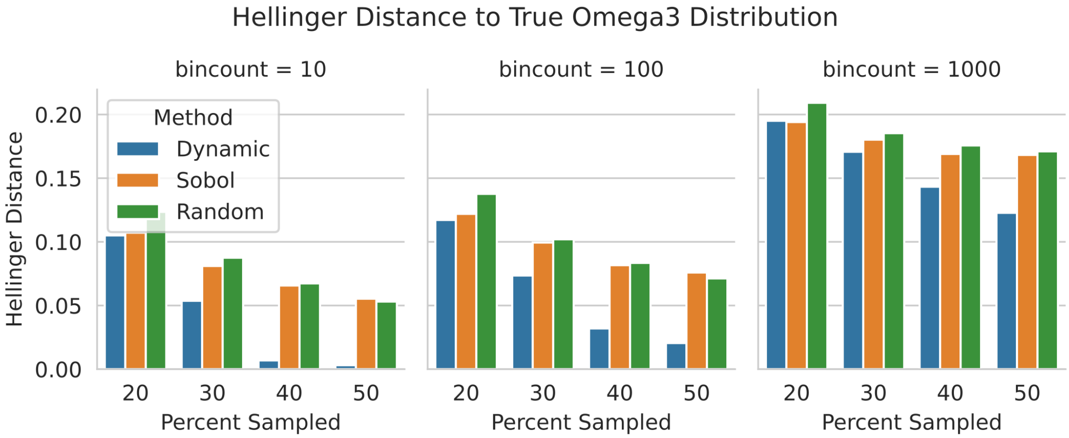

Figure 3.7: For uniform binning with 10, 100, and 1000 bins, the Hellinger distances are shown between the Omega3 distributions computed from sampling and the corresponding distribution from the original dataset.

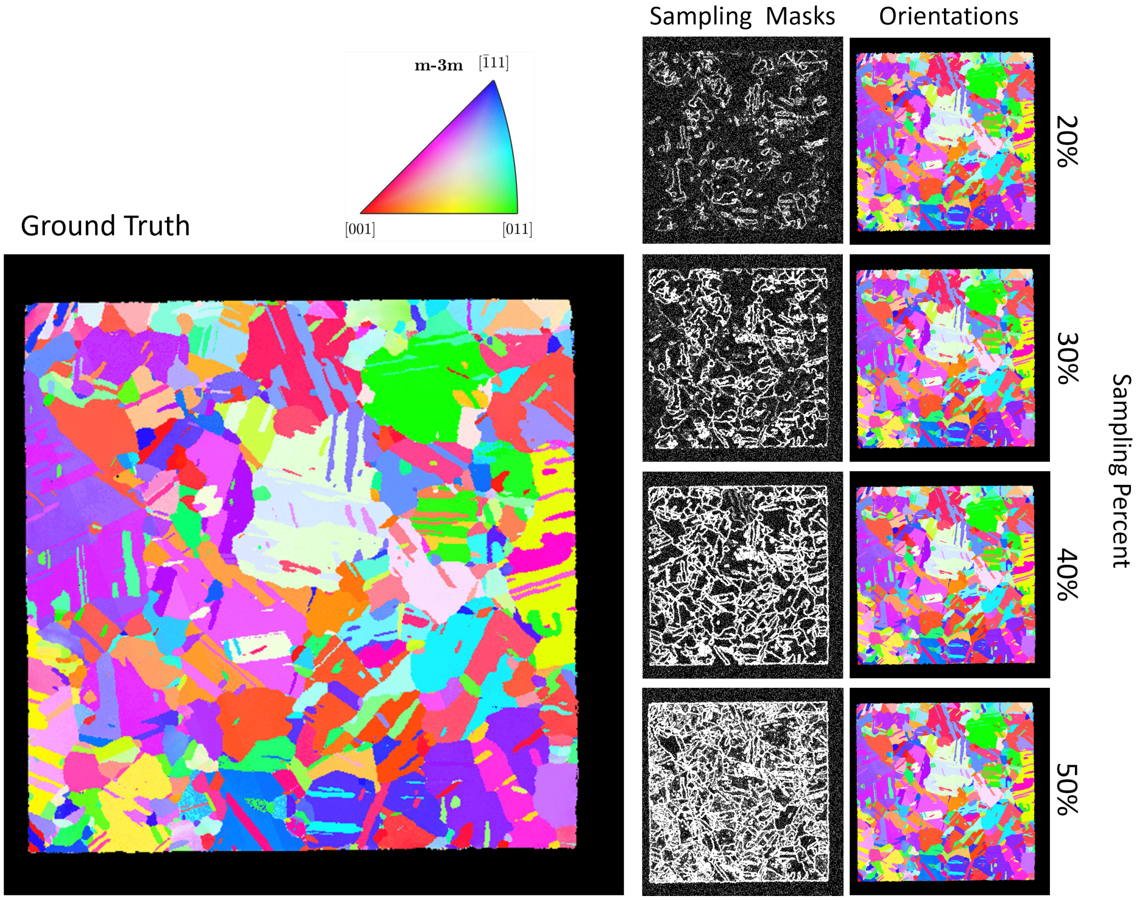

Figure 3.8: Reconstructions at various sampling percents under the proposed dynamic sampling algorithm of the medial slice in the 3D MIDAS dataset. The ground truth orientation map and masks of the sampling locations are also shown.

3.3.3 User Parameter Dependency

In order to explore the impact of various algorithm settings, a parametric study was conducted on an example BSE micrograph from the IN625 MIDAS dataset. The following subsections report on the importance of each of the user-configurable parameters.

Weighting, K-neighbors, Seed Size

All k-NN dynamic sampling methods perform better than a Sobol sequence sampling when 3% or more of the images had been observed, as seen in Fig. 3.9(a). Fig. 3.9(b) shows that the inverse squared distance weighted mean infilling is consistently better than unweighted mean infilling across all values of K except for a few ranges of percent pixels sampled. For this reason, a weighted mean infilling is utilized henceforth. When comparing Fig. 3.9 (c) and (d), the number of neighbors K is pivotal to sampling performance at both extremes of F and W. However, comparing Fig. 3.9(e) and (f), the initial Sobol seed size has little impact on the sampling performance, except at excessive sampling rates above 40%.

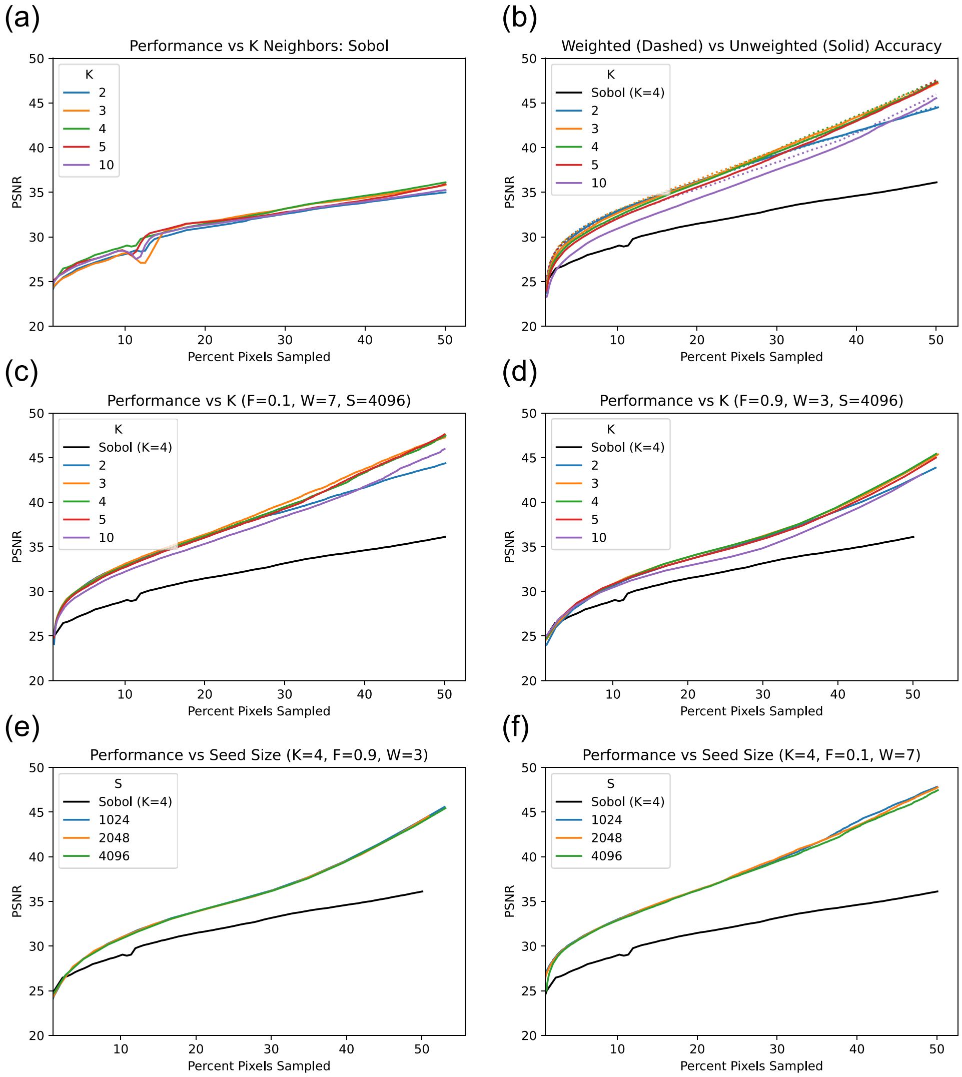

Figure 3.9: Percent of pixels sampled up to 50.0% vs. PSNR plotted across K number of nearest neighbors and weighting.

Acceptance Fraction and Window Size

The impact of window size W appears to be contingent on the value of F, and vice-versa. Sufficient rejection of poorly scoring local optima, whether by accepting fewer optima or by broadening the optima window, leads to high PSNR. Comparing Fig. 3.10 column (a,c,e) with column (b,d,f), we see that both acceptance fractions F and W are similar in their impact on dynamic sampling performance. Whether the window size is 3×3 or 7×7, if F is set sufficiently low as in image (b), the sampling performance is identical.

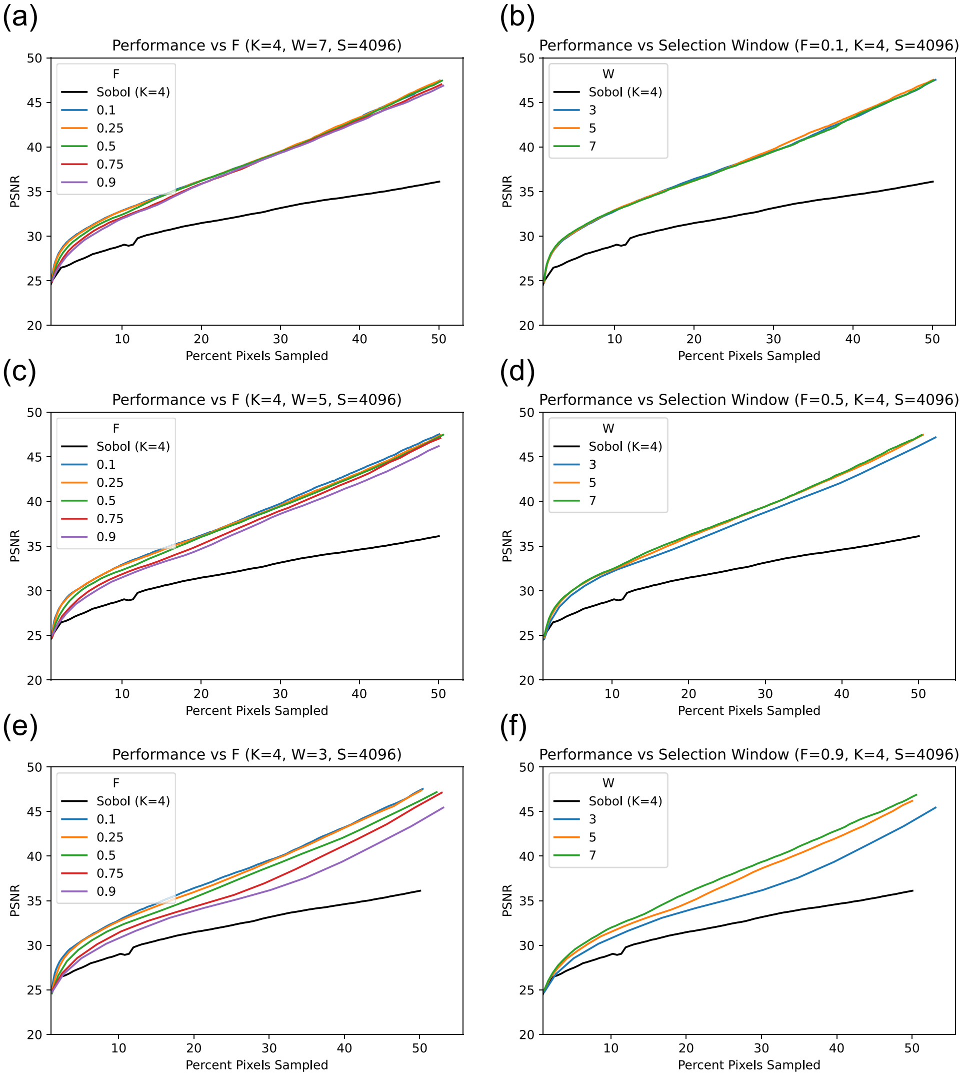

Figure 3.10: Percent of pixels sampled up to 50.0% vs PSNR plotted across several user parameters

3.3.4 Synthetic Runtime Results

Run-time results were studied across values of W and F. Fig. 3.11 shows heat-maps corresponding to (a) runtime required to sample 30%, and (b) 30% reconstruction PSNR. Given $K = 3$ and $S = 4096$ determined from previously shown analysis, parameter settings between $F = 0.25$ and $F = 0.5$, along with $W = 3$ were found to balance reconstruction fidelity and runtime. These are thus chosen as the default parameter values for the sampling algorithm.

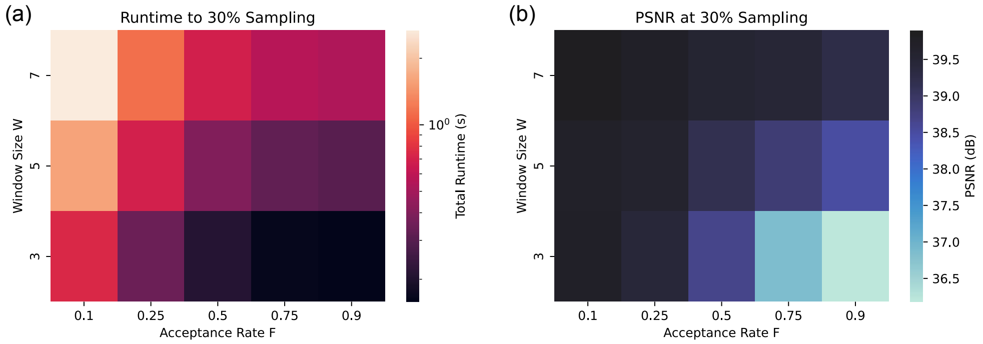

Figure 3.11: Across F and W, heatmaps of (a) time to 30% (log-scale legend) (b) PSNR at 30% sampled

3.3.5 Sampling Performance vs. Image Complexity

To extend these results, the algorithm's performance was evaluated on the other two aforementioned datasets. These images are naturally heterogeneous in visual appearance, as they include varied steel micrographs and images from a 3D mouse heart cell scan. After sampling all images in the dataset utilizing parameters $W = 3$, $F = 0.3$, $S = 4096$, and $K = 3$, the resulting mean image complexity is plotted in Fig. 3.12 using image entropy and Q score previously defined. Each discrete heatmap box shows the mean image complexity that led to the corresponding reconstruction PSNR at a given percent of pixels sampled. Both image pixel histogram nats (natural logarithm analogue of base 2 bits) in (a) and Q score in (b) indicate that more complex images are more difficult to sample and impute. However, nats in (a) show a noisier relationship with image complexity, but the contrast is maintained across the range of image complexities. Complexity score Q is more binary over these two datasets but lacks clear outliers at 2-3 % sampling and 8-10 PSNR.

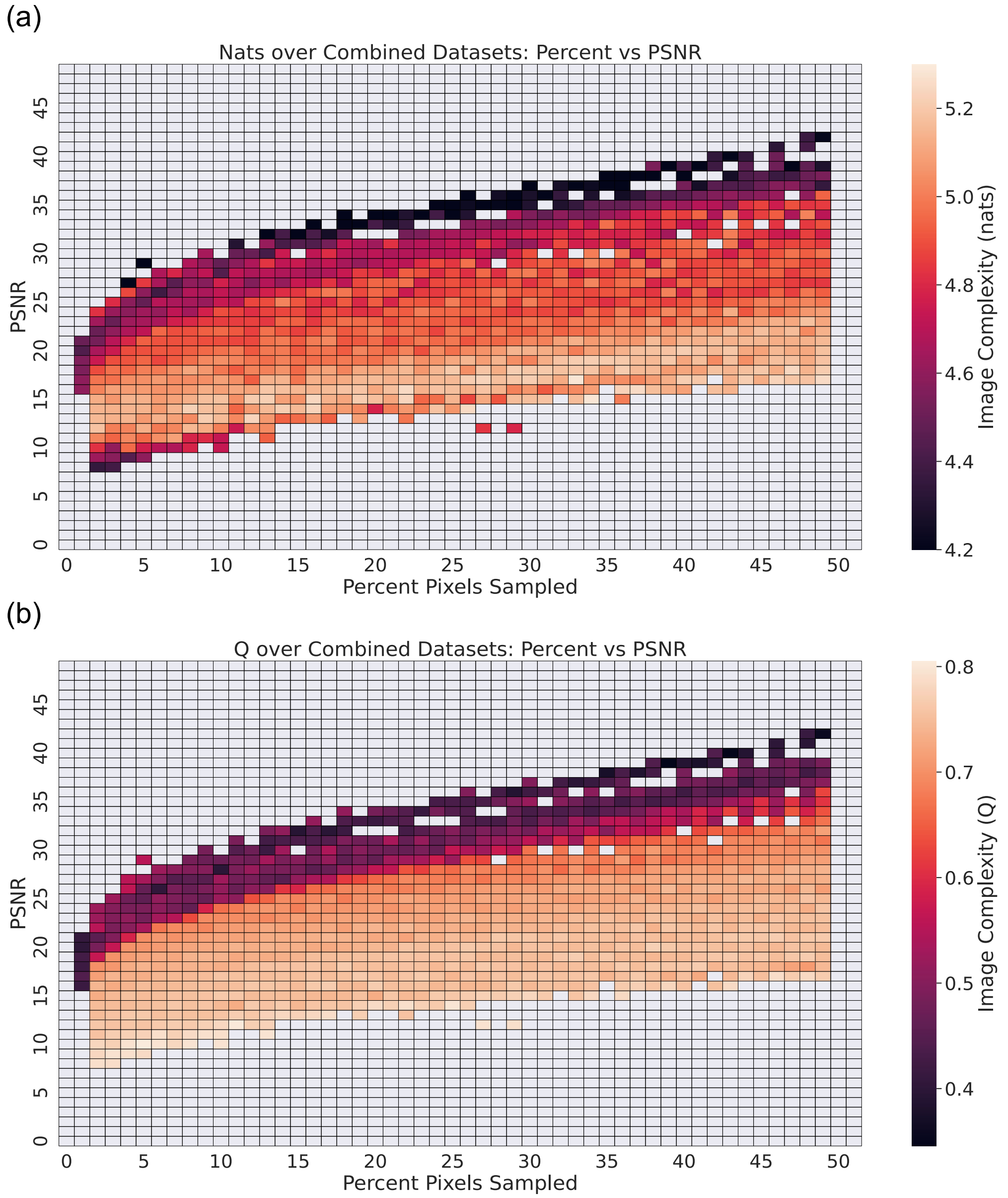

Figure 3.12: Heatmap showing the average image complexity that led to a given PSNR at each percent sampled. Only mean values with more than 3 contributing data-points appear in the heatmap. UHCS and CLSM datasets are combined for both plots.

3.4 Conclusion and Summary

In summary, we have developed a UDS algorithm for 2D images which leverages the parallel processing power of the GPU. As this algorithm requires no prior training, it can be utilized for an arbitrarily shaped ROI. The current state of the art, pretrained SLADS-Net, operates approximately 1000 times slower and produces imputed micrographs with slightly lower PSNR than UDS. The provided Python implementation is easily adoptable and requires minimal to no tuning of default user parameters. We have analyzed its performance as a function of two different image complexity measures across three different datasets. When executed on modest (laptop) hardware, the algorithm implementation runs faster than respective measurement acquisition times in a variety of imaging modalities. By using the closest measured neighbor alone, multi-channel data such as orientations can be imputed. For real experimental orientation data, measuring 1 in 3 voxels results in visually indistinguishable IPF maps. In the future, we aim to decrease this required sampling fraction with the incorporation of information from previous serial sections and the utilization of less costly modalities to further inform dynamic sampling in EBSD.

For EBSD applications in particular, this work has assumed that indexing can be performed in real time as measurements are gathered. This may be true with Hough or Radon transformation based approaches; however, dictionary indexing might not be able to be performed in real time. A large hurdle to turning this slice-by-slice sampling into a truly 3D sampling technique is establishing a practical method to realign the sample surface to the same grid locations of the previous slice. Besides this, there are several other improvements planned for UDS. The first planned change for this algorithm is an automatic stopping condition. By periodically comparing the incremental changes in micrograph reconstruction as sampling proceeds, a threshold in PSNR or mean disorientation angle could be used to determine if sufficient sampling has been achieved. Further, electron microscope scan controls often allow arbitrary floating point precision in beam control across the image plane. By using non-discretized coordinates, reconstruction accuracy may improve sufficiently to justify the associated sampling runtime of not caching possible grid queries. Thirdly, an adaptive dwell time, where possible, could allow increased sampling speed and reconstruction accuracy. By briefly, but noisily, checking the center of undiscovered regions, small image features might be more easily discovered.

CHAPTER IV

PCA Dictionary Indexing

4.1 Introduction

Over the past three decades, Electron BackScatter Diffraction (EBSD) has established itself as an indispensable tool for polycrystal microstructure characterization. By rastering the electron beam across a region of interest (ROI) on a polished sample, and matching each resultant diffraction pattern to a crystalline lattice state, rich data sets can be produced which map crystalline orientation, phase, stress, and dislocation concentrations across the ROI. Efforts to automate and enhance the EBSD indexing accuracy and efficiency have been numerous. A first approach to automating the orientation indexing, Hough indexing, uses its namesake transform (which produces peaks from lines/bands in images) along with a voting scheme to automatically determine the Miller indices of diffraction bands in electron backscatter patterns (EBSPs) [72–74]. Thereafter, researchers developed physics-based forward models of electron diffraction to simulate the expected backscatter electron intensity as a function of the lattice state [75, 76]. The simulation of electron diffraction for a given accelerating voltage, sample geometry, and crystalline cell, generates a reference signal on the 2-sphere (referred to here as the Kikuchi sphere or K-sphere). With this tool in hand, two approaches to orientation indexing have been subsequently developed: dictionary indexing and spherical indexing [38, 77].

By iterating over a uniform sampling of the given crystallographic orientation fundamental zone (FZ), conventional dictionary indexing orients and then projects the K-sphere onto a virtual camera sensor, to create a dictionary of possible idealized detector observations. Orientations are determined by procuring the closest simulated pattern to each experimental pattern; normalized cross correlation or pattern dot products are commonly used as matching metrics. Dictionary indexing offers superb pattern noise robustness as compared to traditional Hough based methods, but this comes at a dramatically increased computational cost from pattern projection and matching [78, 79]. For orientation FZ sampling, a popular default target mean neighbor disorientation of 1.4° requires $330,000$ orientations for high symmetry (cubic) materials such as Nickel; however, in the absence of symmetry, save for inversion symmetry which follows from the application of Friedel's Law to the diffraction process, the same 1.4° target threshold requires just over 8 million sample orientations [80].

To address this low symmetry problem, a second approach leveraging physics-based forward models, known as spherical indexing, introduced a complementary viewpoint to EBSD orientation indexing. Therein, experimental EBSPs are instead back projected onto the 2-sphere, and then a cross correlation over $SO(3)$ is performed with the reference K-sphere; the maximum in the cross-correlation is then associated with the desired orientation. In order to dramatically speedup computations, the cross correlation is calculated in the frequency domain via harmonics on $SO(3)$ computed with real-valued Fast Fourier Transforms explicitly leveraging systematic zeros due to the crystallographic K-sphere symmetry. Without needing a dictionary of potentially millions of diffraction patterns, spherical indexing has similarly improved noise robustness over Hough indexing as individual diffraction bands need not be identified, although indexing speed can suffer when an increased band-limit, dictating the number of harmonics to be used, is required for noisy patterns [38]. This band-limit is used to control the speed-accuracy trade-off to address the needs of each experimental dataset.

The present paper proposes and analyzes a modification to conventional dictionary indexing which provides a control parameter, the number of PCA components utilized, akin to the band-limit in spherical harmonic indexing, while maintaining the computational simplicity and parallelisms arising from matching on the detector plane. Specifically, we project both the dictionary and experimental patterns along the first p principal components of the entire simulated EBSD dictionary. The resulting lower dimensional projections can then be matched as is done in conventional dictionary indexing. The two important findings of our work are 1) PCA compression marginally, and counter-intuitively, improves the indexing accuracy (and speed) via noise suppression as compared to conventional pattern matching which uses all pattern pixels; and 2) neither half precision nor dynamic 8-bit integer quantization modify the indexing results on a real world dataset.

4.2 Methods

4.2.1 PCA Calculation

PCA is a statistical analysis method used to reduce the dimensionality of data sets through a low-rank linear approximation that maximizes the captured variance [81]. The principal components in "principal component analysis" are formulated in two different ways here: direct Singular Value Decomposition (SVD) of the centered data matrix or Eigenvalue Decomposition (EVD) of the data covariance matrix. Let $\mathbf{X}$ be a data matrix with zero column means of shape $n \times d$ where n represents the number of samples and d denotes the number of features. In the context of pattern matching, n is the number of EBSPs, and d is the number of pixels in each flattened pattern. The SVD of $\mathbf{X}$ is given by (4.1):

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{4.1}$$

where $\mathbf{U}$ is an orthogonal matrix $n \times d$ whose columns are called left singular vectors. The columns of the $d \times d$ matrix $\mathbf{V}^T$ are the right singular vectors. The diagonal matrix $\mathbf{\Sigma}$ has entries corresponding to the singular values. In PCA, a user-specified number of components, of right singular vectors with the highest singular values, are used as the new basis of the data. SVD factorization algorithms that can be used for PCA are an area of active research [82]. PCA via SVD directly on $\mathbf{X}$ is employed in the default implementation of PCA in many public libraries, often among many other available algorithms.

Because the principal components of $\mathbf{X}$ are not necessarily integer vectors even if $\mathbf{X}$ only contains integers, floating-point representations are required to compute them. In EBSD, low-bit-depth representations like unsigned 8 or 16-bit integers are the usual space-saving data storage formats for EBSPs. When cast as 32-bit floating point numbers, the entire EBSD dictionary data set matrix $\mathbf{X}$, can easily exceed the available computer memory, especially for low symmetry materials. Therefore, an alternative incremental approach is desirable, one that avoids storing $\mathbf{X}$ in memory all at once.

To this end, the sample covariance matrix, given in (4.2), can be constructed by row, column, or by block when appropriately iterating over $\mathbf{X}$.

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \tag{4.2}$$

And further, to find the principal components of $\mathbf{X}$, the singular vectors and singular values of the SVD of $\mathbf{X}$ can be acquired through the eigenvectors and eigenvalues of $\mathbf{C}$. The covariance matrix $\mathbf{C}$ can be factored according to (4.3):

$$\mathbf{C} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T \tag{4.3}$$

where $\mathbf{Q}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{C}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix containing eigenvalues. Substituting the SVD of $\mathbf{X}$ from (4.1) into (4.2):

$$\begin{aligned}
\mathbf{C} &= \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \\
\mathbf{C} &= \frac{1}{n-1}(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T)^T(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) \\
\mathbf{C} &= \frac{1}{n-1}\mathbf{V}\boldsymbol{\Sigma}^T\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \\
\mathbf{C} &= \frac{1}{n-1}\mathbf{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\mathbf{V}^T
\end{aligned} \tag{4.4}$$

When comparing (4.3) to (4.4), one observes that by a factor of $\frac{1}{n-1}$, $(\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})$ is the eigenvalue matrix of $(\mathbf{A}^T\mathbf{A})$, meaning each square singular value σ^2 is the eigenvalue $\lambda(\mathbf{A}^T\mathbf{A})$. See "Introduction to Linear Algebra" by Strang for a complete proof of the equivalent PCA formulation [83].

We opt to use a generalized batched version of a waterfall method called Welford's online algorithm for variance and covariance [84–86]. This exact streamed algorithm facilitates a single pass covariance matrix calculation for pattern dictionaries that would not fit in memory. Although at higher pattern resolutions, such as 256×256, the covariance matrix itself stored in double precision is 64×256^4 bits (34.3 GB), and at 512×512 resolution it is 550 GB in size. This prohibitive memory cost can necessitate either pattern binning to a lower resolution, or randomized PCA methods which skip covariance matrix calculation, such as that of Halko et. al [87], and it is already available in PyTorch under the name ”pca_lowrank”. Another notable alternative being considered which ties PCA to the Schrödinger equation is Algorithm 2 in Liu et. al in [88].

4.2.2 K-Nearest Neighbors and Dynamic Numerical Quantization

Once the dictionary's PCA components are computed, and both the dictionary patterns and experimental patterns are centered and projected, all that remains is to search for the closest dictionary projection entries for each experimental projection. Each dictionary pattern (or its PCA loadings) called x_i and each experimental query pattern (or loading) called q_i compose d dimensional rows of the dictionary X and the queries Q, respectively. The dimensionality of the data points, d, is either the number of pixels in each pattern or the chosen number of principal components. The results of a KNN query are denoted as $\text{KNN}_k(X, q_i)$, where k signifies the number of nearest neighbors to be retrieved, as defined in equation 4.5 below:

$$\text{KNN}_k(X, q_i) = \text{argmin}_{S_k} \left\{ \text{DIST}(x_i, q_i) \,\middle|\, x_i \in X, |S_k| = k \right\}, \tag{4.5}$$

where $\text{DIST}(x_i, q_i)$ can be any number of metrics such as a Minkowski distance given in equation 4.6:

$$\text{dist}(x_i, q_i) = \left(\sum_{j=1}^{d} |x_{ij} - q_{ij}|^p \right)^{1/p}. \tag{4.6}$$

We opt to use a dot product based metric for pattern and PCA loading distances due to its computational simplicity as seen in equation 4.7:

$$\text{dist}(x_i, q_i) = -\sum_{j=1}^{d} x_{ij} q_{ij}. \tag{4.7}$$

There are many approaches to accelerate the KNN search task by approximation algorithms, and they are broadly divided into locality sensitive hash functions (LSH), Product Quantization (PQ), and graph based methods [89]. Treatment of especially high dimensional datasets (several hundred dimensions), as is the case with the EBSD KNN problem, has been a recent focus area in the literature [90].

It is assumed that the structure of the EBSD dictionary is that of the given crystallographic fundamental zone embedded into image space. Grids on this embedded manifold might not lend themselves well to clustering or graph traversal due to the “curse of dimensionality” combined with the fact that incoming queries (experimental patterns or their PCA loadings) are often noisy and out of distribution (far off of the ideal dictionary manifold) compared to the dictionary patterns or their PCA loadings [91, 92]. To maintain the noise robustness of full dictionary indexing, for now we avoid using approximate lookup algorithms, and plan to explore their application in the future, both on the patterns themselves, and on their PCA loadings.

Instead, we focus on quantization methods, opting to simply use lowered numerical precision in the form of 16-bit floats and 8-bit integers to improve the indexing performance. This has been a recent area of development for accelerating both the training and inference of neural networks [93]. By leveraging the FBGEMM and QNNPACK packages (on AMD64 and mobile processors respectively), we can dynamically transform the data before casting to 8-bit integers, and then apply the appropriate inverse transform to closely estimate each full 32-bit float dot product [94, 95]. The eight x86 Intel CPU cores used in this study leveraged FBGEMM via PyTorch. Recent results using 2 bit quantization via lookup tables have further accelerated operations, and we are considering exploring this technique as well [96].

4.3 Test Dataset

The test dataset is a series of Nickel EBSD scans of the same region of interest taken at camera gain levels elevated by $0, 3, 6, 9, 12, 17, 20, 22$, and 24 dB respectively, shown in Figure 4.1, and referred to here as scans 1 through 10 [97]. We benchmark on scans $1, 5$, and 10 for low, medium and high noise levels. The patterns were masked with a disk that included $2,819$ of the $3,600$ pattern pixels. This dataset is used to exhibit the performance of PCA-based dictionary compression in the face of increasing noise levels, and to demonstrate identical indexing results regardless of reduced precision distance calculations. From table 1 in the original publication, patterns within scans 5 and 10, required just 0.95 ms and

0.25 ms of the exposure time as compared to 3.50 ms for scan 1 [98]. In this context, if indexing can be performed reliably at 0.95 ms or 0.25 ms exposure, a relative speedup of approximately 4× or 14×, respectively, could be achieved.

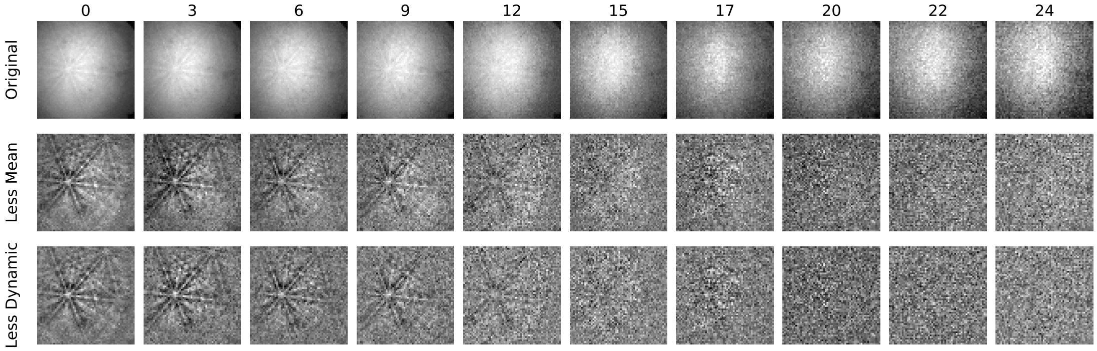

Figure 4.1: Example patterns from the (4,4) point on the region of interest in the camera gain dataset. The camera gain (in dB) is over each column. The middle and bottom rows show static and dynamic background subtraction preprocessing steps.

4.4 Results and Discussion

4.4.1 PCA Components

Figure 4.2 shows example eigenvectors resulting from eigen-decomposition of the Nickel EBSD dictionary at increasing dictionary size. The principal components stabilize once the dictionary reaches 100, 000 entries. For reference, Figure 4.3 shows the result of the same process (identical projection center, and Laue group) carried out with a faux master pattern consisting of a checkerboard. Both eigen-decompositions have azimuthal oscillations in the primary component radiating around the projection center in the upper left quadrant of the detector plane. Generally, the directions appear visually similar, indicating that the pixel intensity covariance matrix may be more heavily dependent on the symmetry at hand, and the experimental geometry, rather than the specific appearance of the Kikuchi bands, or more generally the signal on the 2-sphere.

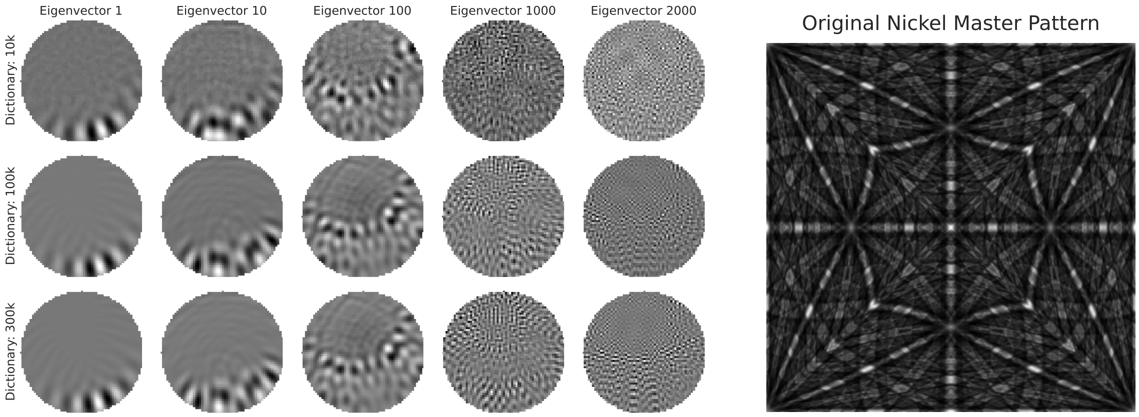

Figure 4.2: Representative principal components from the projection of the Nickel master pattern onto the virtual detector plane. Lower rows use a larger sampling of the orientation fundamental zone.

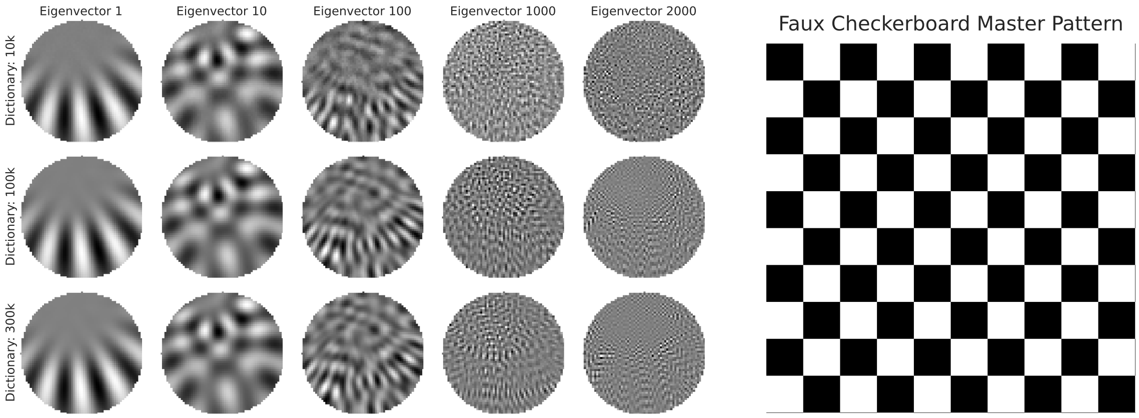

Figure 4.3: Representative principal components from the projection of a faux checkerboard master pattern onto the virtual detector plane. Lower rows use a larger sampling of the orientation fundamental zone.

4.4.2 Pattern Matching Speed and Hardware Utilization

Tables 4.1 - 4.3 show the indexing speed and compute utilization across dictionary size and indexing mode, with increasing numeric precision. Each reported value is the average across Scan 1, 5, and 10, as the performance is independent of the data content. Table 4.1 illustrates the EBSD indexing performance when using FBGEMM on 8 Intel Skylake CPU cores (on Google Cloud) at INT8 precision. Patterns per second (pps) and Trillions of Operations Per Second (TOPS) are provided for different dictionary sizes (100k, 200k, and 300k). A clear trend of decreasing pps as the number of FZ samples increases is observed, indicating increased computational load. Further, the TOPS tend to decrease with a reduction in the dimension of the entries, indicating a relative under-utilization of the hardware for smaller dimensions. Overall, the results show that PCA with a compressed representation (e.g., 500 PCA loadings instead of 2,819 pixels) significantly improves the indexing performance by a factor of approximately 4. Similar trends are observed in the Nvidia benchmarks at both half and single precision. At half precision, when 500 PCA components were used with a reduced dictionary size of 100k patterns, over 70,000 pps throughput was observed. For reference, a more powerful Nvidia 4090 achieved 35,000 throughput using the Radon transform [99]. The most accuracy-focused setting tested with a 300k pattern dictionary and 1,500 PCA components yielded around 12,000 pps at half precision on a modest GPU. While this increased GPU throughput is useful for very large datasets and real time indexing applications, a corresponding identical matching can be achieved with 320 pps on a consumer grade CPU. An accuracy-focused bandwidth of 158 for spherical indexing that is comparable to the these PCA results on the CPU was benchmarked at around 50 pps on a 24 core workstation [38]. Overall, the hardware utilization indicates that there is some room for improvement. The theoretical FP32 throughput for the Nvidia T4 is 8.1 TOPS, and our results are around $2-4$ TOPS. For FP16, the gap widens to a 65 TOPS theoretical limit and actual performance of $7-15$ TOPS.

FZ Samples	100k	200k	300k	100k	200k	300k
	Patterns / Second			TOPS		
Full 2819	530	271	191	0.383	0.393	0.413
PCA 1500	880	457	334	0.265	0.276	0.300
PCA 1000	1287	639	454	0.258	0.257	0.272
PCA 500	2060	1001	719	0.206	0.201	0.215

Table 4.1: PPS and TOPS on 8 CPU cores using INT8 quantization.

FZ Samples	100k	200k	300k	100k	200k	300k
	Patterns / Second			TOPS		
Full 2819	18 991	9741	6537	13.7	14.1	14.1
PCA 1500	35 502	18 044	12 130	10.7	10.9	10.9
PCA 1000	55 485	27 441	18 979	11.1	11.0	11.4
PCA 500	72 417	36 988	24 498	7.3	7.4	7.3

Table 4.2: PPS and TOPS on a Nvidia T4 using FP16 precision.

FZ Samples	100k	200k	300k	100k	200k	300k
	Patterns / Second			TOPS		
Full 2819	5819	2802	1867	4.2	4.1	4.0
PCA 1500	9594	4844	3372	2.9	2.9	3.0
PCA 1000	14 137	7017	4859	2.8	2.8	2.9
PCA 500	23 547	11 563	7926	2.4	2.3	2.4

Table 4.3: PPS and TOPS on a Nvidia T4 using FP32 precision.

4.4.3 Disorientation Distributions

Figures 4.4 - 4.7 plot disorientation angle Empirical Cumulative Distribution Functions (ECDFs) of disorientations with respect to a reference indexing with a dictionary of 1 million patterns on Scan 1. Figure 4.4 clearly shows that neither reduced precision with INT8 quantization nor with FP16 changed the indexing outcome, even at the maximum noise level scan. This outcome indicates that dictionary based EBSD indexing could be a viable approach for low symmetry materials, as 8 million entries of 1500 PCA components occupies 24 GB at half precision which can fit entirely into memory on some newer GPUs. Further research is required to find the minimum bit-depth representation possible for dictionary indexing on EBSD patterns or loadings, and it is possible that this floor is dependent on the pattern noise level. For relatively noise free patterns, it may be sufficient to treat each

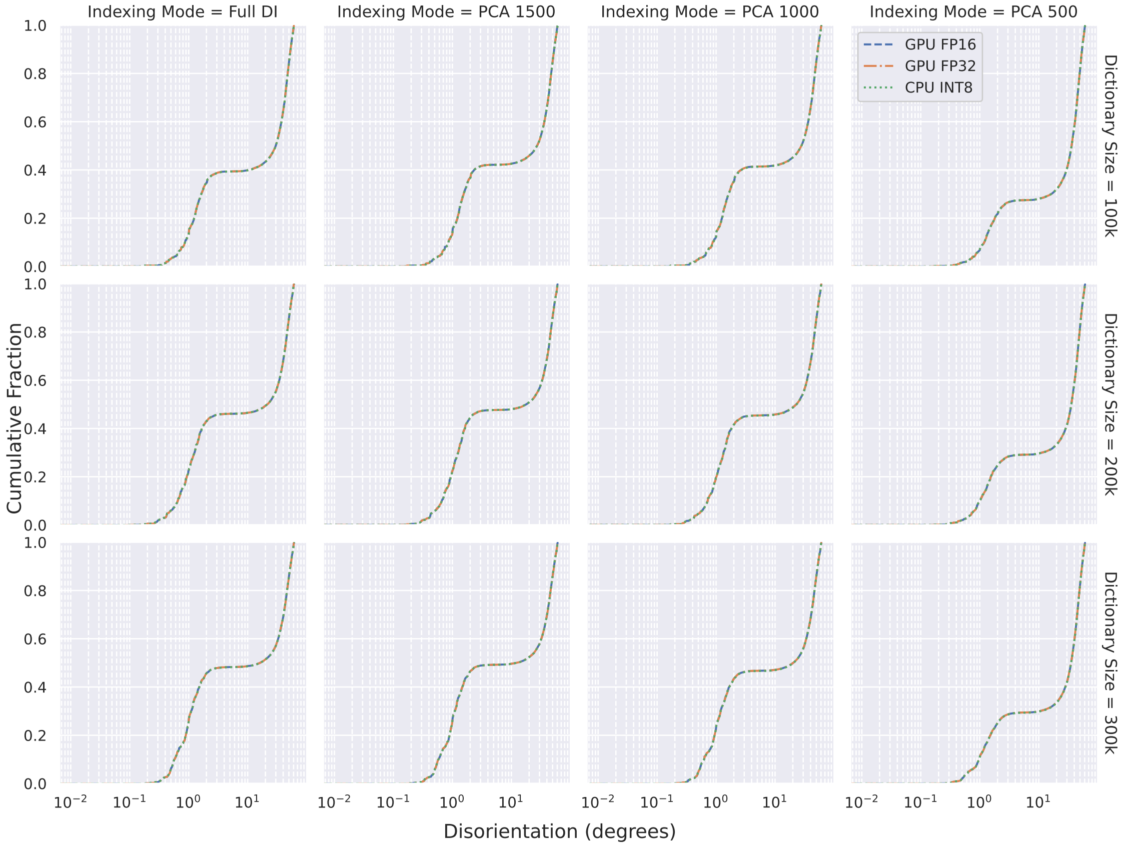

Figure 4.4: A grid of disorientation angle ECDFs across indexing mode and dictionary size for Scan 10 alone. Each plot line is colored and styled according to the numeric precision and compute device utilized.

pixel or PCA loading as a single bit indicating the sign.

Figure 4.5 shows that a sufficient dictionary size is required to achieve accuracy on the order of 2°. In our analysis we have assumed that either gradient based optimization (as is done in spherical indexing), or a pattern search (name of an optimization scheme) refinement will follow the initial indexing. Local optimization in orientation space is much less demanding than the initial indexing. Thus, the height of the plateau region in the ECDF indicates the likely final indexing success (approximately 99%, 90%, and 50% on Scans 1, 5, and 10). Figure 4.6 shows the relative deterioration of the indexing quality moving from scan 1 to 10. For each plot the indexing success rate significantly drops as the noise level increases.

Lastly, Figure 4.7 shows that using PCA compression has improved the disorientation ECDFs when using 1500 components. Not shown in the plot, using the full rank PCA

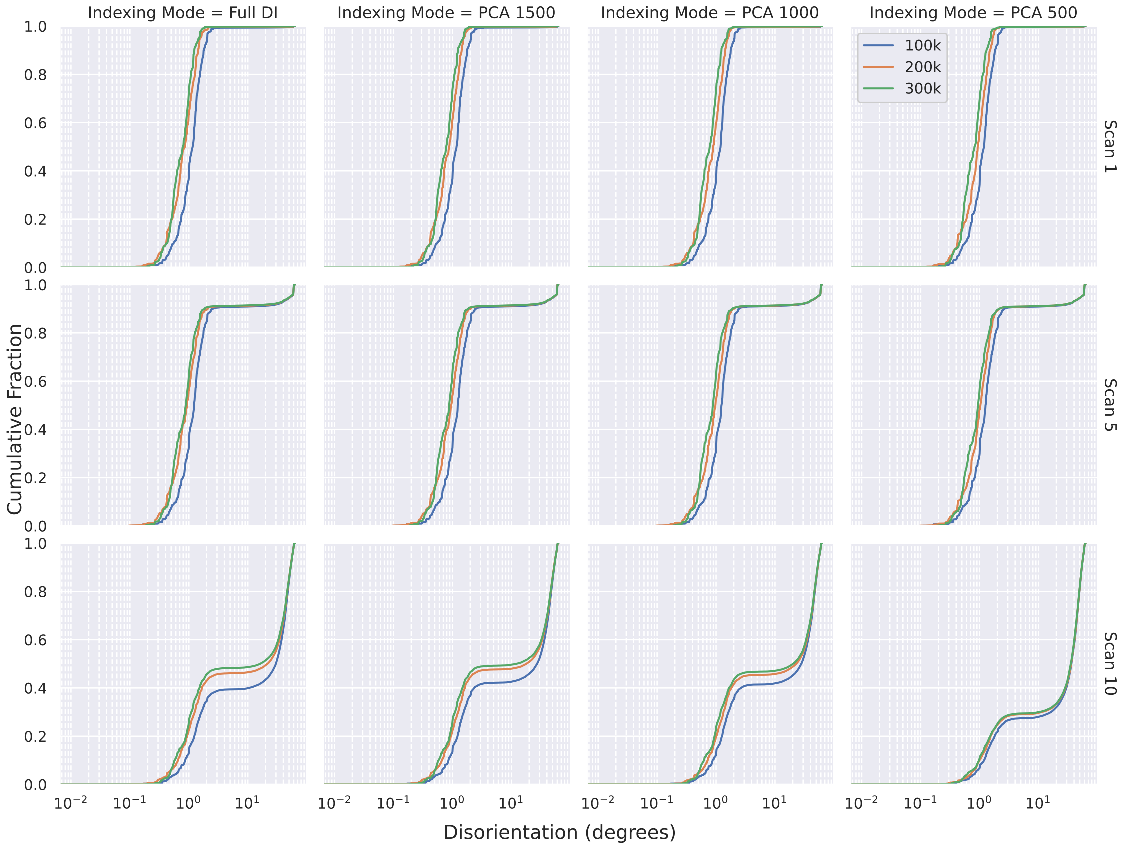

Figure 4.5: A grid of disorientation angle ECDFs across indexing mode and scan number with each plot line colored according to the size of the dictionary.

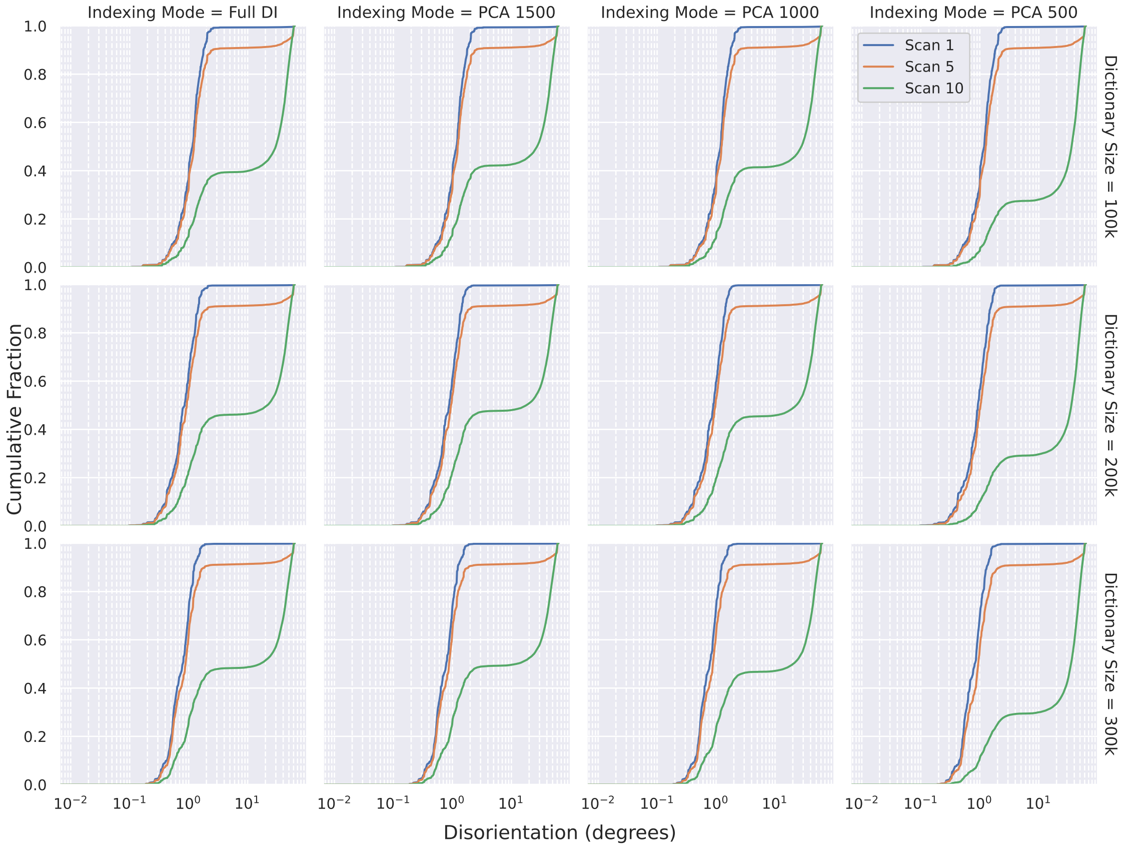

Figure 4.6: A grid of disorientation angle ECDFs across indexing mode and dictionary size with each plot line colored according to the size of the scan number.

loadings (same number of components - 2819 - as pixels), the indexing disorientation ECDF lowered back to match that of conventional dictionary indexing. This suggests that there exists an ideal number of PCA components to utilize before noise robustness lowers back to that of the conventional pattern matching. That implies the existence of a certain characteristic mode above whose frequency, additional loadings will fall below the noise level present in the patterns. By truncating the linear transformation of the data, we avoid incorporating those high frequency directions along which pattern noise could exceed the signal strength. However, this effect is only significant in the presence of very high noise levels. PCA indexing is orthogonal to other proposed methods to improve noise robustness, such as NLPAR, or the eigendecomposition of the experimental dataset itself [52, 98]. Here, neither nearby experimental patterns, nor similar patterns contribute to noise suppression for an individual experimental pattern.

Figure 4.8 shows the orientation maps for the best performing indexing mode, PCA with 1500 components, across scan number and dictionary size. The ECDFs in Figure 4.6 implied that an indexing rate of only around 90% was achieved on scan 5 at most, but it appears visually identically to scan 1 results because the discrepancies are all located at grain boundaries, where the indexed orientation flipped from one grain's orientation to that of its neighboring grain.

To better understand how PCA with 1500 components is outperforming Full DI, we map where the disorientation with the reference indexing exceeded 3° for both methods. The maps in Figure 4.9 show that there is no clear single grain which contained all of the patterns corrected by PCA 1500 as compared to Full DI. The disparities are sparsely distributed both at grain boundaries and within grain interiors. Further, Table 4.4 shows the confusion matrices for this same threshold, revealing that the discrepancy widens as the dictionary size diminishes.

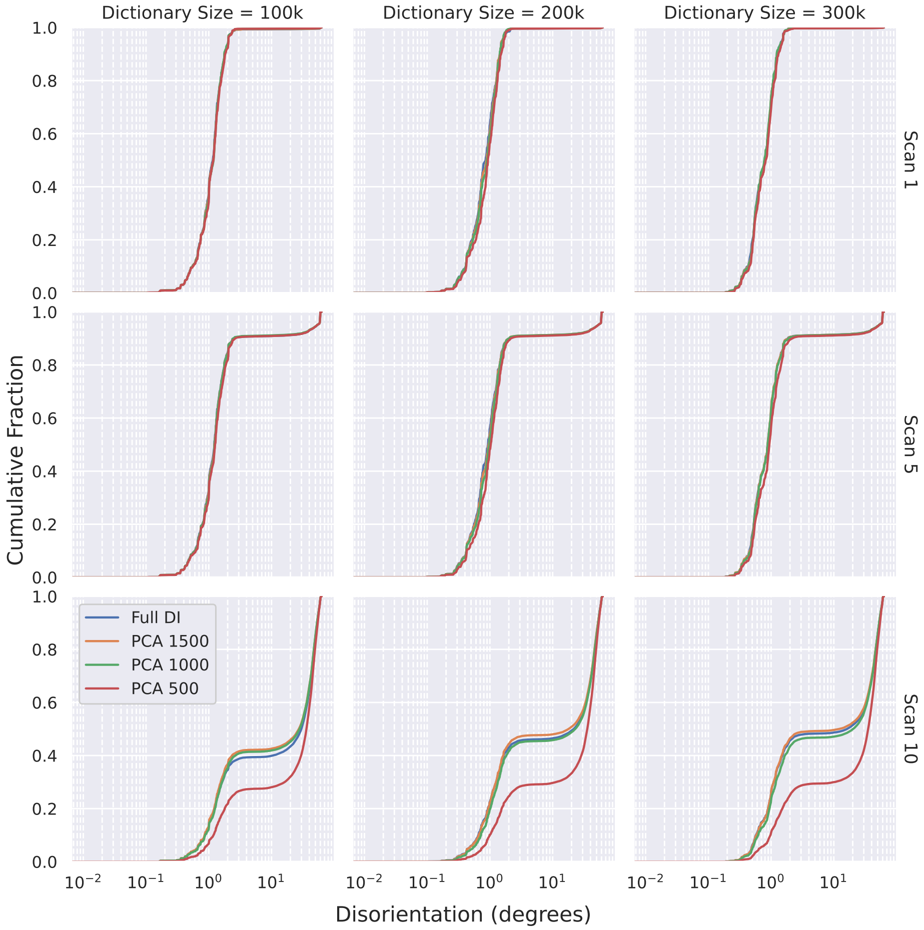

Figure 4.7: A grid of disorientation angle ECDFs across dictionary size and scan number with each plot line colored according to the indexing mode.

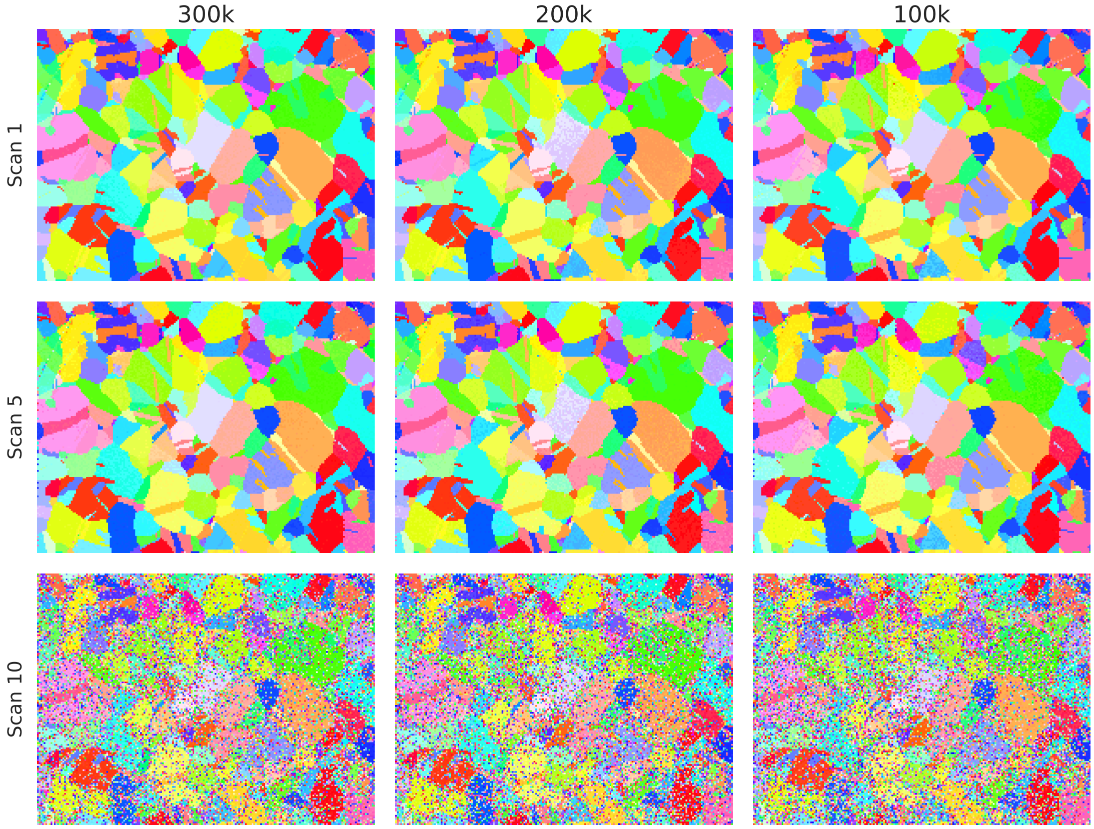

Figure 4.8: A grid of inverse pole figure (IPF) Z-axis orientation maps across scans and dictionary sizes at a fixed precision (FP32) and indexing method (1500 PCA components)

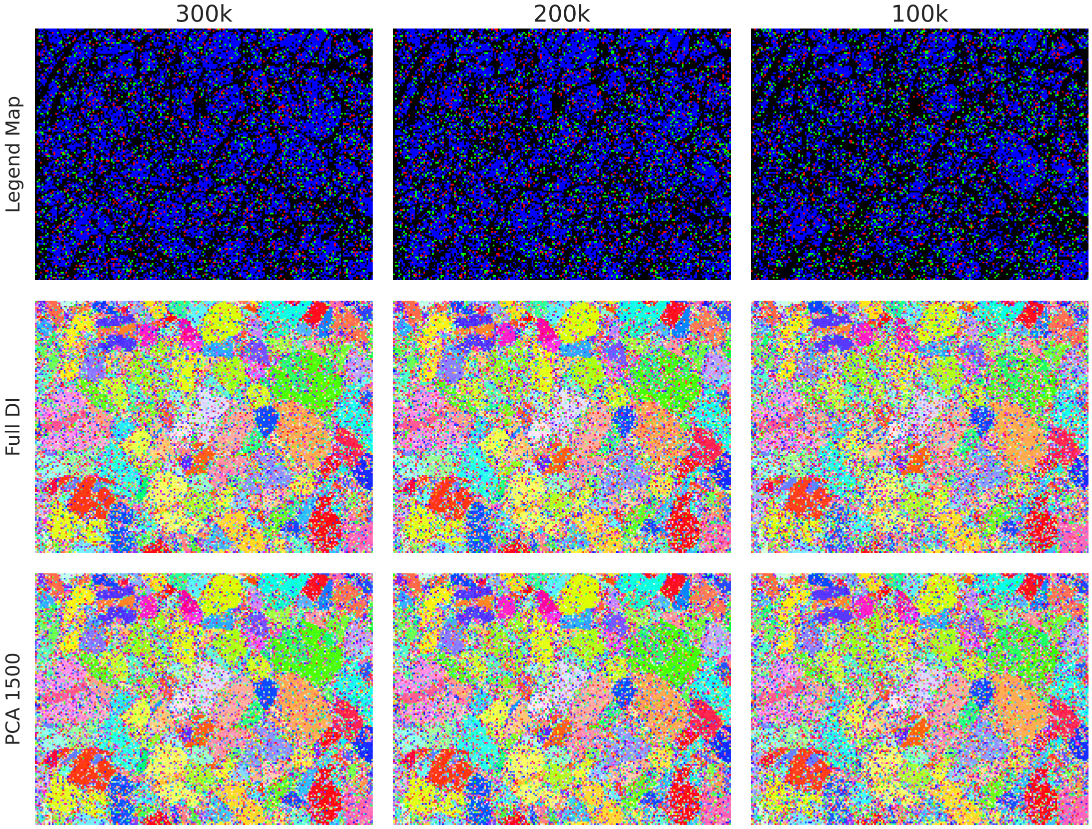

Figure 4.9: Comparison on Scan 10 between PCA with 1500 component and Full DI IPF Z-axis color maps. Top row legend: both methods exceed 3° disorientation to reference (black pixels), neither (blue) PCA 1500 only (red) Full DI only (green).

FZ Samples	**100k**		**200k**		**300k**	
	PCA 1500 Correct					
Full DI Correct	True	False	True	False	True	False
True	10368	1185	12297	1315	12923	1340
False	2020	16227	1768	14420	1636	13901

Table 4.4: Confusion matrices for each dictionary size of the 1500 component PCA and Full resolution DI across dictionary size.

4.5 Conclusion

Employing PCA and dynamic quantization, we have demonstrated a novel modification to the conventional dictionary approach to EBSD orientation indexing. This innovation allows for the processing of many hundreds of patterns per second on consumer grade CPUs, widening access to noise robust indexing. On the GPU, we achieved comparable indexing speeds to those reported for the Hough transform approach. Further, we observed an unexpected marginal increase in the indexing accuracy as compared to conventional dictionary indexing due to the suppression of higher frequency components during matching. These results come with EBSDTorch, a new Python library for EBSD pattern analysis, that inherits PyTorch's portability and dynamic quantization routines.

4.6 Data Availability

The aforementioned test dataset, acquired by Ånes et. al, is available at the following URL: https://doi.org/10.5281/zenodo.7498632. EBSDTorch code is hosted on Github at the following URL: https://github.com/ZacharyVarley/ebsdtorch.

CHAPTER V

Multimodal SEM Image Registration Tools

In the evolving field of materials characterization, the SEM has emerged as an indispensable tool for revealing the intricate details of material structure across multiple scales. Not only can a modern SEM acquire high-resolution micrographs, but it can generate diverse data through various imaging modalities such as electron backscatter diffraction, ion-induced secondary electron, forward scattered electron, and energy dispersive X-ray spectroscopy. However, the wealth of information generated presents a significant challenge: how can one accurately align and integrate these diverse data sources to extract meaningful insights about material properties and behavior. Multimodal image registration in the SEM occupies a unique niche in the larger field of image alignment problems. In contrast with medical imaging, SEM modalities typically do not exhibit large non-linear distortions; often the sample is relatively planar and a linear planar homography can describe much of the coordinate transformations to register multimodal pairs. Further, similar to satellite imaging registration problems local texture and gradient information are often correlated between modalities. In this context, this chapter briefly introduces two novel tools designed to advance multimodal SEM image registration.

5.1 Difference of Local Shannon Entropy Key Points

5.1.1 Rotation and Scaling Invariance

The preliminary results for DoLE key point matching for serial sectioning show promise for general application to both serial sectioning and data collation due to their invariance. For an example serial section from each of the 3D serial sectioning datasets, the EBSD (grayscale of IPF coloring used for key points) maps were warped onto either their corresponding BSE micrograph for MIDAS and NIST, or ISE micrograph in the case of IN100. To show the benefit of the DoLE key point approach, the EBSD maps were rotated every 1° through 359° about the center of the maps. The resultant rotation invariance is demonstrated in Figures 5.1, where the number of discovered key point inliers (out of 5000) are plotted against the rotation angle. In the disk local entropy implementation, the MIDAS image pair led to approximately 250 inliers followed by 175 for NIST, and 125 for IN100. In the box local entropy implementation, the local entropy kernel is no longer radially symmetric and dramatic so-called ”grid effect” for the three roughly rectangular sample sections is observed wherein every 90° the number of inliers spikes. If the NIST EBSD map is rotated by 45°, 135°, 225°, 315° and the box filter implementation of DoLE is used, fewer than 25 inliers are discovered.

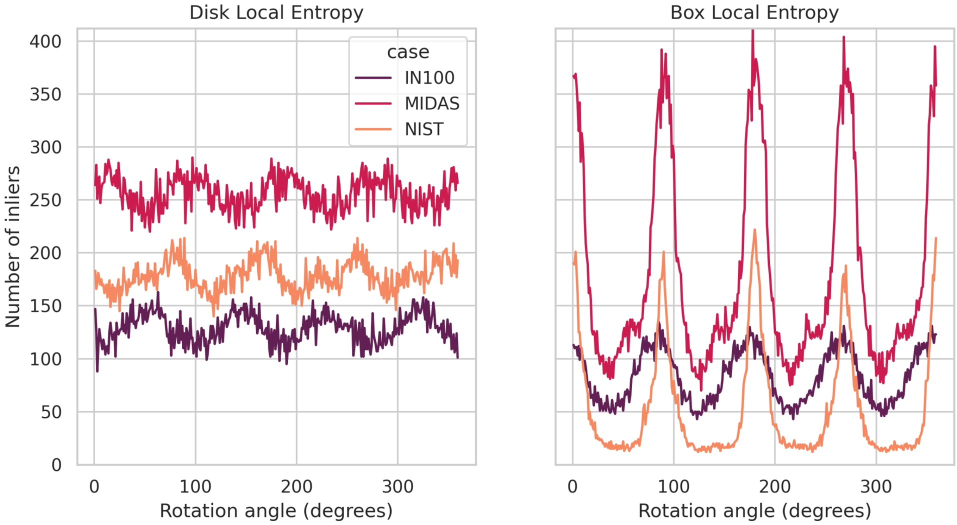

Figure 5.1: Inliers vs. centered Rotation Angle for IN100 / MIDAS / NIST for (left) disk local entropy DoLE and (right) box local entropy DoLE. The y-axis ticks and grid lines are shared.

In Figure 5.2, the EBSD maps of the same example pairs were rescaled and the number of inliers out of 5000 for each pair are plotted against rescaling factor. If the rescaling factor exceeded 1.0, the BSE/ISE modality was instead rescaled by the inverse of the factor so that the EBSD modality did not exceed the image canvas. Interestingly, the scale invariance between the box and disk DoLE implementations are comparable. The peak number of inliers for the IN100 pair is between 0.8 and 0.85 rescaling because the EBSD map had less padding outside of the sample region. For MIDAS and NIST, the true reduction factor when mapping the EBSD to BSE/ISE is closer to 0.95. Further, the scale invariance suggests invariance to axial stretching which is needed in the case of SEM systems without automatic tilt correction.

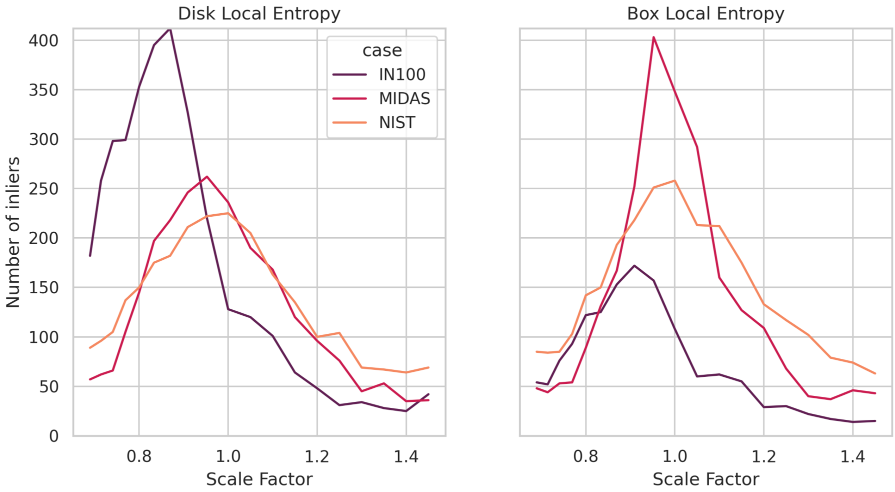

Figure 5.2: Inliers vs. centered Rescaling Factor for IN100 / MIDAS / NIST for (left) disk local entropy DoLE and (right) box local entropy DoLE. The y-axis ticks and grid lines are shared.

5.1.2 Registration Results

With a larger number of key points, the estimated homography between the EBSD maps and the BSE/ISE micrographs becomes more accurate. Figure 5.3 shows the distribution of inlier matches out of 18,000 for each of the three serial sectioning example pairs. For IN100 and MIDAS the inliers are generally uniformly distributed within the sample, but for NIST, the as-built sample microstructure texture over-emphasizes the curved exterior borders to the top and left of the section. The homography estimation can suffer if the key point inliers are selectively found within one region, and further improvements to DoLE, perhaps non-maximal suppression of optima in the scale space, are warranted.

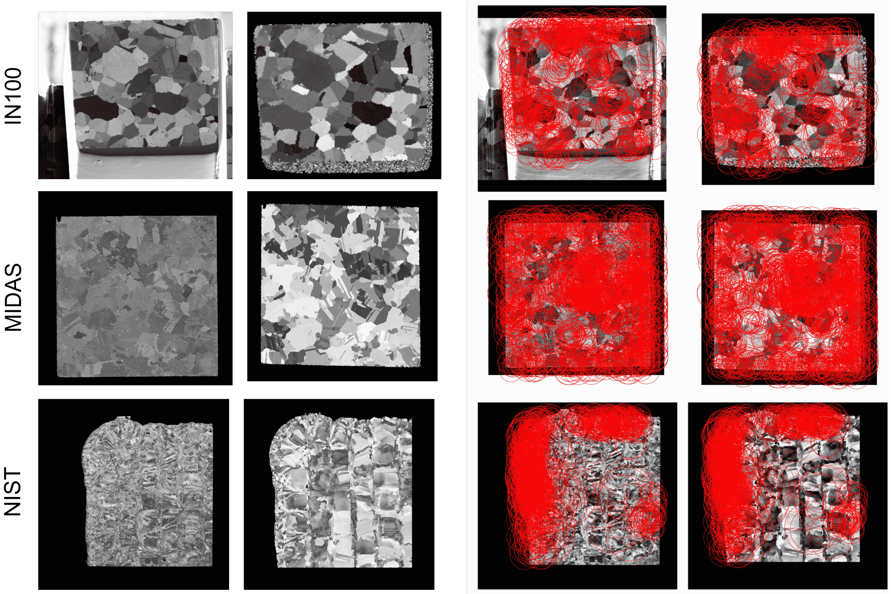

Figure 5.3: (left) serial sectioning example pairs and (right) inlier matched image patches

Following the coarse DoLE key point registration with MIND loss optimization [100], registration results are further improved, as shown in Figure 5.4. The IN100 EBSD map is correctly stretched by several pixels towards the bottom of the section, and the MIDAS EBSD is correctly shifted by marginal but important amounts. MIND loss optimization improved the binned mutual information, which has been used a standard multimodal registration tool in materials engineering for many years.

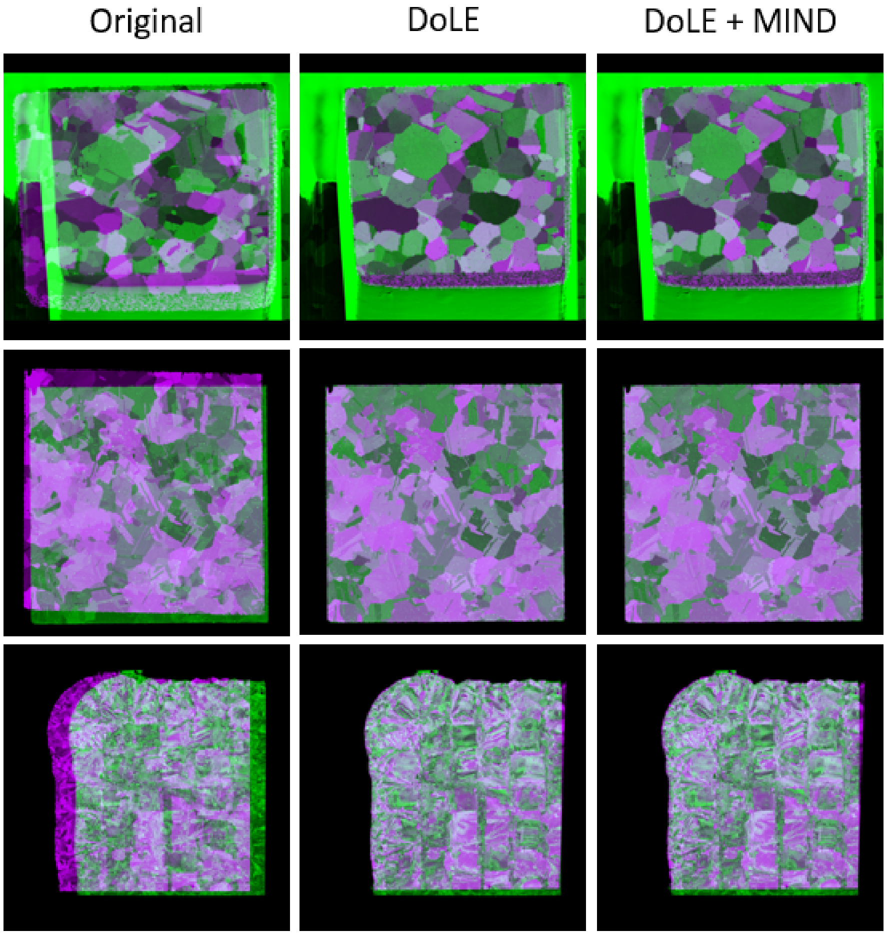

Figure 5.4: Green and magenta color channels show the ISE/BSE and EBSD data registration results. Top to bottom rows show IN100 / MIDAS / NIST results respectively. Left to right, the initial unregistered images, DoLE key point registered, and DoLE + MIND refinement registration results are displayed.

Under closer inspection, the NIST pair in Figure 5.5 indicates that MIND improved the alignment, as the red arrows denote erroneous double boundaries which are corrected following MIND loss optimization.

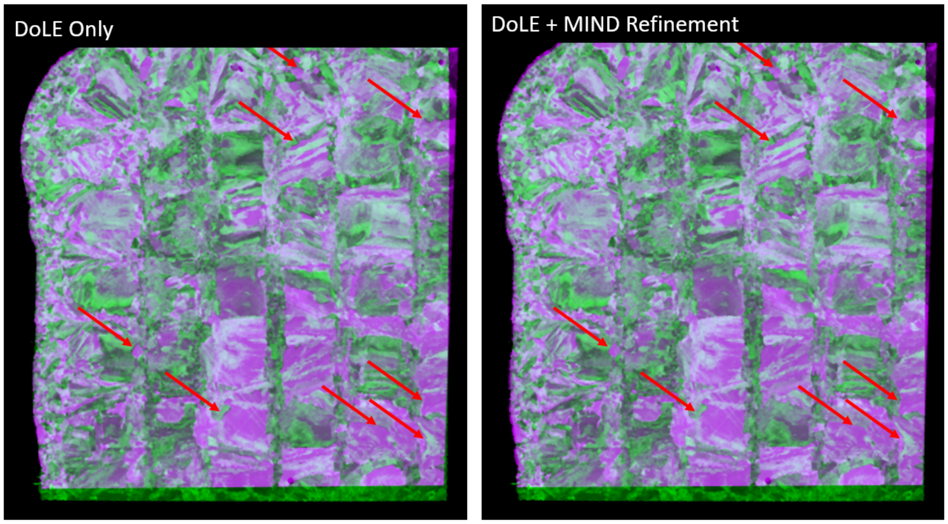

Figure 5.5: DoLE key point registered, and DoLE + MIND refinement registration results on the NIST as-built sample alone. Red arrows highlight refinement corrections.

5.2 Edgeworth Cross Mutual Information Function

Since its adoption in 1995, mutual information has been well established as a tool for the registration of multimodal image data [101]. One decade later, August and Kanade proposed using fast Fourier transforms to densely compute mutual information over discrete image shifts based on binning masks [102]. Recently, this dense computation, called the Cross Mutual Information Function (CMIF), via FFTs has been re-imagined in combination with K-means clustering of intensities in lieu of pixel intensity binning [103]. All of these methods employ a discretization of the pixel intensities to compute Shannon entropies of the marginal and joint pixel distributions of a given discrete image shift. Here we propose to marry the CMIR framework with well known asymptotic statistics based estimates of Shannon entropy in terms of Edgeworth series in order to improve accuracy and speed [104–107].

Our first contribution is to show that the previously derived 4th order Edgeworth series estimate of mutual information is correct (we suspect an error in the original implementation as suggested by the original author), which had been in question due to nonphysical highly negative mutual information values in registration tests. Our second contribution is to revisit these previous derivations of Edgeworth expansions for mutual information and remove the constraint that each marginal probability distribution possess unit standard deviation. It is not immediately apparent whether a "true" asymptotic estimate of the entropy of a pixel intensity distribution should be based on deviation from its standardized form. We compare Edgeworth cross mutual information functions for examples with and without standardization for both 3rd and 4th order expansions.

5.2.1 Theory

Shannon Entropy

For univariate and bivariate distributions with probability distribution functions (pdf's) $p(x)$ and $p(x, y)$, Shannon entropy is given by:

$$H(x) = -\int_{\mathbb{R}} p(x) \ln p(x) dx \tag{5.1}$$

$$H(x, y) = -\int_{\mathbb{R}^2} p(x, y) \ln p(x, y) dx dy \tag{5.2}$$

where where $\int_{\mathbb{R}} p(x)dx = 1$ and $\int_{\mathbb{R}^2} p(x, y)p(x, y)dxdy = 1$. In both cases, the entropy is calculated by taking the expected value of the negative logarithm of the pdf over the entire domain.

Binned Mutual Information

The mutual information of a bivariate distribution measures the amount of shared information between its two constituent variables and is given by:

$$I(x, y) = H(x) + H(y) - H(x, y) \tag{5.3}$$

Intuitively, those bivariate distributions which are highly uniform and entropic along their marginal distributions, but have relatively compact, impulse-like, low entropy joint distributions will possess high mutual information. Mutual information may be normalized in a variety of ways [108], although for the scope of this publication, mutual information values are arbitrarily left un-normalized for simplicity. These integrals in equations (5.1) and (5.2) can be evaluated numerically by discretization of their respective domains, yielding:

$$H(x) = -\sum_{i=1}^{N} p(x_i) \ln p(x_i) \tag{5.4}$$

$$H(x,y) = -\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} p(x_i, y_j)\ln p(x_i, y_{2j}) \tag{5.5}$$

where N is the number of bins for the univariate distribution, and N_1 and N_2 are the number of bins for each variable in the bivariate distribution. The mutual information can then be computed using the discretized entropies with equation 5.3.

Edgeworth Series Mutual Information

Following an adaptation of the cited derivation (see appendix of [106] in French), the 3rd order Edgeworth expansion estimate of joint entropy is given by:

$$\begin{aligned} \mathrm{H}_{3\mathrm{E}}(\boldsymbol{x},\boldsymbol{y}) &= 1 + \log 2\pi + \frac{\log\left(1-\kappa_{1,1}^2\right)}{2} \\ &- \frac{1}{12(1-\kappa_{1,1}^2)^3}\Bigg(\kappa_{30}^2 + 3\kappa_{21}^2 + 3\kappa_{12}^2 + \kappa_{03}^2 \\ &\qquad - 2\kappa_{11}^3\left(\kappa_{03}\kappa_{30} + 3\kappa_{12}\kappa_{21}\right) \\ &\qquad + 6\kappa_{11}^2\left(\kappa_{12}\kappa_{30} + \kappa_{21}^2 + \kappa_{03}\kappa_{21} + \kappa_{12}^2\right) \\ &\qquad - 6\kappa_{11}\left(\kappa_{21}\kappa_{30} + 2\kappa_{02}\kappa_{12}\kappa_{21} + \kappa_{03}\kappa_{12}\right)\Bigg) \end{aligned} \tag{5.6}$$

where κ_{ij} is the ith and jth indexed cumulant according to variables x and y. As expected, the estimate is invariant under the exchanging of x and y. If the distributions are not necessarily centered and standardized, κ_{ij} can be calculated from the so-called raw bivariate moments $\mu_{i,j}$ according to the following well known equations [109]:

$$\begin{aligned}
\kappa_{1,0} &= \mu'_{1,0} \\
\kappa_{2,0} &= \mu'_{2,0} - \mu'^{2}_{1,0} \\
\kappa_{1,1} &= \mu'_{1,1} - \mu'_{1,0}\mu'_{0,1} \\
\kappa_{3,0} &= \mu'_{3,0} - 3\mu'_{2,0}\mu'_{1,0} + 2\mu'^{3}_{1,0} \\
\kappa_{2,1} &= \mu'_{2,1} - \mu'_{2,0}\mu'_{0,1} - 2\mu'_{1,1}\mu'_{1,0} + 2\mu'^{2}_{1,0}\mu'_{0,1} \\
\kappa_{4,0} &= \mu'_{4,0} - 3\mu'^{2}_{2,0} - 4\mu'_{3,0}\mu'_{1,0} + 12\mu'_{2,0}\mu'^{2}_{1,0} - 6\mu'^{4}_{1,0} \\
\kappa_{3,1} &= \mu'_{3,1} - \mu'_{3,0}\mu'_{0,1} - 3\mu'_{2,1}\mu'_{1,0} - 3\mu'_{2,0}\mu'_{1,1} + 6\mu'_{2,0}\mu'_{1,0}\mu'_{0,1} \\
&\quad + 6\mu'_{1,1}\mu'^{2}_{1,0} - 6\mu'^{3}_{1,0}\mu'_{0,1} \\
\kappa_{2,2} &= \mu'_{2,2} - 2\mu'_{2,1}\mu'_{0,1} - \mu'_{2,0}\mu'_{0,2} - 2\mu'_{1,2}\mu'_{1,0} - 2\mu'^{2}_{1,1} + 8\mu'_{1,1}\mu'_{1,0}\mu'_{0,1} \\
&\quad - 6\mu'^{2}_{1,0}\mu'^{2}_{0,1} + 2\mu'^{2}_{1,0}\mu'_{0,2}
\end{aligned} \tag{5.7}$$

Where $\kappa_{0,2}$ can be found by swapping the indices in the definition of $\kappa_{2,0}$. The raw moments are given by:

$$\mu'_{i,j} = \mathbf{E}[x^i y^j] \tag{5.8}$$

$$
\begin{aligned}
&\mathrm{H}_{4\mathrm{E}}(\boldsymbol{x},\boldsymbol{y}) = \mathrm{H}_{3\mathrm{E}}(\boldsymbol{x},\boldsymbol{y}) - \\
&\frac{1}{48\left(1-\rho^2\right)^4}\Big(k_{4,0}^2 + 4k_{3,1}^2 + 6k_{2,2}^2 + 4k_{1,3}^2 + k_{0,4}^2 \\
&\qquad - 8\rho\left(k_{4,0}k_{3,1} + 3k_{3,1}k_{2,2} + 3k_{2,2}k_{1,3} + k_{1,3}k_{0,4}\right) \\
&\qquad + 12\rho^2\left(k_{3,1}^2 + 2k_{2,2}^2 + k_{1,3}^2 + k_{4,0}k_{2,2} + 2k_{3,1}k_{1,3} + k_{2,2}k_{0,4}\right) \\
&\qquad - 8\rho^3\left(k_{4,0}k_{1,3} + 3k_{3,1}k_{2,2} + k_{3,1}k_{0,4} + 3k_{2,2}k_{1,3}\right) + 2\rho^4\left(3k_{2,2}^2 + k_{4,0}k_{0,4} + 4k_{3,1}k_{1,3}\right)\Big) \\
&-\frac{10}{144\left(1-\rho^2\right)^6}\Big(k_{3,0}^4 + 6k_{3,0}^2k_{2,1}^2 + 15k_{3,0}^2k_{1,2}^2 + 20k_{3,0}^2k_{0,3}^2 + 15k_{2,1}^2k_{0,3}^2 + 6k_{1,2}^2k_{0,3}^2 + k_{0,3}^4 \\
&\qquad - 12\rho\left(k_{3,0}^3k_{2,1} + 10k_{3,0}^2k_{1,2}k_{0,3} + 5k_{3,0}^2k_{2,1}k_{1,2} + 10k_{3,0}k_{2,1}k_{0,3}^2 + 5k_{2,1}k_{1,2}k_{0,3}^2 + k_{1,2}k_{0,3}^3\right) \\
&\qquad + 30\rho^2\big(k_{3,0}^2k_{2,1}^2 + 4k_{3,0}^2k_{1,2}^2 + 6k_{3,0}^2k_{0,3}^2 + 4k_{2,1}^2k_{0,3}^2 + k_{1,2}^2k_{0,3}^2 + k_{3,0}^3 \\
&\qquad\qquad k_{1,2} + 4k_{3,0}^2k_{2,1}k_{0,3} + 6k_{3,0}k_{2,1}k_{1,2}k_{0,3} + 4k_{3,0}k_{1,2}k_{0,3}^2 + k_{2,1}k_{0,3}^3\big) \\
&\qquad - 40\rho^3\big(k_{3,0}^3k_{0,3} + 9k_{3,0}^2k_{1,2}k_{0,3} + 3k_{3,0}^2k_{2,1}k_{1,2} + 3k_{3,0}k_{2,1}^2k_{0,3} \\
&\qquad\qquad + 9k_{3,0}k_{2,1}k_{0,3}^2k_{3,0}k_{1,2}^2k_{0,3} + 3k_{2,1}k_{1,2}k_{0,3}^2 + k_{3,0}k_{0,3}^3\big) \\
&\qquad + 30\rho^4\left(3k_{3,0}^2k_{1,2}^2 + 6k_{3,0}^2k_{0,3}^2 + 3k_{2,1}^2k_{0,3}^2 + 5k_{3,0}^2k_{2,1}k_{0,3} + 10k_{3,0}k_{2,1}k_{1,2}k_{0,3} + 5k_{3,0}k_{1,2}k_{0,3}^2\right) \\
&\qquad - 12\rho^5\left(5k_{3,0}k_{2,1}^2k_{0,3} + 11k_{3,0}k_{2,1}k_{0,3}^2 + 11k_{3,0}^2k_{1,2}k_{0,3} + 5k_{3,0}k_{1,2}^2k_{0,3}\right) \\
&\qquad + 2\rho^6\left(11k_{3,0}^2k_{0,3}^2 + 21k_{3,0}k_{2,1}k_{1,2}k_{0,3}\right)\Big)
\end{aligned}
\tag{5.9}
$$

where the mean and variance of each variable are zero and unity. Equations 5.6 and 5.9 provide estimates of the joint distribution entropy, while the marginal distribution is estimated as to 4th order as:

$$
\mathrm{H(x)}_{4\mathrm{E}}(\boldsymbol{x}) = 0.5\log(2\pi e) - \frac{\kappa_{30}^2}{12} - \frac{\kappa_{40}^2}{48} - \frac{5\kappa_{30}^4}{72} \tag{5.10}
$$

and to 3rd order as:

$$
\mathrm{H(x)}_{3\mathrm{E}}(\boldsymbol{x}) = 0.5\log(2\pi e) - \frac{\kappa_{30}^2}{12} \tag{5.11}
$$

If we relax the constraint that each marginal distribution have variance of one, we find the following form of the 3rd order approximation:

$$
\begin{aligned}
\mathrm{H}_{3\mathrm{E}}(\boldsymbol{x},\boldsymbol{y}) &= 1+\log(2\pi)+0.5\log(|\Sigma|)\\
&-\frac{1}{12|\Sigma|^3}\Bigg(\kappa_{02}^3\kappa_{30}^2+3\kappa_{02}^2\kappa_{20}\kappa_{21}^2+3\kappa_{02}\kappa_{12}^2\kappa_{20}^2+\kappa_{03}^2\kappa_{20}^3\\
&\qquad-2\kappa_{11}^3\left(\kappa_{03}\kappa_{30}+3\kappa_{12}\kappa_{21}\right)\\
&\qquad+6\kappa_{11}^2\left(\kappa_{02}\kappa_{12}\kappa_{30}+\kappa_{02}\kappa_{21}^2+\kappa_{03}\kappa_{20}\kappa_{21}+\kappa_{12}^2\kappa_{20}\right)\\
&\qquad-6\kappa_{11}\left(\kappa_{02}^2\kappa_{21}\kappa_{30}+2\kappa_{02}\kappa_{12}\kappa_{20}\kappa_{21}+\kappa_{03}\kappa_{12}\kappa_{20}^2\right)\Bigg)
\end{aligned}
\tag{5.12}
$$

and for the 4th order approximation:

$$
\begin{aligned}
&\mathrm{H}_{4\mathrm{E}}(\boldsymbol{x},\boldsymbol{y}) = 1 + \log(2\pi) + 0.5\log(|\Sigma|) \\
&- \frac{1}{12|\Sigma|^3}\Bigg(\kappa_{02}^3\kappa_{30}^2 + 3\kappa_{02}^2\kappa_{20}\kappa_{21}^2 + 3\kappa_{02}\kappa_{12}^2\kappa_{20}^2 + \kappa_{03}^2\kappa_{20}^3 \\
&\qquad - 2\kappa_{11}^3\left(\kappa_{03}\kappa_{30} + 3\kappa_{12}\kappa_{21}\right) \\
&\qquad + 6\kappa_{11}^2\left(\kappa_{02}\kappa_{12}\kappa_{30} + \kappa_{02}\kappa_{21}^2 + \kappa_{03}\kappa_{20}\kappa_{21} + \kappa_{12}^2\kappa_{20}\right) \\
&\qquad - 6\kappa_{11}\left(\kappa_{02}^2\kappa_{21}\kappa_{30} + 2\kappa_{02}\kappa_{12}\kappa_{20}\kappa_{21} + \kappa_{03}\kappa_{12}\kappa_{20}^2\right)\Bigg) \\
&- \frac{1}{48|\Sigma|^4}\Bigg(\kappa_{02}^4\kappa_{40}^2 + 4\kappa_{02}^3\kappa_{20}\kappa_{31}^2 + 6\kappa_{02}^2\kappa_{20}^2\kappa_{22}^2 + 4\kappa_{02}\kappa_{13}^2\kappa_{20}^3 + \kappa_{04}^2\kappa_{20}^4 \\
&\qquad + 2\kappa_{11}^4\left(\kappa_{04}\kappa_{40} + 4\kappa_{13}\kappa_{31} + 3\kappa_{22}^2\right) \\
&\qquad - 8\kappa_{11}^3\left(\kappa_{02}\kappa_{13}\kappa_{40} + 3\kappa_{02}\kappa_{22}\kappa_{31} + \kappa_{04}\kappa_{20}\kappa_{31} + 3\kappa_{13}\kappa_{20}\kappa_{22}\right) \\
&\qquad + 12\kappa_{11}^2\left(\kappa_{02}^2\kappa_{22}\kappa_{40} + \kappa_{02}^2\kappa_{31}^2 + 2\kappa_{02}\kappa_{13}\kappa_{20}\kappa_{31} + 2\kappa_{02}\kappa_{20}\kappa_{22}^2 + \kappa_{04}\kappa_{20}^2\kappa_{22} + \kappa_{13}^2\kappa_{20}^2\right) \\
&\qquad - 8\kappa_{11}\left(\kappa_{02}^3\kappa_{31}\kappa_{40} + 3\kappa_{02}^2\kappa_{20}\kappa_{22}\kappa_{31} + 3\kappa_{02}\kappa_{13}\kappa_{20}^2\kappa_{22} + \kappa_{04}\kappa_{13}\kappa_{20}^3\right)\Bigg) \\
&- \frac{10}{144|\Sigma|^6}\Bigg(\kappa_{02}^6\kappa_{30}^4 + 6\kappa_{02}^5\kappa_{20}\kappa_{21}^2\kappa_{30}^2 + 15\kappa_{02}^4\kappa_{12}^2\kappa_{20}^2\kappa_{30}^2 + 20\kappa_{02}^3\kappa_{03}^2\kappa_{20}^3\kappa_{30}^2 \\
&\qquad + 15\kappa_{02}^2\kappa_{03}^2\kappa_{20}^4\kappa_{21}^2 + 6\kappa_{02}\kappa_{03}^2\kappa_{12}^2\kappa_{20}^5 + \kappa_{03}^4\kappa_{20}^6 \\
&\qquad + 2\kappa_{11}^6\kappa_{03}\kappa_{30}\left(11\kappa_{03}\kappa_{30} + 21\kappa_{12}\kappa_{21}\right) \\
&\qquad - 12\kappa_{11}^5\kappa_{03}\kappa_{30}\left(11\kappa_{02}\kappa_{12}\kappa_{30} + 5\kappa_{02}\kappa_{21}^2 + 11\kappa_{03}\kappa_{20}\kappa_{21} + 5\kappa_{12}^2\kappa_{20}\right) \\
&\qquad + 30\kappa_{11}^4\big(5\kappa_{02}^2\kappa_{03}\kappa_{21}\kappa_{30}^2 + 3\kappa_{02}^2\kappa_{12}^2\kappa_{30}^2 + 6\kappa_{02}\kappa_{03}^2\kappa_{20}\kappa_{30}^2 \\
&\qquad\qquad + 10\kappa_{02}\kappa_{03}\kappa_{12}\kappa_{20}\kappa_{21}\kappa_{30} + 5\kappa_{03}^2\kappa_{12}\kappa_{20}^2\kappa_{30} + 3\kappa_{03}^2\kappa_{20}^2\kappa_{21}^2\big) \\
&\qquad - 40\kappa_{11}^3\big(\kappa_{02}^3\kappa_{03}\kappa_{30}^3 + 3\kappa_{02}^3\kappa_{12}\kappa_{21}\kappa_{30}^2 + 9\kappa_{02}^2\kappa_{03}\kappa_{12}\kappa_{20}\kappa_{30}^2 + 3\kappa_{02}^2\kappa_{03}\kappa_{20}\kappa_{21}^2\kappa_{30} \\
&\qquad\qquad + 9\kappa_{02}\kappa_{03}^2\kappa_{20}^2\kappa_{21}\kappa_{30} + 3\kappa_{02}\kappa_{03}\kappa_{12}^2\kappa_{20}^2\kappa_{30} + \kappa_{03}^3\kappa_{20}^3\kappa_{30} + 3\kappa_{03}^2\kappa_{12}\kappa_{20}^3\kappa_{21}\big) \\
&\qquad + 30\kappa_{11}^2\big(\kappa_{02}^4\kappa_{12}\kappa_{30}^3 + \kappa_{02}^4\kappa_{21}^2\kappa_{30}^2 + 4\kappa_{02}^3\kappa_{03}\kappa_{20}\kappa_{21}\kappa_{30}^2 + 4\kappa_{02}^3\kappa_{12}^2\kappa_{20}\kappa_{30}^2 \\
&\qquad\qquad + 6\kappa_{02}^2\kappa_{03}^2\kappa_{20}^2\kappa_{30}^2 + 6\kappa_{02}^2\kappa_{03}\kappa_{12}\kappa_{20}^2\kappa_{21}\kappa_{30} + 4\kappa_{02}\kappa_{03}^2\kappa_{12}\kappa_{20}^3\kappa_{30} \\
&\qquad\qquad + 4\kappa_{02}\kappa_{03}^2\kappa_{20}^3\kappa_{21}^2 + \kappa_{03}^3\kappa_{20}^4\kappa_{21} + \kappa_{03}^2\kappa_{12}^2\kappa_{20}^4\big) \\
&\qquad - 12\kappa_{11}\big(\kappa_{02}^5\kappa_{21}\kappa_{30}^3 + 5\kappa_{02}^4\kappa_{12}\kappa_{20}\kappa_{21}\kappa_{30}^2 + 10\kappa_{02}^3\kappa_{03}\kappa_{12}\kappa_{20}^2\kappa_{30}^2 \\
&\qquad\qquad + 10\kappa_{02}^2\kappa_{03}^2\kappa_{20}^3\kappa_{21}\kappa_{30} + 5\kappa_{02}\kappa_{03}^2\kappa_{12}\kappa_{20}^4\kappa_{21} + \kappa_{03}^3\kappa_{12}\kappa_{20}^5\big)\Bigg)
\end{aligned}
\tag{5.13}
$$

Where $|\Sigma|$ is the determinant of the covariance matrix. When setting κ_{20} and κ_{02} to unity we recover the original formulation in equation 5.9, and as expected the formulation is equivalent under an exchange of x and y with the relaxed constraint.

Cross Mutual Information

The cross mutual information function is an intensity change invariant analogue to the cross correlation between two signals. It can be estimated using the fast Fourier transform (FFT). By systematically computing the bin occupancy going into equations (5.1) and (5.2) as a function of shift, the shared information content as estimated by binned mutual information may be densely computed over shifts. For Edgeworth series estimates of mutual information, the bivariate k-statistics can be densely computed over discrete pixel shifts, instead, and there are dramatically fewer raw moments in equation (5.7) used to compute the bivariate k-statistics than there are combinations of grayscale intensity bins between the two images.

5.2.2 Preliminary Results

Figure 5.6 shows the result of applying the CMIF and ECMIF functions to an example image pair from the classic IN100 dataset, with the two modalities as a grayscale rendering of the EBSD map IPF coloring, and the intensity from the ion-induced secondary electron images. The first 4 rows of results are for the standard and nonstandard formulations for 4th and 3rd order Edgeworth mutual information. This figure represents the first time that mutual information has been densely computed over discrete shifts using Edgeworth's series expansion. The latter 3 rows show the conventional binned normalized mutual information maps with 4, 16, and 32 bins. These required 16, 256, and 1024 cross correlations to compute, while the Edgeworth series approaches required 10 and 15 cross correlations for 3rd and 4th order approximations respectively. This is the time limiting step for these computations. The results generally show that the mutual information surfaces are comparable, but the binned mutual information clearly more reproducibly has the peak value in near the correct position at a variety of binning levels. Significant further investigation is warranted.

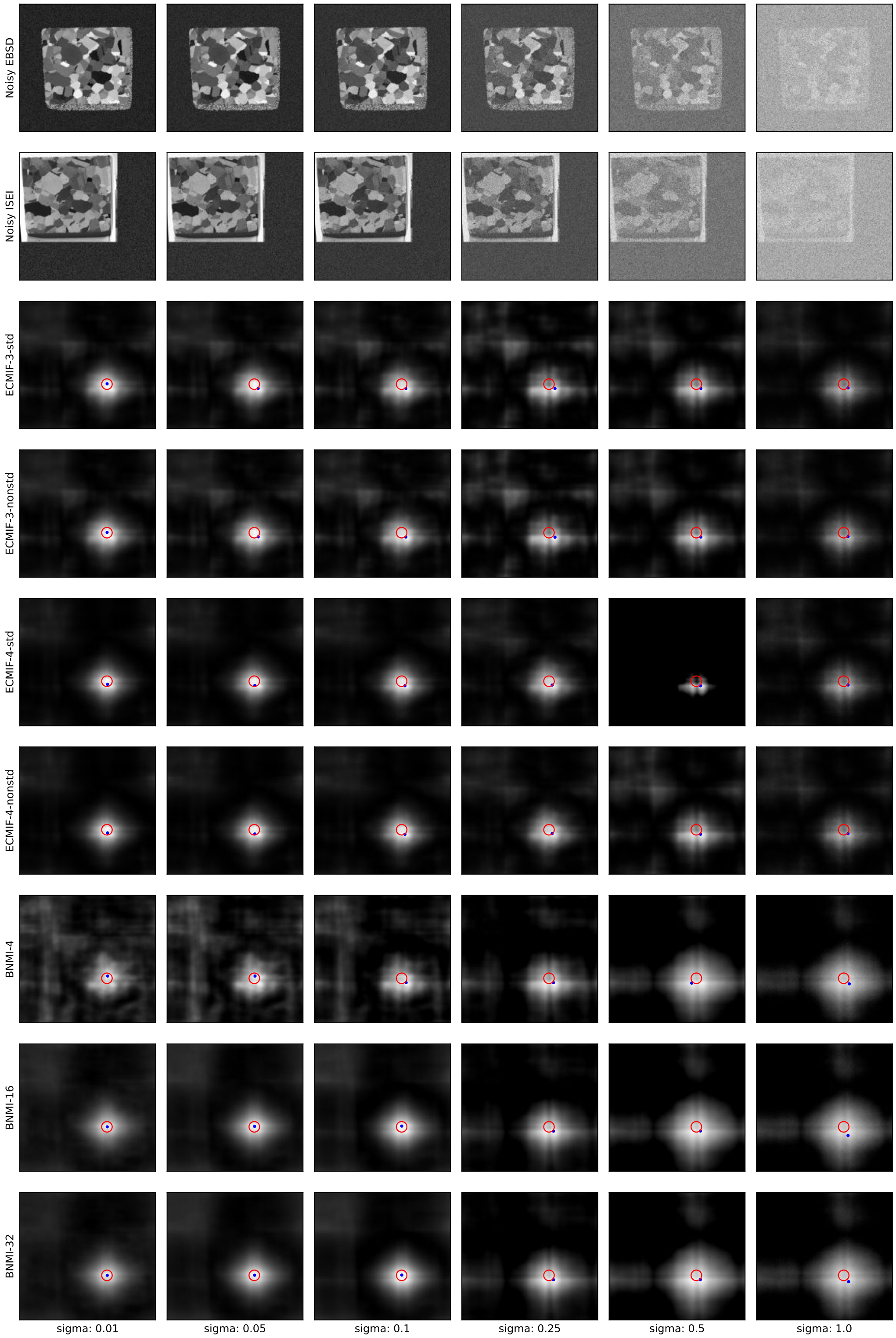

Figure 5.6: Cross mutual information surfaces indicating the estimated mutual information (normalized per image) between the example ISEI and EBSD scan with both the existing and new Edgeworth series formulation. Additive Gaussian noise has been applied with the labeled σ for each column. The ideal peak position indicating the correct offset is circled in red, and the actual locations are in blue.

CHAPTER VI

Decision Tree Reductions for Laue Classes

6.1 Introduction and Motivation

The reduction of a generic orientation from SO(3) to the symmetry reduced quotient space SO(3)/K, for a given crystallographic (proper rotational) point group (C_1, C_2, C_3, C_4, C_6, D_2, D_3, D_4, D_6, T, or O in Schoenflies notation) is a common subproblem in many texture analysis tasks. For example, when reconstructing time-sequential data volumes via High Energy X-ray Diffraction Microscopy (HEDM), the tracking of grains across time-steps, called grain matching, often requires $O(n^2)$ pairwise disorientation angle calculations where n is the number of grains; see section C. of the methods in [110]. Another such example is the computation of Grain Reference Orientation Deviation (GROD), wherein every voxel within a grain must be compared to the average orientation of that grain, via a disorientation angle computation, which can quickly grow in cost with increasing voxels per grain and grain count [111]. A third and final example is the computation of Kernel Average Misorientation (KAM), which measures local misorientations by computing the average disorientation angle between a central voxel and all other voxels in a fixed kernel size [112]. Any improvements to efficiency realized in reducing orientations to the fundamental zone provide immediate benefits to a variety of such downstream analyses involving repetitive disorientation angle calculations.

6.2 Theory and Background

The idea of a fundamental zone to represent the reduced (true) orientations of crystals was put forth by V. P. Yashnikov and H.J. Bunge [113–115]. Formally, for a given space group

K_O, let SO(K_o) denote the admissible finite subgroup of SO(3):

$$SO(K_o) = \{k_o = e; k_1; \dots k_N\} \tag{6.1}$$

For a general rotation, g, of the crystal lattice it is either closest to one of the proper rotational point group elements $\{k_o = e; k_1; \dots k_N\}$ or it is on the boundary, equidistant to two or more of these elements:

$$D(k_i) = \{g; k_i = \operatorname*{argmin}_{k_o \dots k_N} \operatorname{dist}(g, k_i)\} \tag{6.2}$$

This provides a regular partition of all of SO(3) into topologically equivalent domains called the Dirichlet-Voronoi partition dual to the given finite rotation set:

$$D(k_o), D(k_1) \dots D(k_N) \tag{6.3}$$

6.2.1 Generator Adjacency

By enumerating both the generators of a given Laue group and their southern real hemisphere $\mathbb{S}^3_-$ equivalent counterparts (negate each generator for the total double covering of SO(3)), the correct convex hull of the generators can be procured [116]. The resulting tetrahedra from the Delaunay triangulation are each defined with four generators as vertices, and their edges directly indicate which generators have touching Voronoi cells on the 3-sphere (S^3). Despite computing the convex hull in $\mathbb{R}^4$ without any constraint unto S^3, the results are identical with respect to adjacency because the angle between two points on S^3 is a monotonic function of their Euclidean distance in $\mathbb{R}^4$. Overall, this procedure is useful for understanding the adjacency relationships amongst the generators in the proper rotational point groups.

6.2.2 Infeasible Generators

Two proper crystallographic rotational symmetry groups, D6 and O, contain symmetry elements whose Voronoi cells cannot be reached via the composition of two rotations both within that group's fundamental zone. We call these operations "(composition) infeasible generators." Crystallographic misorientations, wherein there are two entities both possessing their own crystallographic symmetry, inherit these results. In the following two sections, we demonstrate the infeasibility of generators in the Laue groups D6 and O, all of which are 180° rotations about [100], [010] or [010]. For Laue group D6, we show via constrained optimization that while the maximum FZ angle is sufficient to reach the aforementioned infeasible generator's Voronoi cell under composition, restricting the rotation axes to the FZ prohibits reaching the infeasible generator (generator 4 in Table 6.1). Similarly, for Laue group O, we prove the infeasibility of the generators corresponding to 180° rotations about the [001], [100], and [010] axes (generators 2, 3 and 4 in Table 6.3). Although generators 19-24 in O are technically feasible under composition, the probability of reaching them via composing two independently and uniformly sampled rotations from the fundamental zone is vanishingly small.

Laue Group D6

The composition $q_a q_b^{-1}$ of two orientations, q_a and q_b both within the Laue D6 fundamental zone (those orientations nearest to the identity amongst D6's generators) cannot be closest to the generator associated with 180° rotation about [001]. The 12 generators for the D6 group are given below as quaternions:

Generator	w	x	y	z	Description
1	1	0	0	0	identity operator
2	$\frac{1}{2}$	0	0	$\frac{\sqrt{3}}{2}$	60 about [001]
3	$\frac{1}{2}$	0	0	$-\frac{\sqrt{3}}{2}$	120 about [001]
4*	**0**	**0**	**0**	**1**	**180 about [001]**
5	$\frac{\sqrt{3}}{2}$	0	0	$\frac{1}{2}$	240 about [001]
6	$\frac{\sqrt{3}}{2}$	0	0	$-\frac{1}{2}$	300 about [001]
7	0	1	0	0	180 about [100]
8	0	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	0	180 about $[y = \frac{x}{\sqrt{3}}]$
9	0	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	0	180 about $[y = \sqrt{3}x]$
10	0	0	1	0	180 about [010]
11	0	$-\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	0	180 about $[y = -\sqrt{3}x]$
12	0	$-\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	0	180 about $[y = \frac{-x}{\sqrt{3}}]$

Table 6.1: Generators for Laue group D6 with the zone infeasible under FZ-FZ composition bolded.

In order to prove that the generator 4 zone is infeasible, it would be convenient if the rotation angle magnitude of $q_a q_b^{-1}$ were bounded well below 180°; however, the maximum angle of the D6, $\theta_{\mathrm{D6}} = (2 \arctan \sqrt{15 - 8\sqrt{3}}) \approx 1.6378$ radians or 93.841°, is clearly sufficient under composition of two rotations to reach $180° - \theta_{\mathrm{D6}}$, the minimum possible angular distance to the generator 4 zone. This fact necessitates for the sake of a proof, further analysis beyond angle magnitude arguments. To proceed, we constrain two quaternions, q_a and q_b, to the D6 FZ and work to minimize the angle between $q_a q_b^{-1}$ and the generator 4 zone. There are 14 boundary planes flanking the D6 RFZ. The boundaries are listed in the following table:

Generators	RFZ Inequalities (from $\pm q$)	
5	$w(2-\sqrt{3}) - z \geq 0$	
6	$w(2-\sqrt{3}) + z \geq 0$	
7	$w - x \geq 0$	$w + x \geq 0$
8	$2w - \sqrt{3}x - y \geq 0$	$2w + \sqrt{3}x + y \geq 0$
9	$2w - x - \sqrt{3}y \geq 0$	$2w + x + \sqrt{3}y \geq 0$
10	$w - y \geq 0$	$w + y \geq 0$
11	$2w + x - \sqrt{3}y \geq 0$	$2w - x + \sqrt{3}y \geq 0$
12	$2w + \sqrt{3}x - y \geq 0$	$2w - \sqrt{3}x + y \geq 0$

Table 6.2: Laue group D6 generators that touch the RFZ, and their S^3 bounds with the identity.

Via naive brute force, there are $2^{(14+14)} = 268435456$ possible combinations of active or inactive inequality boundary constraints. This can be simplified by removing combinations of active planar RFZ boundaries lacking any intersection points for either q_a or q_b. To enumerate non-empty active boundary combinations, firstly, there exists the single case of stationary points of the objective function with no zone constraints with respect to either q_a or q_b. Next, there are 14 boundary planes of the D6 RFZ which can be the sole active constraint. Next, there are 36 possible combinations of two active constraint bounding planes that have nonempty intersection (12 for lateral-top, 12 for lateral-bottom, and 12 for lateral-lateral). There are 24 intersection points of 3 bounding planes, which define the D6 dodecagonal prism RFZ. The overall problem is reduced to $(1 + 14 + 36 + 24)^2 = 5625$ combinations of boundary inequalities.

However, we can further simplify the analysis by relaxing the RFZ bounds to include additional points, and ignore solutions which are not contained in the true D6 RFZ. Instead of constraining the FZ with all of the 14 planes that surround it (12 lateral planes and 2 axial planar cuts with normal vectors parallel to the RF z-axis), we simply retain the 2 axial planes and constrain the angle below the maximum known for the D6 RFZ (a ball touching the D6 RFZ at all 24 vertices). As all points within the true RFZ are contained in the relaxed RFZ, solutions to the relaxed optimization problem only need be intersected with the true RFZ. This provides an upper bound on the proximity of $q_a q_b^{-1}$ to generator

4.

Equation (6.4) gives the relaxed faux-RFZ constraints (applied to both q_a and q_b):

$$\begin{aligned} w_i^2 + x_i^2 + y_i^2 + z_i^2 &= 1 \\ w_i(2-\sqrt{3}) - z_i &\geq 0 \\ w_i(2-\sqrt{3}) + z_i &\geq 0 \\ w_i - \frac{1+\sqrt{3}}{4} &\geq 0 \end{aligned} \tag{6.4}$$

and to maximize the dot product (minimize the angle between) of the composition $q_a q_b^{-1}$ with the generator 4 zone, we maximize the z-component of $q_a q_b^{-1}$:

$$\begin{aligned} f(w_1, x_1, y_1, z_1, w_2, x_2, y_2, z_2) &= q_{G4} \cdot q_a q_b^{-1} \\ &= -w_1 z_2 - x_1 y_2 + y_1 x_2 + z_1 w_2 \end{aligned} \tag{6.5}$$

First we recall the formulation for Lagrangian multipliers for constraints $g_k(\mathbf{x})$:

$$\nabla_{x_1,\ldots,x_n,\lambda_1,\ldots,\lambda_M} \mathcal{L}(x_1,\ldots,x_n,\lambda_1,\ldots,\lambda_M) = 0 \iff \begin{cases} \nabla f(\mathbf{x}) - \sum_{k=1}^{M} \lambda_k \nabla g_k(\mathbf{x}) = 0 \\ g_1(\mathbf{x}) = \cdots = g_M(\mathbf{x}) = 0 \end{cases} \tag{6.6}$$

yielding the following Lagrangian:

$$\begin{aligned}\mathcal{L}(w_1, x_1, y_1, z_1, w_2, x_2, y_2, z_2, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8) = {} & -w_1 z_2 - x_1 y_2 + y_1 x_2 + z_1 w_2 \\ & + \lambda_1(w_1^2 + x_1^2 + y_1^2 + z_1^2 - 1) \\ & + \lambda_2(w_2^2 + x_2^2 + y_2^2 + z_2^2 - 1) \\ & + \lambda_3(w_1(2-\sqrt{3}) - z_1) \\ & + \lambda_4(w_1(2-\sqrt{3}) + z_1) \\ & + \lambda_5(w_1 - \frac{1+\sqrt{3}}{4}) \\ & + \lambda_6(w_2(2-\sqrt{3}) - z_2) \\ & + \lambda_7(w_2(2-\sqrt{3}) + z_2) \\ & + \lambda_8(w_2 - \frac{1+\sqrt{3}}{4})\end{aligned} \tag{6.7}$$

We need to check the stationary points for all $2^6 = 64$ combinatorial cases where the six inequality constraints (controlled by $\lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8$) are or are not active. As mentioned before, many combinations will have no stationary points at all (e.g. combinations requiring points concurrently on distinct parallel planes), and some constraint relaxations will yield invalid stationary points with either q_a or q_b, or both outside of the true FZ, and these solutions are discarded. Following this procedure, there are two relaxed FZ solutions which maximize the objective term and intersect the true D6 FZ. For all points parameterized by the free variable t, the dot product with the zone 4 generator is maximally equal to 0.75:

$$(w_1, x_1, y_1, z_1) = \left(\frac{\sqrt{3}+1}{4}, \mp\sqrt{\frac{1}{2} - t^2}, t, \frac{\sqrt{3}-1}{4}\right) \tag{6.8}$$

$$(w_2, x_2, y_2, z_2) = \left(\frac{\sqrt{3}+1}{4}, t, \pm\sqrt{\frac{1}{2} - t^2}, -\frac{\sqrt{3}-1}{4}\right) \tag{6.9}$$

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8) = \left(-\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, 0, \frac{\sqrt{3}-1}{2}, 0, \frac{1}{2}, \frac{\sqrt{3}-1}{2}\right) \tag{6.10}$$

and for both q_a and q_b, eqs. (6.8) and (6.9) each parameterize two halves of a circle touching only the vertices of the D6 RFZ, with the axis of q_b offset relative to that of q_a by 90° about [001]. The angle between the vector components of q_a and q_b is twice the angle of rotation of $q_a q_b^{-1}$. A maximal dot product of 0.75 corresponds to 82.8192... degrees rotation which falls below the $\theta_{\mathrm{D6}} \approx 1.6378$ radians or 93.841°. Constraining the rotation axes of q_a and q_b yields a minimum achievable distance to the Zone 4 generator that could be within its boundary. This means that each stationary point solution to the Lagrangian with a sufficiently high dot product had to be exhaustively checked to lack a valid intersection with the Zone 4 cell.

Laue Group O

Generator	w	x	y	z	Description
1	1	0	0	0	identity operator
2*	**0**	**0**	**0**	**1**	**180 about [001]**
3*	**0**	**1**	**0**	**0**	**180 about [100]**
4*	**0**	**0**	**1**	**0**	**180 about [010]**
5	$\frac{1}{\sqrt{2}}$	0	0	$\frac{1}{\sqrt{2}}$	90 about [001]
6	$\frac{1}{\sqrt{2}}$	0	0	$-\frac{1}{\sqrt{2}}$	270 about [001]
7	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	0	90 about [100]
8	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	0	0	270 about [100]
9	$\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$	0	90 about [010]
10	$\frac{1}{\sqrt{2}}$	0	$-\frac{1}{\sqrt{2}}$	0	270 about [010]
11	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	120 about [111]
12	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	120 about [-1-1-1]
13	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	120 about [1-11]
14	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	120 about [-11-1]
15	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	120 about [-111]
16	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	120 about [1-1-1]
17	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	120 about [-1-11]
18	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	120 about [11-1]
19	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	180 about [110]
20	0	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	180 about [-110]
21	0	$\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$	180 about [101]
22	0	$-\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$	180 about [-101]
23	0	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	180 about [011]
24	0	0	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	180 about [0-11]

Table 6.3: Generators for Laue group O with zones infeasible under FZ-FZ composition bolded.

For group O, it suffices to show that generator 2 is not feasible under the composition of two fundamental zone orientations, and by symmetrical equivalence (permuting the axes via rotation retains O symmetry), then generators 3 and 4 are also infeasible under FZ-FZ composition. It is important to note that while generators 19 through 24 are feasible under FZ-FZ composition the probability of reaching them under joint uniform sampling of the RFZ is exceedingly low (less than 1 in 10^9 verified numerically). Table 6.4 gives the bounding inequalities for the RFZ of the octahedral Laue group.

Generators	RFZ Inequalities (from $\pm q$)
5-6	$w(\sqrt{2}-1) \geq \lvert z \rvert$
7-8	$w(\sqrt{2}-1) \geq \lvert x \rvert$
9-10	$w(\sqrt{2}-1) \geq \lvert y \rvert$
11-18	$w \geq \lvert x \rvert + \lvert y \rvert + \lvert z \rvert$

Table 6.4: The 14 Laue group O generators that touch the RFZ, and their combined S^3 bounds with the identity.

Again we make the relaxation that our quaternion is only bounded along z and w. The stationary points have to be similarly checked to lack any valid intersection with the Zone 2 generator.

6.3 Computational Methods

Algorithm 3 is a helper function that describes a modified **sign** function which returns 1 instead of 0 when the argument is 0. This helper function makes the following algorithms more compact. Algorithm 4 similarly makes unit quaternion have positive real part by negating it if necessary.

Algorithm 3 Neg: Modified Sign Function

```
Input: quaternion component $a$
Output: modified sign function of $a$
  if $a < 0$ then
      return $-1$
  else
      return 1
```

Algorithm 4 MakePos: Make Positive Scalar

```
Input: quaternion $w, x, y, z$ with w potentially negative
Output: quaternion with positive real component
  if $w < 0$ then
      return $(-w, -x, -y, -z)$
  else
      return $q$
```

6.3.1 Cyclic Groups: C2, C3, C4, C6

Algorithms 5 through 8 are each tantamount to switch statements along the z component of the input quaternion. Additionally the reduced quaternions are gleaned directly from the components so as to avoid invoking a separate subroutine to perform general quaternion multiplication with a symmetry operator that most likely has known zeros. The reduction for C2 in 5 checks if the absolute value of the z-component is less than w, and if it is not then it modifies the quaternion via the only rotational operation in the group.

Algorithm 5 FZ Orientation Reduction for C2 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in C2 FZ
  if $|z| < w$ then
      return $q$
  else
      return neg($z$)$(-z, y, -x, w)$
```

In 6, the 120-degree rotational symmetry changes the thresholds for the z-component relative to w, but now the sign of the z component must also be checked if the quaternion is not already in the FZ to discern if a +120 or -120 degree rotation is required to reduce the quaternion.

Algorithm 6 FZ Orientation Reduction for C3 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in C3 FZ
  if $|z| < \frac{w}{\sqrt{3}}$ then
      return $q$
  else
      return $\frac{1}{2}(w, x, y, z) + \mathrm{neg}(z)\frac{\sqrt{3}}{2}(z, -y, x, -w)$
```

Algorithm 7 FZ Orientation Reduction for C4 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in C4 FZ
  if $|z| \leq (\sqrt{2} - 1)w$ then
      return $q$
  else if $|z| > (\sqrt{2} + 1)w$ then
      return $\mathrm{neg}(z)(z, -y, x, -w)$
  else
      return $\frac{1}{\sqrt{2}}((w, x, y, z) + \mathrm{neg}(z)(z, -y, x, -w))$
```

Algorithm 8 FZ Orientation Reduction for C6 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in C6 FZ
  if $|z| \leq (2 - \sqrt{3})w$ then
    return $q$
  else if $|z| > (2 + \sqrt{3})w$ then
    return $\text{neg}(z)(z, -y, x, -w)$
  else if $|z| > w$ then
    return $\frac{1}{2}(w, x, y, z) + \text{neg}(z)\frac{\sqrt{3}}{2}(z, -y, x, -w)$
  else
    return $\frac{\sqrt{3}}{2}(w, x, y, z) + \text{neg}(z)\frac{1}{2}(z, -y, x, -w)$
```

6.3.2 Dihedral Groups: D2, D3, D4, D6

The Dihedral groups require the most computational work to reduce relative to the cyclic groups and T/O. This is due to the need for a preliminary check of which lateral face of the RFZ the xy components face. For D2, given in 9, this is alleviated by the fact that the lateral face may be determined by which component has the largest magnitude.

Algorithm 9 FZ Orientation Reduction for D2 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in D2 FZ
  if $w > \max(|x|, |y|, |z|)$ then
    return $q$
  else if $|z| = \max(|x|, |y|, |z|)$ then
    return $\text{neg}(z)(z, -y, x, -w)$
  else if $|y| = \max(|x|, |y|, |z|)$ then
    return $\text{neg}(y)(y, z, -w, -x)$
  else
    return $\text{neg}(x)(x, -w, -z, y)$
```

Figure 6.1 shows uniformly drawn (Shoemake) orientations plotted as Rodrigues-Frank

(RF) vectors with the fundamental zone (the identity) for D2 flanked by the three other symmetry operators. As all three of the other generators reside at infinity, each Voronoi cell when projected and drawn in RF space has two separated antipodal regions as these 180° rotation generators are at infinity in this representation.

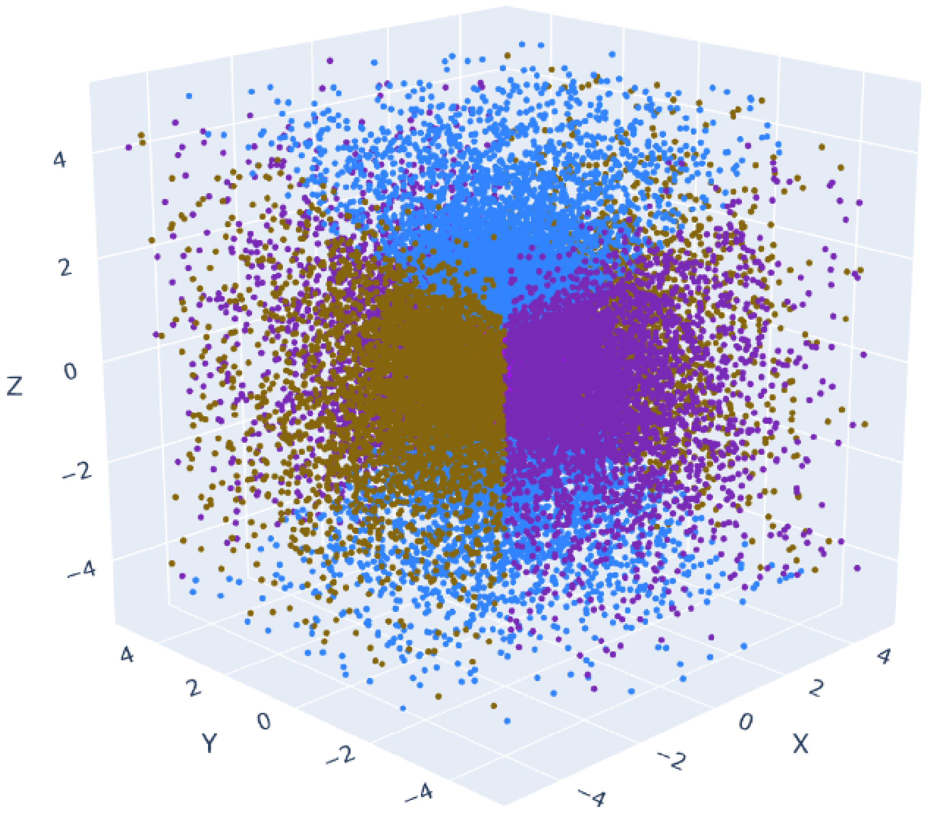

Figure 6.1: Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the three other Voronoi cells flanking the cube shaped fundamental zone of D2.

However, starting with D3, a separate computations must be carried out in the case of lateral RFZ boundaries with normals not parallel to a coordinate axis: computing $\sqrt{3}|y|$ to potentially then compute $x_{\text{rot}} \leftarrow \frac{\sqrt{3}}{2}|y| + \frac{1}{2}|x|$ in algorithm 10. For D3 two of three lateral boundaries require these computations. One important note here is that the helper sign function neg is used in two ways here. Firstly, it denotes the check to return the scalar part of the quaternion to be positive if needed as in: $\text{neg}(x)(x, -w, -z, y)$. For clarity this could also be expressed as $(|x|, -w\ \text{neg}(x), -z\ \text{neg}(x), y\ \text{neg}(x))$. Secondly, neg compactly represents a separate branching of two possibilities in the return value as in: $\frac{1}{2}(w, x, y, z) + \text{neg}(z)\frac{\sqrt{3}}{2}(z, -y, x, -w)$.

Algorithm 10 FZ Orientation Reduction for D3 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in D3 FZ
  if $|x| > \sqrt{3}|y|$ then
    $x_{\text{rot}} \leftarrow |x|$
  else
    $x_{\text{rot}} \leftarrow \frac{\sqrt{3}}{2}|y| + \frac{1}{2}|x|$
  if $|z| < \frac{w}{\sqrt{3}}$ and $x_{\text{rot}} < w$ then
    return $q$
  else if $\frac{1}{2}w + \frac{\sqrt{3}}{2}|z| > x_{\text{rot}}$ then
    return $\frac{1}{2}(w, x, y, z) + \text{neg}(z)\frac{\sqrt{3}}{2}(z, -y, x, -w)$
  else if $|x| > \sqrt{3}|y|$ then
    return $\text{neg}(x)(x, -w, -z, y)$
  else
    $S \leftarrow \text{neg}(xy)$
    return $\text{neg}(-Sx - \sqrt{3}y)\frac{1}{2}(-Sx - \sqrt{3}y, Sw - \sqrt{3}z, Sz + \sqrt{3}w, -Sy + \sqrt{3}x)$
```

Figure 6.2 shows uniformly drawn orientations plotted as Rodrigues-Frank (RF) vectors with the fundamental zone (the identity) for D3 flanked by the five other symmetry operators: two z-axial and three laterally positioned. As all three of the lateral generators reside at infinity with 180 degree rotations, each Voronoi cell when projected and drawn in RF space has two separated antipodal regions. The two z-axial ones sit at ±120 about z and thus are not broken apart in this projection. This algorithm 10 begins by shifting the point into the 180 about x-axis zone or either of the other laterals zones and then doing a unified check first against the fundamental zone and then against the z-axial zones.

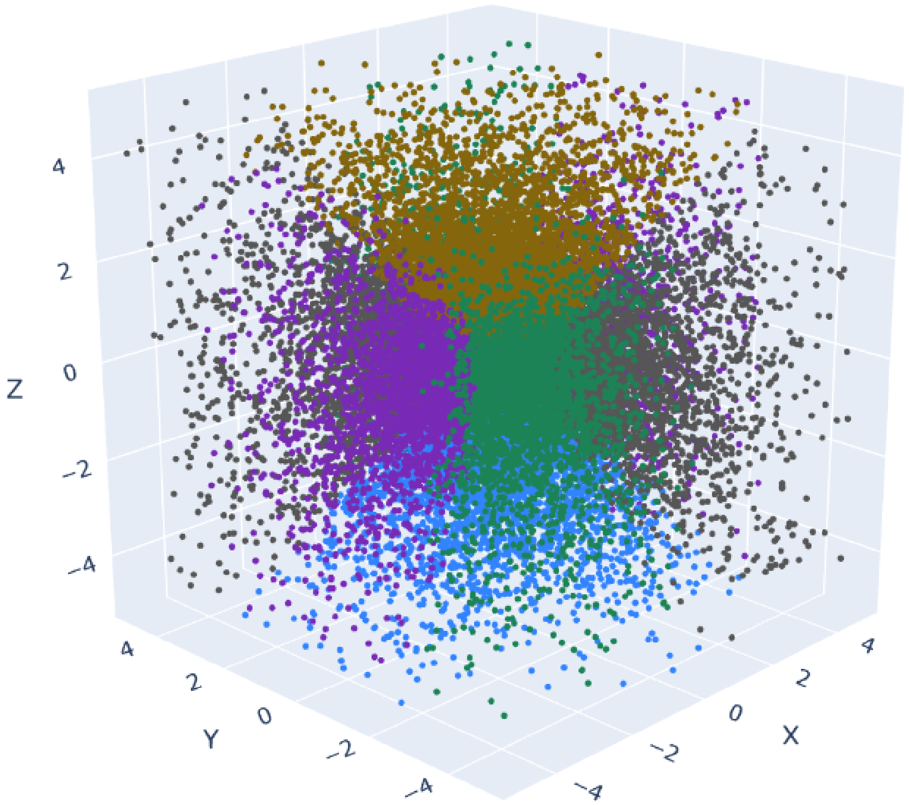

Figure 6.2: Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the five other Voronoi cells flanking the hexagonal prism shaped fundamental zone of D3.

From D2 to D3 the fundamental zone changed from a cube to a hexagonal prism, and with D4 it is an octagonal prism flanked by 4 different lateral Voronoi cells and two z-axial ±90 degree rotation zones, and a third non-touching zone at 180 about the z-axis. Similar to that of D3, the D4 algorithm 11 first reduces the rotation about the z-axis to quickly and in a unified manner check the planar boundaries with the fundamental zone and then with the lateral flanking zones.

Algorithm 11 FZ Orientation Reduction for D4 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in D4 FZ
  $m, n \leftarrow \max(|x|, |y|), \min(|x|, |y|)$
  if $m > (\sqrt{2}+1)n$ then
      $y\_\text{rot} \leftarrow m$
  else
      $y\_\text{rot} \leftarrow \frac{1}{\sqrt{2}}(m+n)$
  if $|z| < (\sqrt{2}-1)w$ and $y\_\text{rot} < w$ then
      return $q$
  else if $|z| > (\sqrt{2}+1)w$ and $|z| > y\_\text{rot}$ then
      return $\text{neg}(z)(-z, y, -x, w)$
  else if $|z| > (\sqrt{2}-1)w$ and $\frac{1}{\sqrt{2}}(|z|+w) > y\_\text{rot}$ then
      return $\frac{1}{\sqrt{2}}((w, x, y, z) + \text{neg}(z)(z, -y, x, -w))$
  else if $m > (\sqrt{2}+1)n$ then
      if $|y| > |x|$ then
          return $\text{neg}(y)(-y, -z, w, x)$
      else
          return $\text{neg}(x)(-x, w, z, -y)$
  else
      return $\text{MakePos}(\frac{1}{\sqrt{2}}(\text{neg}(xy)(-x, w, z, -y) + (-y, -z, w, x)))$
```

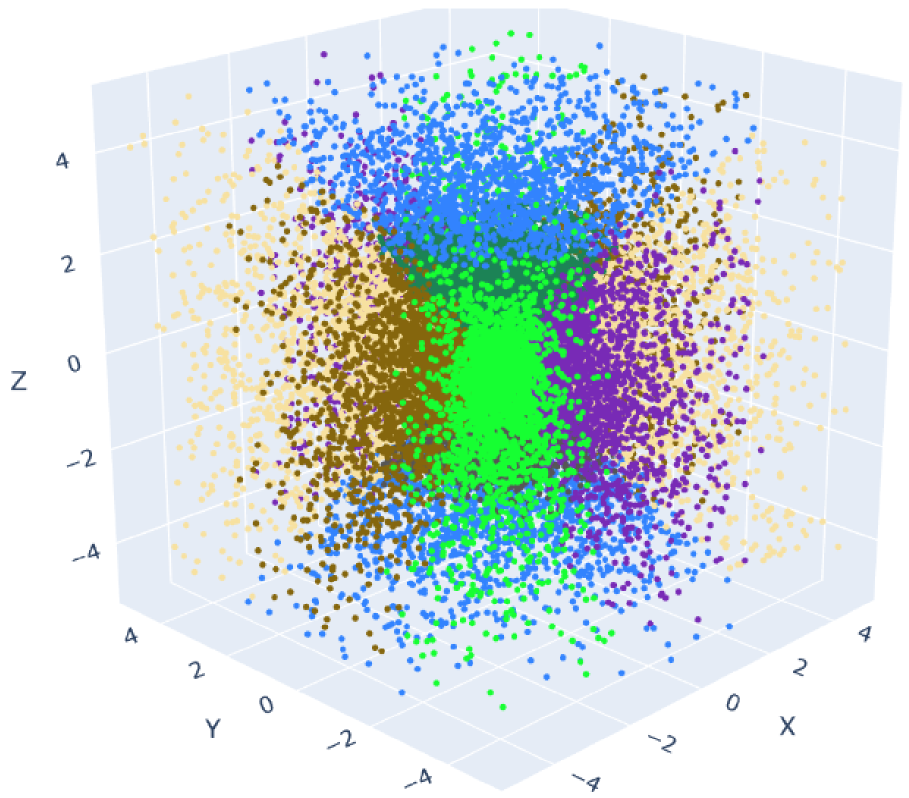

Figure 6.3: Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the seven other Voronoi cells surrounding the octagonal prism shaped fundamental zone of D4.

D6 is similar to D4 with two more z-axial and two more lateral Voronoi cells, as seen in figure 6.4. Algorithm 12 takes a similar path as the other dihedral groups to the achieve the reduction.

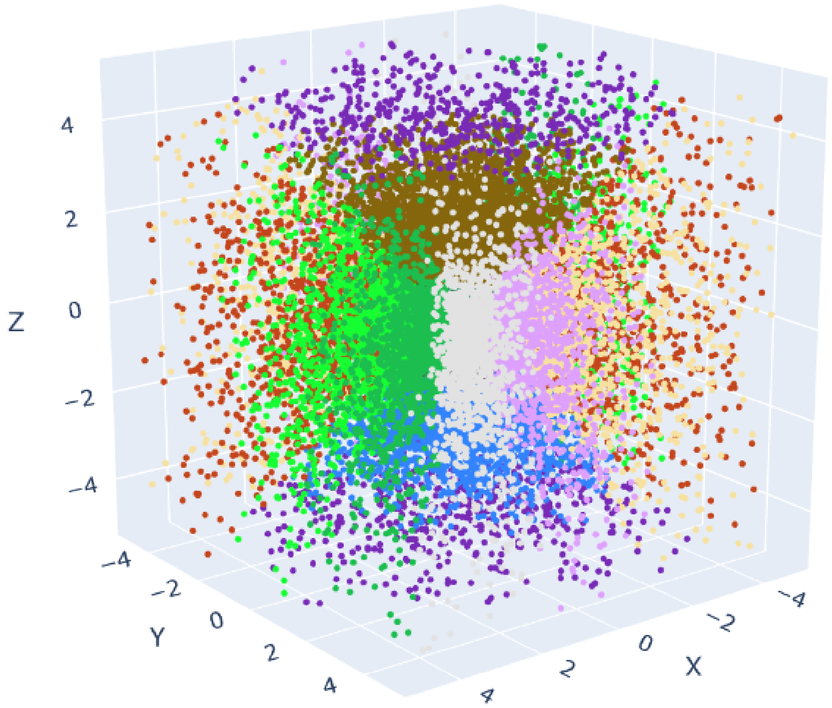

Figure 6.4: Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the eleven other Voronoi cells surrounding the dodecagonal prism shaped fundamental zone of D6.

Algorithm 12 FZ Orientation Reduction for D6 Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in D6 FZ
  $m, n \leftarrow \max(|x|, |y|), \min(|x|, |y|)$
  if $m > (2 + \sqrt{3})n$ then
    $y\_\text{rot} \leftarrow m$
  else
    $y\_\text{rot} \leftarrow \frac{\sqrt{3}}{2}m + \frac{1}{2}n$
  if $|z| < (2 - \sqrt{3})w$ and $y\_\text{rot} < w$ then
    return $q$
  else if $|z| < w$ and $\frac{1}{2}(|z| + \sqrt{3}w) > y\_\text{rot}$ then
    return $\frac{1}{2}(\sqrt{3}w + |z|, \sqrt{3}x - \text{neg}(z)y, \sqrt{3}y + \text{neg}(z)x, \sqrt{3}z - \text{neg}(z)w)$
  else if $|z| < (2 + \sqrt{3})w$ and $\frac{1}{2}(\sqrt{3}|z| + w) > y\_\text{rot}$ then
    return $\frac{1}{2}(w + \sqrt{3}|z|, x - \text{neg}(z)\sqrt{3}y, y + \text{neg}(z)\sqrt{3}x, z - \text{neg}(z)\sqrt{3}w)$
  else if $|z| > y\_\text{rot}$ then
    return $\text{neg}(z)(z, -y, x, -w)$
  else if $m > (2 + \sqrt{3})n$ then
    if $|y| > |x|$ then
      return $\text{neg}(y)(y, z, -w, -x)$
    else
      return $\text{neg}(x)(x, -w, -z, y)$
  else
    $S \leftarrow \text{neg}(xy)$
    if $|y| > |x|$ then
      return $\text{MakePos}(\frac{1}{2}(-Sx - \sqrt{3}y, Sw - \sqrt{3}z, Sz + \sqrt{3}w, -Sy + \sqrt{3}x))$
    else
      return $\text{MakePos}(\frac{1}{2}(-\sqrt{3}Sx - y, \sqrt{3}Sw - z, \sqrt{3}Sz + w, -\sqrt{3}Sy + x))$
```

6.3.3 Groups O and T

Group T is aided by the fact that the $x = 0$, $y = 0$, and $z = 0$ planes are the boundaries dividing several of the Voronoi cells, and the sign may be quickly used to construct the correct fundamental zone quaternion seen in the last return statement of Algorithm 13. Further, figure 6.5 displays a coloring of a uniformly drawn sampling of Rodrigues-Frank vectors with the per coordinate axis planar division boundaries between several of the zones identifiable, except for the 180° rotations about each of the primary axes. Those zones are divided into two regions in the projection and unlike the fundamental zone and the eight other 120° rotation Voronoi cells.

Algorithm 13 FZ Orientation Reduction for T Symmetry

```
Input: unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
Output: Reduced quaternion $\mathbf{q}$ in T FZ
  if $|x| + |y| + |z| < w$ then
      return $q$
  else if $|z| > w + |x| + |y|$ then
      return $\text{neg}(z)(z, -y, x, -w)$
  else if $|x| > w + |y| + |z|$ then
      return $\text{neg}(x)(x, -w, -z, y)$
  else if $|y| > w + |x| + |z|$ then
      return $\text{neg}(y)(y, z, -w, -x)$
  else
      return $\frac{1}{2}(w + |x| + |y| + |z|,$
                $x - \text{neg}(x)w + \text{neg}(y)z - \text{neg}(z)y,$
                $y - \text{neg}(x)z - \text{neg}(y)w + \text{neg}(z)x,$
                $z + \text{neg}(x)y - \text{neg}(y)x - \text{neg}(z)w)$
```

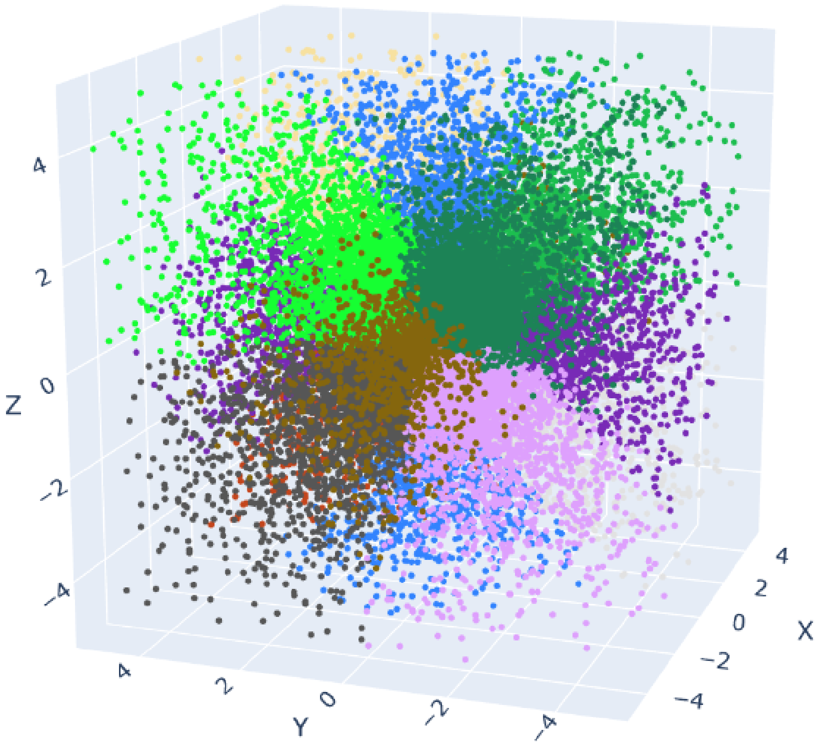

Figure 6.5: Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the eleven other Voronoi cells surrounding the octahedron shaped fundamental zone of T.

Lastly, Octahedral symmetry has the greatest number of operators of any of the Laue classes. The decision tree based reduction takes a similar path to the dihedral reductions in first collapsing the points to a single octant by taking the absolute value and additionally sorting the values into small, medium, and large. This allows for a unified check against each of the rotations around the z-axis again by magnitude. After finding which type of rotation (e.g. a 180° about a $< 100 >$ direction), the particular signs or combination of signs can be used to distinguish to which Voronoi cell a point belongs. Figure 6.6 shows the 23 surrounding non-fundamental zone Voronoi cells.

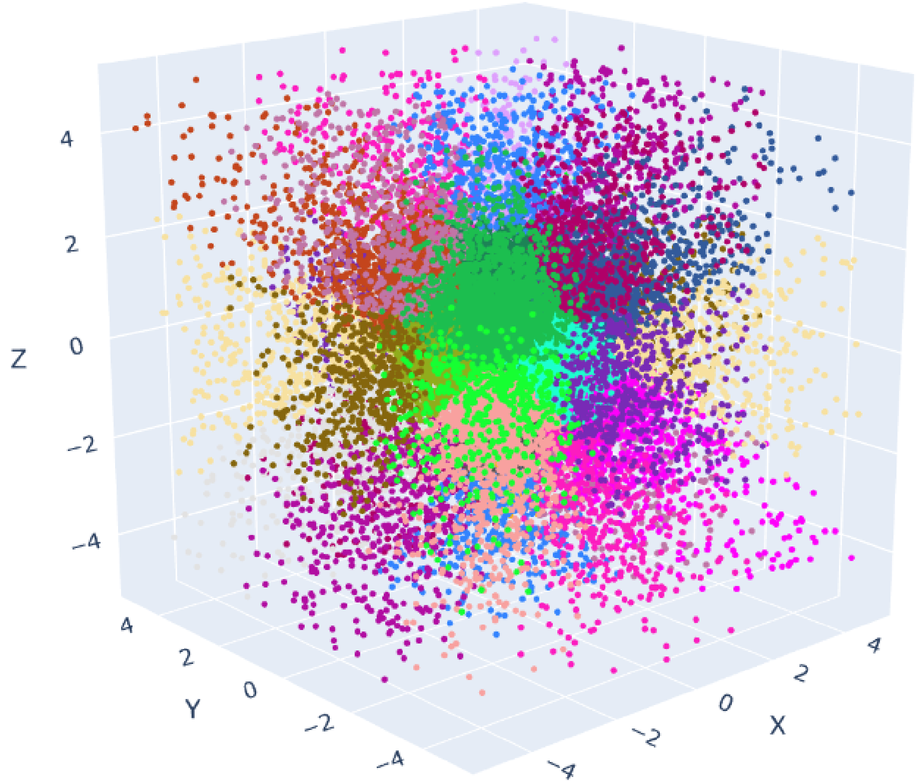

Figure 6.6: Uniformly sampled orientations plotted as Rodrigues-Frank vectors with truncated axes showing the 23 other Voronoi cells surrounding the truncated cube shaped fundamental zone of O.

Algorithm 14 FZ Orientation Reduction for O Symmetry

```
**Input:** unit quaternion $\mathbf{q} = (w, x, y, z)$ with $w \geq 0$
**Output:** Reduced quaternion $\mathbf{q}$ in O FZ
  $\mathrm{big}, \mathrm{mid}, \mathrm{sml} \leftarrow \mathrm{sort}(|x|, |y|, |z|)$
  $\mathrm{zone} \leftarrow \textbf{DetermineGenType}(\mathbf{q}, \mathrm{big}, \mathrm{mid}, \mathrm{sml})$
  $\mathbf{q} \leftarrow \textbf{ReduceQuaternion}(\mathbf{q}, \mathrm{zone}, \mathrm{big}, \mathrm{mid}, \mathrm{sml})$
  **return q**
```

Algorithm 15 DetermineOGenType

```
Input: quaternion $\mathbf{q} = (w, x, y, z)$, big, mid, sml
Output: zone
  if big $< (\sqrt{2} - 1)w$ then
      if $w >$ sml + mid + big then
          zone $\leftarrow 0$
      else
          zone $\leftarrow 3$
  else if big $< (\sqrt{2} + 1)w$ then
      if $(\sqrt{2} - 1)(w + \text{big}) > \text{sml} + \text{mid}$ then
          if $w >$ mid then
              zone $\leftarrow 1$
          else
              zone $\leftarrow 6$
      else
          if $w + \text{sml} > (\sqrt{2} - 1)(\text{mid} + \text{big})$ then
              zone $\leftarrow 3$
          else
              zone $\leftarrow 6$
  else
      if big $> \frac{1}{2}(w + \text{sml} + \text{mid} + \text{big})$ then
          if big $> (\sqrt{2} + 1)$mid then
              zone $\leftarrow 2$
          else
              zone $\leftarrow 6$
      else
          if $w + \text{sml} > (\sqrt{2} - 1)(\text{mid} + \text{big})$ then
              zone $\leftarrow 3$
          else
              zone $\leftarrow 6$
```

Algorithm 16 ReduceOQuaternion

```
Input: quaternion $\mathbf{q} = (w, x, y, z)$, zone, big, mid, sml
Output: Reduced quaternion $\mathbf{q}$
  if zone == 0 then
      $\mathbf{q} \leftarrow \mathbf{q}$
  else if zone == 2 then
      $\mathbf{q} \leftarrow \text{neg}(\text{big})(-\text{big}, \text{mid}, -\text{sml}, w)$  if  big == $|x|$
      $\mathbf{q} \leftarrow \text{neg}(\text{big})(-\text{big}, -z, w, x)$  if  big == $|y|$
      $\mathbf{q} \leftarrow \text{neg}(\text{big})(-\text{big}, y, -x, w)$  if  big == $|z|$
  else if zone == 1 then
      $\mathbf{q} \leftarrow \frac{1}{\sqrt{2}}(w + \text{big}, x - \text{neg}(\text{big})w, y - \text{neg}(\text{big})z, z + \text{neg}(\text{big})y)$  if  big == $|x|$
      $\mathbf{q} \leftarrow \frac{1}{\sqrt{2}}(w + \text{big}, x + \text{neg}(\text{big})z, y - \text{neg}(\text{big})w, z - \text{neg}(\text{big})x)$  if  big == $|y|$
      $\mathbf{q} \leftarrow \frac{1}{\sqrt{2}}(w + \text{big}, x - \text{neg}(\text{big})y, y + \text{neg}(\text{big})x, z - \text{neg}(\text{big})w)$  if  big == $|z|$
  else if zone == 3 then
      return $\frac{1}{2}(w + |x| + |y| + |z|,$
          $x - \text{neg}(x)w + \text{neg}(y)z - \text{neg}(z)y,$
          $y - \text{neg}(x)z - \text{neg}(y)w + \text{neg}(z)x,$
          $z + \text{neg}(x)y - \text{neg}(y)x - \text{neg}(z)w)$
  else
      if sml == $|z|$ then
          $\mathbf{q} \leftarrow \text{MakePos}(\frac{1}{\sqrt{2}}(-\text{neg}(xy)x - y, \text{neg}(xy)w - z, \text{neg}(xy)z + w, -\text{neg}(xy)y + x))$
      else if sml == $|y|$ then
          $\mathbf{q} \leftarrow \text{MakePos}(\frac{1}{\sqrt{2}}(-\text{neg}(xz)x - z, \text{neg}(xz)w + y, \text{neg}(xz)z - x, -\text{neg}(xz)y + w))$
      else
          $\mathbf{q} \leftarrow \text{MakePos}(\frac{1}{\sqrt{2}}(-\text{neg}(yz)y - z, -\text{neg}(yz)z + y, \text{neg}(yz)w - x, \text{neg}(yz)x + w))$
  return $\mathbf{q}$
```

6.4 Reduction Benchmarks

Figure 6.7 summarizes the final speedup gain from using these decision tree based algorithms instead of either exhaustive angle or bounds methods in both the general case of an orientation uniformly sampled from SO(3) and in the case of reducing the orientations resulting from the composition of two quaternions already residing in the fundamental zone for a given Laue group. In the latter case, groups D3 and D4 are not advantageous over the angle method and that approach should instead be utilized. However, for cubic symmetry groups O and T, a significant speedup is realized by using decision tree based methods. As compared to the angle exhaustion method roughly a twofold speedup is observed for cubic symmetry. For group D6 a less significant 25% to 40% improvement was observed due to the additional computation complexity of checking off-axis bounds and the relatively lower number of symmetry operators.

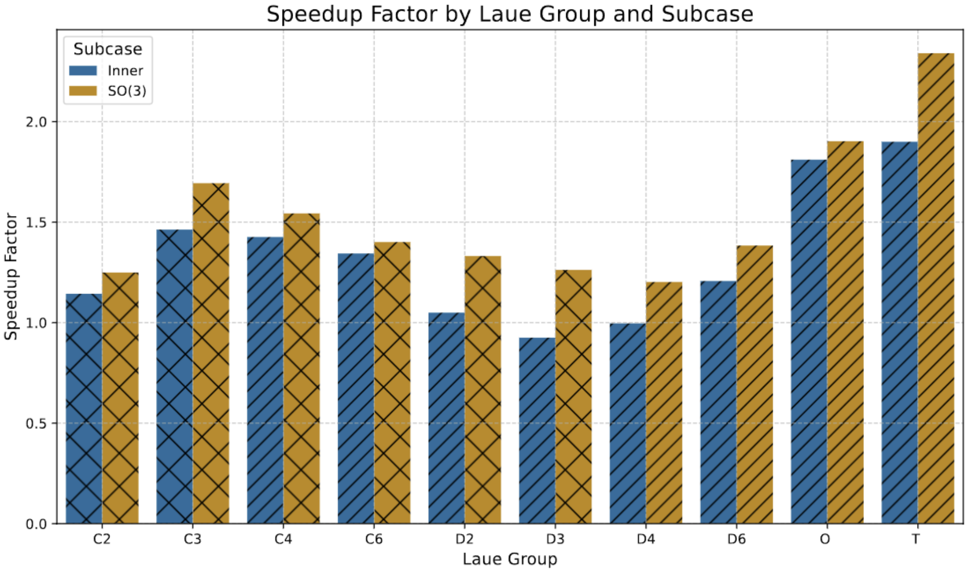

Figure 6.7: Bar plot showing the relative speedup of the tree based method compared to the best of either the angle and bounds methods for a given Laue group. Single hash indicates comparison with the angle method while double hash indicates the bounds method.

CHAPTER VII

Conclusions and Future Work

7.1 Summary

The primary objective of this thesis was to provide a collection of numerical tools for the analysis of crystallographic data, especially EBSD data. In particular, this work sought to alleviate the big data and multimodal challenges posed by 3D serial sectioning experiments in the SEM in three ways: dynamic sampling for scanned images, EBSD orientation indexing acceleration, and decision tree algorithms for crystallographic fundamental zone reductions. In addition, two preliminary results, one using key points and another using harmonics, in the area of multimodal image registration were presented. In general, we sought to overcome scale challenges and the multifaceted nature of microstructure characterization in the SEM.

The first tool, nearest neighborhood based dynamic sampling was proposed to provide a rapid suggestion of new image pixels to acquire under a simple heuristic. The end result was a dramatic reduction in the number of pixels required to measure functionally identical data in EBSD and BSE imaging modalities, as well as optical modalities. The second tool, an application of quantization and PCA to dictionary indexing of orientations in EBSD dramatically reduced the computational burden of the historically slowest but most noise-robust method of orientation indexing. Thereafter, two approaches to multimodal image registration and their preliminary results were presented. Finally, decision-tree based algorithms for the reduction of crystallographic orientations to their respective Laue classes were proposed.

7.2 Hypotheses Revisited

Revisiting the hypotheses, we begun with "Rasterization-based scans, (e.g. BSE micrographs or EBSD orientation maps), acquired in the electron microscope can be subsampled using dynamic sampling, and then infilled, yielding at twofold reduction in scan time while maintaining an error rate (grayscale pixel values different by 10% relative deviation or misorientations exceeding 3°) not exceeding 1%". This hypothesis is supported by the results in Chapter III demonstrating a heuristic-based dynamic sampling algorithm operating at speeds sufficient for electron microscopy applications. The target thresholds for accuracy were maintained across a variety of settings for grayscale algorithms, while a single scan was used to support this claim for orientation data from EBSD.

The hypothesis that principal component analysis can be used to improve the indexing speed by an order of magnitude and accuracy for EBSD applications was supported by analysis on a Nickel test dataset with varied noise levels in Chapter IV. We find that the noise robustness was improved by PCA and that the speed threshold from the hypothesis was met. These findings must be taken in context as the pattern size was relatively small at 60×60. Further, at high pattern noise levels the required number of PCA components needed to maintain indexing accuracy at the level of conventional dictionary indexing brought the relative speed improvement down to roughly 8-fold. For much larger patterns the benefit of using PCA is greatly magnified. For example, the ratio of 1000 PCA components to the number of pixels in a 60×60 pattern versus a 256×256 pattern is 28% versus 1.5%. Thus, these findings are rather conservative in estimating the benefit of low rank approximations to electron backscatter diffraction signals via PCA under the assumption that the higher resolution patterns can index under a comparable number of PCA components, warranting further investigation.

The hypothesis that fundamental zone reductions can benefit from decision tree based algorithms which systematically identify the Voronoi cell to which an orientation belongs was tested in Chapter VI. Compared to the other two methods for reduction, the decision tree algorithm did not necessarily bring benefit for all of the Laue classes. However, for the Octahedral and Tetrahedral symmetries in particular, an approximately two-fold improvement in reduction speed was observed in both the general case of orientations uniformly drawn over SO(3) and in the sub-case where two uniformly drawn fundamental zone orientations were composed together and then needed reduction to the fundamental zone. Thus, one cannot say that the hypothesis is verified for all Laue classes. Certain Laue classes have more expensive orientation reductions due to Rodrigues vectors having planar boundaries that must be checked which do not coincide with a simple sum of the magnitude of the components or sign check of an individual vector component.

7.3 Future Outlook

Beginning with dynamic sampling, the first major direction to pursue is the implementation of the proposed sampling heuristic on real world hardware. This task requires detailed knowledge of the control board for the SEM or other scanned microscopy equipment. Although this is considered beyond the scope of this thesis, several attempts to manually control the beam position were made on one SEM using a vendor-provided software interface resulting in a latency three orders of magnitude slower than the unsupervised dynamic sampling computations.

For the topic of PCA-based dictionary indexing of EBSD patterns there are many research directions to consider, particularly for scaling to higher pattern resolutions. One naive approach to avoid large covariance matrices for larger patterns would be to bin the patterns prior to eigendecomposition and to upscale the resulting eigenvectors to the same size as the original patterns. This however, would implicitly ignore any information content in the patterns at a frequency above the binning rate. In lieu of this naive approach, the eigendecomposition could be replaced by a streamed PCA algorithm which only retains the top k principal components [117, 118]. However, neither approach leverages the prior knowledge that the patterns arise from the convolution over SO(3) or one of its quotient groups between a detector shaped mask projected onto $\mathbb{S}^2$ and the master pattern on $\mathbb{S}^2$. Seeking out closed form expressions for the eigenvectors as a function of the master pattern and detector projection center, in the language of spherical and generalized spherical harmonics, could alleviate the computational burden of higher resolution patterns without losing finer details.

Another exciting area for future research is the potential integration of dynamic sampling techniques with advanced EBSD indexing methods. For EBSD orientation maps, often the local orientation gradient is relatively small, and the maps have low overall complexity, suggesting subsampling could be tenable. Joining the unsupervised dynamic sampling framework with real time dictionary indexing could make the theoretical time savings presented in this thesis possible.

With regards to the Edgeworth approach to estimating the marginal and joint entropy of a underlying data distribution, two key approximations were made. Firstly, a Taylor series expansion of $(1+x)\log(1+x)$ was truncated to two terms about $x = 0$ as $x + \frac{x^2}{2}$ to make the integral over $\mathbb{R}$ and then over $\mathbb{R}^2$ tractable (see equations C.7 for marginals and C.37-C.39 for joint distributions in Appendix C in citation [106]). The second approximation was taking the truncated Edgeworth series. For the fourth order series, the truncation is in equations C.1 and C.52. It could be fruitful to examine the difference between a second and third order Taylor series truncation at a given Edgeworth series truncation and to compare the end result to binned mutual information on exemplar datasets.

While optimized ECMIF and CMIF implementations which use FFT on the GPU via PyTorch were originally developed with the aim of multimodal image registration, these routines appear to be well catered towards exhaustively searching for the correct pattern center for a given EBSD dataset. By projecting the periphery around a simulated EBSP in addition to the EBSP itself for an entire EBSD dictionary, cross-correlation can be used to densely compute normalized dot products, binned mutual information, or Edgeworth mutual information as a function of discrete changes in the pattern center X and Y coordinate. The original experimental pattern can be resampled to simulate the corresponding pattern center Z coordinate change in the simulated pattern. An exhaustive search of the projection center would help facilitate more widespread adoption of dictionary-based orientation indexing in EBSD as it is a crucial input parameter for accurate indexing. In terms of applications for multimodal image registration, image shifts have already seen generalization via non-commutative harmonic analysis to other groups such as SE(2) and SIM(2) [119,120]. Cross correlation based on convolution theorems for these frameworks can be similarly used to compute mutual information over these non-commutative groups. For a detailed example with SE(2) see chapter 3 (particularly subsection 3.3.3) of the dissertation of Erik Franken [121].

Another area of active research is the construction of embeddings into higher dimensional Euclidean space for orientations so that statistics and distances may be immediately estimated without explicitly considering the Laue class symmetry operators for a given crystalline orientation. Additionally, these embeddings admit a natural application for machine learning, wherein neural networks and other regression tools treat orientations in the embedding space. The foundational work in this area began with vector space embeddings of orthogonal axial frames (solely applicable to orthorhombic crystals) [122, 123]. That research gave way to locally isometric embeddings for various discrete quotient groups of SO(3) [20]. One interesting direction to explore would be to relax the need for a local isometry which facilitates statistics over the discrete quotient groups of SO(3), and merely seek to construct the compact simply connected 4-manifold embedding in $\mathbb{R}^5$ which is guaranteed to exist for every compact orientable 3-manifold (see theorem 3 in citation [124]). The end result would be 5-dimensional embeddings for each of the quotient groups of SO(3), analogous to those developed for SO(N) itself (as proposed in citation [125]). The primary application would be a simply connected parameterization of minimal dimension for crystallographic orientations with generative machine learning applications for textures. An alternative route towards the same ends for machine learning applications would be to design SO(3) quotient group invariant and equivariant neural network architectures as has been achieved for SO(3) itself [126, 127].

Lastly, further exploration is warranted of the implications of inaccessible Voronoi cells on the 3-sphere in the context of reducing misorientations to the equivalent unique disorientations for each combination of point groups. This could help alleviate the computational burden of computing heterophase interface disorientations. Such an extension would be welcome in the context of advanced manufacturing techniques like additive manufacturing, where complex microstructures with multiple phases can be common.

APPENDIX A

Lie Algebras and Lie Groups

One very useful construct in mathematics for many engineering applications is the Lie algebra. A Lie algebra is a vector space $\mathfrak{g}$ over a field $\mathbf{F}$ possessing an additional operation called the Lie bracket:

$$[x, y] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g} \tag{A.1}$$

which satisfies the following:

$$\begin{aligned} [x, x] &= 0 \\ [x, [y, z]] + [y, [z, x]] + [z, [x, y]] &= 0 \\ [ax + by, z] &= a[x, z] + b[y, z] \\ [z, ax + by] &= a[z, x] + b[z, y] \end{aligned} \tag{A.2}$$

for all x, y, and z in $\mathfrak{g}$ and scalars a, b in $\mathbf{F}$. A Lie algebra is often associated with a Lie group, which is colloquially a continuous symmetry. Examples include the rotational groups SO(3) and SO(2).

APPENDIX B

Polynomial Fits for the Homochoric Inverse

Equations (B.1) and (B.2) provide the polynomial fit that transforms the squared modulus of the homochoric vector residing within the 3D ball with radius $(\frac{3\pi}{4})^{\frac{1}{3}}$ to the cosine of half of the rotation angle ω:

$$
\begin{aligned}
||h||_2 &= h_x^2 + h_y^2 + h_z^2 \\
\left[\cos\frac{\omega}{2}\right]_{\text{EMsoft}} =& 0.9999999999999968 - 0.49999999999986866||h||_2 \\
&- 0.025000000000632055||h||_2^2 - 0.003928571496460683||h||_2^3 \\
&- 0.0008164666077062752||h||_2^4 - 0.00019411896443261646||h||_2^5 \\
&- 0.00004985822229871769||h||_2^6 - 0.000014164962366386031||h||_2^7 \\
&- 1.9000248160936107 \times 10^{-6}||h||_2^8 - 5.72184549898506 \times 10^{-6}||h||_2^9 \\
&+ 7.772149920658778 \times 10^{-6}||h||_2^{10} - 0.00001053483452909705||h||_2^{11} \\
&+ 9.528014229335313 \times 10^{-6}||h||_2^{12} - 5.660288876265125 \times 10^{-6}||h||_2^{13} \\
&+ 1.2844901692764126 \times 10^{-6}||h||_2^{14} + 1.1255185726258763 \times 10^{-6}||h||_2^{15} \\
&- 1.3834391419956455 \times 10^{-6}||h||_2^{16} + 7.513691751164847 \times 10^{-7}||h||_2^{17} \\
&- 2.401996891720091 \times 10^{-7}||h||_2^{18} + 4.386887017466388 \times 10^{-8}||h||_2^{19} \\
&- 3.5917775353564864 \times 10^{-9}||h||_2^{20}
\end{aligned}
\tag{B.1}
$$

$$\begin{aligned}\left[\cos\frac{\omega}{2}\right]_{\text{Kikuchipy}} =& 1.0000000000018852 - 0.5000000002194847||h||_2 \\ & - 0.024999992127593126||h||_2^2 - 0.003928701544781374||h||_2^3 \\ & - 0.0008152701535450438||h||_2^4 - 0.0002009500426119712||h||_2^5 \\ & - 0.00002397986776071756||h||_2^6 - 0.00008202868926605841||h||_2^7 \\ & + 0.00012448715042090092||h||_2^8 - 0.0001749114214822577||h||_2^9 \\ & + 0.0001703481934140054||h||_2^{10} - 0.00012062065004116828||h||_2^{11} \\ & + 0.000059719705868660826||h||_2^{12} - 0.00001980756723965647||h||_2^{13} \\ & + 0.000003953714684212874||h||_2^{14} - 0.00000036555001439719544||h||_2^{15}\end{aligned} \tag{B.2}$$

Three additional polynomial fits are reported for convenience and are included in ebsdtorch:

Table B.1: Polynomial Fits for EBSDtorch

Term	Degree 8	Degree 10	Degree 15
1	1.0000000000000009	1.0000000000000000	1.0000000000000011
2	−4.9999943403867775e-01	−4.9999997124013285e-01	−5.0000000001869205e-01
3	−2.5015165060149020e-02	−2.5001181866044025e-02	−2.4999998320881969e-02
4	−3.8120131548551729e-03	−3.9144209820521038e-03	−3.9286161535054291e-03
5	−1.2106188330642162e-03	−8.9320268104539483e-04	−8.1591329178494823e-04
6	+4.9329295993155416e-04	+3.1181024286083695e-05	−1.9799194822372066e-04
7	−7.0089385526450620e-04	−4.3961032788396477e-04	−3.2864198721441472e-05
8	+3.0979774923589078e-04	+3.9657471727506439e-04	−6.3883724580092850e-05
9	−7.3023474963298843e-05	−2.6379945050586932e-04	+9.8721769583183007e-05
10		+9.1185355979587159e-05	−1.4930153501324233e-04
11		−1.4875867805692529e-05	+1.5269890325177250e-04
12			−1.1245608697316625e-04
13			+5.7371245559076731e-05
14			−1.9467448854651149e-05
15			+3.9512697022509176e-06
16			−3.6952915441964861e-07

Table B.2 lists the absolute error mean and maximum when sampling $100,000$ orientations via the Shoemake method for quaternions. The orientations are converted to scaled axis angle vectors, then to homochoric vectors and back to scaled axis angle vectors, and the round trip errors are tabulated. The purpose of listing the coefficients above for the 8 degree polynomial was to provide a polynomial fit for FP32 precision.

Table B.2: Mean and Max Absolute Errors for Various Polyfits for the Homochoric inverse using FP64 and FP32 Precision

Polynomial Fit	FP64 Mean	FP64 Max	FP32 Mean	FP32 Max
Newton's Method	2.705895e-16	5.731526e-15	1.099045e-07	2.093613e-06
20 Deg (EMsoft)	1.251269e-09	3.373536e-09	9.572175e-08	2.749264e-06
15 Deg (kikuchipy)	9.766781e-10	5.534689e-09	8.984473e-08	2.749264e-06
15 Deg (EBSDTorch)	6.069131e-10	4.373372e-09	1.138714e-07	2.749264e-06
10 Deg (EBSDTorch)	5.397918e-09	2.103315e-08	1.128372e-07	2.749264e-06
8 Deg (EBSDTorch)	1.046371e-07	2.744118e-07	1.477597e-07	2.749264e-06

APPENDIX C

Laue Group Generator Adjacency Graphs

Omitting the known cyclic groups and the fully connected graphs of D2 and D3, tables (C.1-C.4) and figures (C.1-C.4) show the generator adjacency for D4, D6, T, and O.

Gen ID	1	2	3	4	5	6	7	8
1	-	0	1	1	1	1	1	1
2	0	-	1	1	1	1	1	1
3	1	1	-	0	1	1	1	1
4	1	1	0	-	1	1	1	1
5	1	1	1	1	-	0	1	1
6	1	1	1	1	0	-	1	1
7	1	1	1	1	1	1	-	0
8	1	1	1	1	1	1	0	-

Table C.1: Adjacency matrix for Laue group D4.

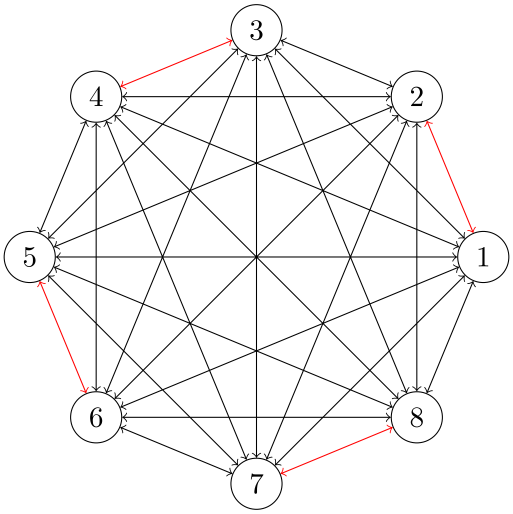

Figure C.1: Adjacency graph for Laue group D4 (red edges indicate non-adjacent generators).

Gen ID	1	2	3	4	5	6	7	8	9	10	11	12
1	-	0	0	0	1	1	1	1	1	1	1	1
2	0	-	0	1	1	0	1	1	1	1	1	1
3	0	0	-	1	0	1	1	1	1	1	1	1
4	0	1	1	-	0	0	1	1	1	1	1	1
5	1	1	0	0	-	0	1	1	1	1	1	1
6	1	0	1	0	0	-	1	1	1	1	1	1
7	1	1	1	1	1	1	-	0	0	1	1	0
8	1	1	1	1	1	1	0	-	0	0	1	1
9	1	1	1	1	1	1	0	0	-	1	0	1
10	1	1	1	1	1	1	1	0	1	-	0	0
11	1	1	1	1	1	1	1	1	0	0	-	0
12	1	1	1	1	1	1	0	1	1	0	0	-

Table C.2: Adjacency matrix for Laue group D6.

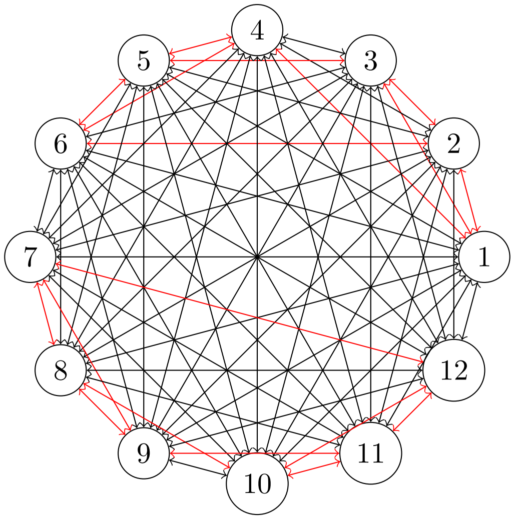

Figure C.2: Adjacency graph for Laue group D6 (red edges indicate non-adjacent generators).

Gen ID	1	2	3	4	5	6	7	8	9	10	11	12
1	-	0	0	0	1	1	1	1	1	1	1	1
2	0	-	0	0	1	1	1	1	1	1	1	1
3	0	0	-	0	1	1	1	1	1	1	1	1
4	0	0	0	-	1	1	1	1	1	1	1	1
5	1	1	1	1	-	1	1	1	1	1	1	1
6	1	1	1	1	1	-	0	1	0	1	1	1
7	1	1	1	1	1	0	-	1	0	1	1	0
8	1	1	1	1	1	1	1	-	1	1	1	1
9	1	1	1	1	1	0	0	1	-	1	1	0
10	1	1	1	1	1	1	1	1	1	-	1	1
11	1	1	1	1	1	1	1	1	1	1	-	1
12	1	1	1	1	1	1	0	1	0	1	1	-

Table C.3: Adjacency matrix for Laue group T.

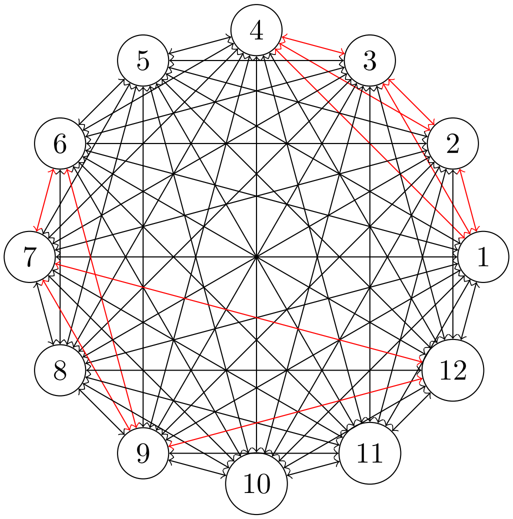

Figure C.3: Adjacency graph for Laue group T (red edges indicate non-adjacent generators).

Gen ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	-	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	0	0
2	0	-	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	-	0	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
4	0	0	0	-	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1
5	1	1	0	0	-	0	1	1	1	1	0	0	1	0	1	0	1	0	1	0	1	1	1	1
6	1	1	0	0	0	-	1	1	1	1	0	0	0	1	0	1	0	1	0	1	1	1	1	1
7	1	0	1	0	1	1	-	0	1	1	1	1	1	0	1	0	0	1	0	1	1	1	0	0
8	1	0	1	0	1	1	0	-	1	1	1	1	0	1	0	1	1	0	1	0	1	1	0	0
9	1	0	0	1	1	1	1	1	-	0	1	1	1	0	0	1	1	0	0	1	0	0	1	1
10	1	0	0	1	1	1	1	1	0	-	1	1	0	1	1	0	0	1	1	0	0	0	1	1
11	0	0	1	1	0	0	1	1	1	1	-	0	1	1	0	0	0	0	1	1	1	1	1	1
12	0	0	1	1	0	0	1	1	1	1	0	-	0	0	1	1	1	1	0	0	1	1	1	1
13	1	1	1	1	1	0	1	0	1	0	1	0	-	1	1	0	1	0	0	1	1	0	1	0
14	1	1	1	1	0	1	0	1	0	1	1	0	1	-	0	1	0	1	1	0	1	0	1	0
15	1	1	1	1	1	0	1	0	0	1	0	1	1	0	-	1	0	1	1	0	1	0	0	1
16	1	1	1	1	0	1	0	1	1	0	0	1	0	1	1	-	1	0	0	1	1	0	0	1
17	1	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	-	1	1	0	0	1	1	0
18	1	1	1	1	0	1	1	0	0	1	0	1	0	1	1	0	1	-	0	1	0	1	1	0
19	1	1	1	1	1	0	0	1	0	1	1	0	0	1	1	0	1	0	-	1	0	1	0	1
20	1	1	1	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	1	-	0	1	0	1
21	0	1	1	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	-	0	1	1
22	0	1	1	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1	1	0	-	1	1
23	0	1	0	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	-	0
24	0	1	0	1	1	1	0	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	-

Table C.4: Adjacency matrix for Laue group O.

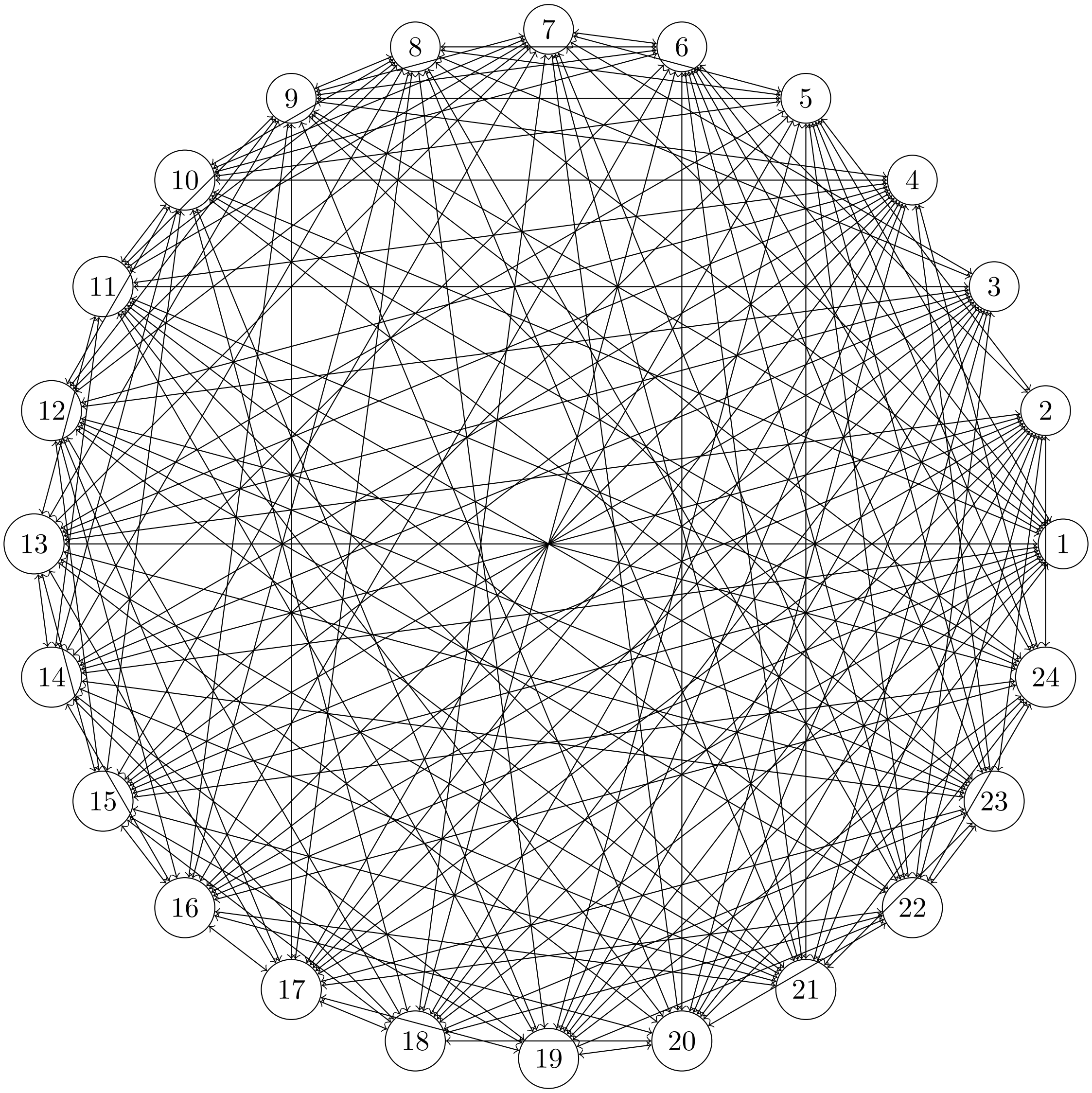

Figure C.4: Adjacency graph for Laue group O.

BIBLIOGRAPHY

[1] H-J Bunge. *Texture analysis in materials science: mathematical methods.* Elsevier, 2013.

[2] D. Roşca, A. Morawiec, and M. De Graef. A new method of constructing a grid in the space of 3D rotations and its applications to texture analysis. *Modelling and Simulation in Materials Science and Engineering*, 22(7):075013, October 2014. Publisher: IOP Publishing.

[3] S. Vespucci, A. Winkelmann, G. Naresh-Kumar, K. P. Mingard, D. Maneuski, P. R. Edwards, A. P. Day, V. O'Shea, and C. Trager-Cowan. Digital direct electron imaging of energy-filtered electron backscatter diffraction patterns. *Physical Review B*, 92(20):205301, November 2015. Publisher: American Physical Society.

[4] Shoji Nishikawa and Seishi Kikuchi. Diffraction of cathode rays by calcite [4], 1928. ISSN: 00280836 Issue: 3080 Pages: 726 Publication Title: Nature Volume: 122.

[5] Z. Ding, E. Pascal, and M. De Graef. Indexing of electron back-scatter diffraction patterns using a convolutional neural network. *Acta Materialia*, 199:370–382, 2020.

[6] Yan Zhang, G. M. Dilshan Godaliyadda, Nicola Ferrier, Emine B. Gulsoy, Charles A. Bouman, and Charudatta Phatak. SLADS-Net: Supervised Learning Approach for Dynamic Sampling using Deep Neural Networks, March 2018. arXiv:1803.02972 [eess].

[7] Olaf Engler and Valerie Randle. *Introduction to Texture Analysis: Macrotexture, Microtexture, and Orientation Mapping, Second Edition.* CRC Press, Boca Raton, 2 edition, November 2009.

[8] Chi-Sing Man. *Crystallographic Texture and Group Representations.* Springer Netherlands, Dordrecht, 2023.

[9] Jian S. Dai. Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 92:144–152, October 2015.

[10] Michael Bloesch, Hannes Sommer, Tristan Laidlow, Michael Burri, Gabriel Nuetzi, Péter Fankhauser, Dario Bellicoso, Christian Gehring, Stefan Leutenegger, Marco Hutter, and Roland Siegwart. A Primer on the Differential Calculus of 3D Orientations, October 2016. arXiv:1606.05285 [cs].

[11] Zachary Teed and Jia Deng. Tangent Space Backpropagation for 3D Transformation Groups, March 2021. arXiv:2103.12032 [cs].

[12] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics, December 2021. arXiv:1812.01537 [cs].

[13] Gregory S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Applied and Numerical Harmonic Analysis. Birkhäuser, Boston, 2012.

[14] Pertti Lounesto. *Clifford Algebras and Spinors*. London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, 2 edition, 2001.

[15] John Voight. The arithmetic of quaternion algebras. *preprint*, 4:16–17, 2014.

[16] William Kahan. IEEE standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE*, 754(94720-1776):11, 1996.

[17] F. C. Frank. Orientation mapping. *Metallurgical Transactions A*, 19(3):403–408, March 1988.

[18] Marc De Graef. Applications of the Clifford torus to material textures. *Journal of Applied Crystallography*, 57(3), June 2024.

[19] R. Arnold, P. E. Jupp, and H. Schaeben. Statistics of ambiguous rotations, January 2017. arXiv:1701.01579 [math, stat].

[20] Ralf Hielscher and Laura Lippert. Locally Isometric Embeddings of Quotients of the Rotation Group Modulo Finite Symmetries, December 2020. arXiv:2007.09664 [math-ph].

[21] Michael O. Buzzy, Andreas E. Robertson, and Surya R. Kalidindi. Statistically conditioned polycrystal generation using denoising diffusion models. *Acta Materialia*, 267:119746, April 2024.

[22] George Marsaglia. Choosing a Point from the Surface of a Sphere. *The Annals of Mathematical Statistics*, 43(2):645–646, April 1972. Publisher: Institute of Mathematical Statistics.

[23] Ken Shoemake. Uniform random rotations. In *Graphics Gems III (IBM Version)*, pages 124–132. Elsevier, 1992.

[24] Marc Alexa. Super-Fibonacci Spirals: Fast, Low-Discrepancy Sampling of SO(3). pages 8291–8300, 2022.

[25] S. I. Wright and M. De Graef. Electron backscatter diffraction. *urn:isbn:*, C:0–0, 2022. Publisher: International Union of Crystallography.

[26] Gert Nolze. Image distortions in SEM and their influences on EBSD measurements. *Ultramicroscopy*, 107(2):172–183, February 2007.

[27] Edward L. Pang, Peter M. Larsen, and Christopher A. Schuh. Global optimization for accurate determination of EBSD pattern centers. *Ultramicroscopy*, 209:112876, February 2020.

[28] Y. G. Li, S. F. Mao, H. M. Li, S. M. Xiao, and Z. J. Ding. Monte Carlo simulation study of scanning electron microscopy images of rough surfaces. *Journal of Applied Physics*, 104(6):064901, September 2008.

[29] Erik Kieft and Eric Bosch. Refinement of Monte Carlo simulations of electron–specimen interaction in low-voltage SEM. *Journal of Physics D: Applied Physics*, 41(21):215310, October 2008.

[30] S. Babin, S. Borisov, A. Ivanchikov, and I. Ruzavin. CHARIOT: Software tool for modeling SEM signal and e-beam lithography. *Physics Procedia*, 1(1):305–313, August 2008.

[31] J. Löffler, J. Thomet, M. Belhaj, L. van Kessel, C. W. Hagen, C. Ballif, and N. Wyrsch. Monte Carlo Modeling of Electron Multiplication in Amorphous Silicon Based Microchannel Plates. In *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–6, October 2019. ISSN: 2577-0829.

[32] David C Joy. *Monte Carlo modeling for electron microscopy and microanalysis*, volume 9. Oxford University Press, 1995.

[33] Stuart I. Wright and Brent L. Adams. Automatic analysis of electron backscatter diffraction patterns. *Metallurgical Transactions A*, 23(3):759–767, March 1992. Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 3 Publisher: Springer-Verlag.

[34] J. Brian Burns, Allen R. Hanson, and Edward M. Riseman. Extracting Straight Lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425–455, 1986.

[35] N. C. Krieger Lassen, D. Juul Jensen, and K. Conradsen. On the statistical analysis of orientation data. *Acta Crystallographica Section A*, 50(6):741–748, 1994.

[36] Robert A. Schwarzer. Automated crystal lattice orientation mapping using a computer-controlled sem, 1997. ISSN: 09684328 Issue: 3 Pages: 249–265 Publication Title: Micron Volume: 28.

[37] Michael van Ginkel, Cris Luengo Hendriks, and Lucas Van Vliet. *A short introduction to the Radon and Hough transforms and how they relate to each other*. March 2004.

[38] W. C. Lenthe, S. Singh, and M. De Graef. A spherical harmonic transform approach to the indexing of electron back-scattered diffraction patterns. *Ultramicroscopy*, 207:112841, December 2019.

[39] Saransh Singh and Marc De Graef. Dictionary indexing of electron channeling patterns. *Microscopy and Microanalysis*, 23(1):1–10, 2017.

[40] Zihao Ding, Chaoyi Zhu, and Marc De Graef. Determining crystallographic orientation via hybrid convolutional neural network. *Materials Characterization*, 178:111213, August 2021.

[41] W. C. Lenthe, L. Germain, M. R. Chini, N. Gey, and M. De Graef. Spherical indexing of overlap EBSD patterns for orientation-related phases – Application to titanium. *Acta Materialia*, 188:579–590, April 2020.

[42] Ralf Hielscher, Felix Bartel, and Thomas Benjamin Britton. Gazing at crystal balls: Electron backscatter diffraction pattern analysis and cross correlation on the sphere. *Ultramicroscopy*, 207:112836, December 2019.

[43] Lorenzo Sorgi and Kostas Daniilidis. Template gradient matching in spherical images. In *Image Processing: Algorithms and Systems III*, volume 5298, pages 88–98. SPIE, May 2004.

[44] Boris Gutman, Yalin Wang, Tony Chan, Paul M. Thompson, and Arthur W. Toga. Shape Registration with Spherical Cross Correlation. page 56, October 2008.

[45] Toshio Fukushima. Numerical computation of spherical harmonics of arbitrary degree and order by extending exponent of floating point numbers. *Journal of Geodesy*, 86(4):271–285, April 2012.

[46] Toshio Fukushima. *Numerical computation of Wigner's d-function of arbitrary high degree and orders by extending exponent of floating point numbers.* November 2016.

[47] Nail A. Gumerov and Ramani Duraiswami. Recursive computation of spherical harmonic rotation coefficients of large degree, March 2014. arXiv:1403.7698 [cs].

[48] Peter Colwell. Solving Kepler's equation over three centuries. *Richmond*, 1993.

[49] Oliver H. E. Philcox, Jeremy Goodman, and Zachary Slepian. Kepler's Goat Herd: An Exact Solution to Kepler's Equation for Elliptical Orbits. *Monthly Notices of the Royal Astronomical Society*, 506(4):6111–6116, August 2021. arXiv:2103.15829 [astro-ph, physics:physics].

[50] Adam J. Schwartz, Mukul Kumar, Brent L. Adams, and David P. Field, editors. *Electron Backscatter Diffraction in Materials Science.* Springer US, Boston, MA, 2009.

[51] Angus J. Wilkinson and T. Ben. Britton. Strains, planes, and EBSD in materials science. *Materials Today*, 15(9):366–376, September 2012.

[52] Patrick T. Brewick, Stuart I. Wright, and David J. Rowenhorst. NLPAR: Non-local smoothing for enhanced EBSD pattern indexing. *Ultramicroscopy*, 200:50–61, May 2019.

[53] Vivian S. Tong, Alexander J. Knowles, David Dye, and T. Ben Britton. Rapid electron backscatter diffraction mapping: Painting by numbers. *Materials Characterization*, 147:271–279, January 2019.

[54] G. M. Dilshan Godaliyadda, Dong Hye Ye, Michael D. Uchic, Michael A. Groeber, Gregery T. Buzzard, Charles A. Bouman, Dong Hye Ye, Michael D. Uchic, Michael A. Groeber, Gregery T. Buzzard, and Charles A. Bouman. A Supervised Learning Approach for Dynamic Sampling. *Electronic Imaging*, 28:1–8, February 2016. Publisher: Society for Imaging Science and Technology.

[55] Stephen Joe and Frances Y. Kuo. Constructing Sobol Sequences with Better Two-Dimensional Projections. *SIAM Journal on Scientific Computing*, 30(5):2635–2654, January 2008. Publisher: Society for Industrial and Applied Mathematics.

[56] I. M Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, January 1967.

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. arXiv:1912.01703 [cs, stat].

[58] Muhammad Asad, Reuben Dorent, and Tom Vercauteren. FastGeodis: Fast Generalised Geodesic Distance Transform. *Journal of Open Source Software*, 7(79):4532, November 2022. arXiv:2208.00001 [cs, eess].

[59] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunès, François-David Collin, and Ghislain Durif. Kernel Operations on the GPU, with Autodiff, without Memory Overflows, April 2021. arXiv:2004.11127 [cs].

[60] Duncan N. Johnstone, Ben H. Martineau, Phillip Crout, Paul A. Midgley, and Alexander S. Eggeman. Density-based clustering of crystal orientations and misorientations and the orix python library. January 2020. arXiv:2001.02716 [cond-mat].

[61] Michael G. Chapman, Michael D. Uchic, J. Michael Scott, Megna N. Shah, Sean P. Donegan, Paul A. Shade, William D. Musinski, Mark Obstalecki, Michael A. Groeber, David Menasche, Marie E. Cox, and Edwin J. Schwalbach. 3D Reconstruction of an Additive Manufactured IN625 Tensile Sample Using Serial Sectioning and Multi-Modal Characterization. *Microscopy and Microanalysis*, 25(S2):342–343, August 2019. Publisher: Cambridge University Press.

[62] Carter K. Cocke, Anthony D. Rollett, Ricardo A. Lebensohn, and Ashley D. Spear. The AFRL Additive Manufacturing Modeling Challenge: Predicting Micromechanical Fields in AM IN625 Using an FFT-Based Method with Direct Input from a 3D Microstructural Image. *Integrating Materials and Manufacturing Innovation*, 10(2):157–176, June 2021.

[63] David B. Menasche, William D. Musinski, Mark Obstalecki, Megna N. Shah, Sean P. Donegan, Joel V. Bernier, Peter Kenesei, Jun-Sang Park, and Paul A. Shade. AFRL Additive Manufacturing Modeling Series: Challenge 4, In Situ Mechanical Test of an IN625 Sample with Concurrent High-Energy Diffraction Microscopy Characterization. *Integrating Materials and Manufacturing Innovation*, 10(3):338–347, September 2021.

[64] Michael G. Chapman, Megna N. Shah, Sean P. Donegan, J. Michael Scott, Paul A. Shade, David Menasche, and Michael D. Uchic. AFRL Additive Manufacturing Modeling Series: Challenge 4, 3D Reconstruction of an IN625 High-Energy Diffraction Microscopy Sample Using Multi-modal Serial Sectioning. *Integrating Materials and Manufacturing Innovation*, 10(2):129–141, June 2021.

[65] Brian L. DeCost, Toby Francis, and Elizabeth A. Holm. Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures, February 2017. arXiv:1702.01117 [cond-mat].

[66] Frank Roels. CIL:48302, Sprague-Dawley rat. CIL. Dataset.

[67] Damián H. Zanette. Quantifying the complexity of black-and-white images. *PLoS ONE*, 13(11):e0207879, November 2018.

[68] E. Hellinger. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, July 1909.

[69] P. G. Callahan, J. P. Simmons, and M. De Graef. A quantitative description of the morphological aspects of materials structures suitable for quantitative comparisons of 3D microstructures. *Modelling and Simulation in Materials Science and Engineering*, 21(1):015003, November 2012. Publisher: IOP Publishing.

[70] P. G. Callahan, M. Groeber, and M. De Graef. Towards a quantitative comparison between experimental and synthetic grain structures. *Acta Materialia*, 111:242–252, June 2016.

[71] Michael A. Groeber and Michael A. Jackson. DREAM.3D: A Digital Representation Environment for the Analysis of Microstructure in 3D. *Integrating Materials and Manufacturing Innovation*, 3(1):56–72, December 2014.

[72] N. C. Krieger Lassen, D. Juul Jensen, and K. Conradsen. Image Processing Procedures for Analysis of Electron Back Scattering Patterns. *Scanning Microscopy*, 6(1), March 1992.

[73] N. Lassen, K. Conradsen, and D. Jensen. Automated Determination of Crystal Orientations from Electron Backscattering Patterns. September 1994.

[74] Brent L. Adams, Stuart I. Wright, and Karsten Kunze. Orientation imaging: The emergence of a new microscopy. *Metallurgical Transactions A*, 24(4):819–831, April 1993.

[75] Aimo Winkelmann, Carol Trager-Cowan, Francis Sweeney, Austin P. Day, and Peter Parbrook. Many-beam dynamical simulation of electron backscatter diffraction patterns. *Ultramicroscopy*, 107(4):414–421, April 2007.

[76] Patrick G Callahan and Marc De Graef. Dynamical Electron Backscatter Diffraction Patterns. Part I: Pattern Simulations. *Microscopy and Microanalysis*, 19(5):1255–1265, October 2013.

[77] Yu-Hui Chen, Se Un Park, Dennis Wei, Gregory Newstadt, Michael Jackson, Jeff P. Simmons, Marc De Graef, and Alfred O. Hero. A Dictionary Approach to EBSD Indexing, February 2015. arXiv:1502.07436 [physics, stat].

[78] M. De Graef. A dictionary indexing approach for EBSD. *IOP Conference Series: Materials Science and Engineering*, 891(1):012009, July 2020. Publisher: IOP Publishing.

[79] M. A. Jackson, E. Pascal, and M. De Graef. Dictionary Indexing of Electron Back-Scatter Diffraction Patterns: a Hands-On Tutorial. *Integrating Materials and Manufacturing Innovation*, 8(2):226–246, June 2019.

[80] S. Singh and M. De Graef. Orientation sampling for dictionary-based diffraction pattern indexing methods. *Modelling and Simulation in Materials Science and Engineering*, 24(8):085013, November 2016. Publisher: IOP Publishing.

[81] Jake Lever, Martin Krzywinski, and Naomi Altman. Principal component analysis. *Nature Methods*, 14(7):641–642, July 2017. Number: 7 Publisher: Nature Publishing Group.

[82] Drew Schmidt. A Survey of Singular Value Decomposition Methods for Distributed Tall/Skinny Data. In *2020 IEEE/ACM 11th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA)*, pages 27–34, GA, USA, November 2020. IEEE.

[83] Gilbert Strang. *Introduction to linear algebra*. Cambridge press, Wellesley, 5th edition edition, 2016.

[84] B. P. Welford. Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics*, 4(3):419–420, August 1962. Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/00401706.1962.10490022.

[85] Andrey A. Efanov, Sergey A. Ivliev, and Alexey G. Shagraev. Welford's algorithm for weighted statistics. In *2021 3rd International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*, pages 1–5, March 2021.

[86] Tony F. Chan, Gene H. Golub, and Randall J. LeVeque. Algorithms for Computing the Sample Variance: Analysis and Recommendations. *The American Statistician*, 37(3):242–247, 1983. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].

[87] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, December 2010. arXiv:0909.4061 [math].

[88] Ziming Liu, Sitian Qian, Yixuan Wang, Yuxuan Yan, and Tianyi Yang. Schr\"{o}dinger PCA: On the Duality between Principal Component Analysis and Schr\"{o}dinger Equation. *Physical Review E*, 104(2):025307, August 2021. arXiv:2006.04379 [cond-mat, physics:physics].

[89] Yao Tian, Ziyang Yue, Ruiyuan Zhang, Xi Zhao, Bolong Zheng, and Xiaofang Zhou. Approximate Nearest Neighbor Search in High Dimensional Vector Databases: Current Research and Future Directions.

[90] Jianyang Gao and Cheng Long. High-Dimensional Approximate Nearest Neighbor Search: with Reliable and Efficient Distance Comparison Operations. *Proceedings of the ACM on Management of Data*, 1(2):1–27, June 2023.

[91] Mario Köppen. The Curse of Dimensionality. April 2009.

[92] Richard Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[93] Olivia Weng. Neural Network Quantization for Efficient Inference: A Survey, January 2023. arXiv:2112.06126 [cs].

[94] Daya Khudia, Jianyu Huang, Protonu Basu, Summer Deng, Haixin Liu, Jongsoo Park, and Mikhail Smelyanskiy. FBGEMM: Enabling High-Performance Low-Precision Deep Learning Inference, January 2021. arXiv:2101.05615 [cs].

[95] Marat Dukhan, Yiming Wu, and Hao Lu. QNNPACK: Open source library for optimized mobile deep learning, October 2018.

[96] Darshan C. Ganji, Saad Ashfaq, Ehsan Saboori, Sudhakar Sah, Saptarshi Mitra, MohammadHossein AskariHemmat, Alexander Hoffman, Ahmed Hassanien, and Mathieu Léonardon. DeepGEMM: Accelerated Ultra Low-Precision Inference on CPU Architectures using Lookup Tables, April 2023. arXiv:2304.09049 [cs].

[97] Håkon Wiik Ånes, Jarle Hjelen, Antonius T. J. van Helvoort, and Knut Marthinsen. Electron backscatter patterns from Nickel acquired with varying camera gain, September 2019.

[98] H. W. Ånes, J. Hjelen, B. E. Sørensen, A. T. J. van Helvoort, and K. Marthinsen. Processing and indexing of electron backscatter patterns using open-source software. *IOP Conference Series: Materials Science and Engineering*, 891(1):012002, July 2020. Publisher: IOP Publishing.

[99] David J. Rowenhorst, Patrick G. Callahan, and Håkon W. Ånes. Fast Radon transforms for high-precision EBSD orientation determination using PyEBSDIndex. *Journal of Applied Crystallography*, 57(1):3–19, 2024. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1107/S1600576723010221.

[100] Zhang Li, Lucas J. van Vliet, and Frans M. Vos. Self Similarity Image Registration Based on Reorientation of the Hessian. In Hiroyuki Yoshida, Simon Warfield, and Michael W. Vannier, editors, *Abdominal Imaging. Computation and Clinical Applications*, Lecture Notes in Computer Science, pages 20–28, Berlin, Heidelberg, 2013. Springer.

[101] Debapriya Sengupta, Phalguni Gupta, and Arindam Biswas. A survey on mutual information based medical image registration algorithms. *Neurocomputing*, 486:174–188, May 2022.

[102] Jonas August and Takeo Kanade. The Role of Non-Overlap in Image Registration. In Gary E. Christensen and Milan Sonka, editors, *Information Processing in Medical Imaging*, Lecture Notes in Computer Science, pages 713–724, Berlin, Heidelberg, 2005. Springer.

[103] Yuanxin Ye, Lorenzo Bruzzone, Jie Shan, Francesca Bovolo, and Qing Zhu. Fast and Robust Matching for Multimodal Remote Sensing Image Registration. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):9059–9070, 2019. arXiv:1808.06194 [cs].

[104] F. Y. Edgeworth. The Generalised Law of Error, or Law of Great Numbers. *Journal of the Royal Statistical Society*, 69(3):497, September 1906.

[105] Mathieu Rubeaux, Jean-Claude Nunes, Laurent Albera, and Mireille Garreau. Edgeworth-based approximation of Mutual Information for medical image registration. In *2010 2nd International Conference on Image Processing Theory, Tools and Applications*, pages 195–200, July 2010. ISSN: 2154-512X.

[106] Mathieu Rubeaux. *Approximation de l'Information Mutuelle basée sur le développement d'Edgeworth : application au recalage d'images médicales.* phdthesis, Université Rennes 1, July 2011.

[107] M. Rubeaux, J. C. Nunes, L. Albera, and M. Garreau. Medical image registration using Edgeworth-based approximation of Mutual Information. *IRBM*, 35(3):139–148, June 2014.

[108] Alessia Amelio and Clara Pizzuti. Correction for Closeness: Adjusting Normalized Mutual Information Measure for Clustering Comparison: Correction For Closeness: Adjusting NMI. *Computational Intelligence*, 33, September 2016.

[109] M. B. Cook. Bi-Variate k-Statistics and Cumulants of Their Joint Sampling Distribution. *Biometrika*, 38(1/2):179–195, 1951. Publisher: [Oxford University Press, Biometrika Trust].

[110] Yu-Feng Shen, S. Maddali, D. Menasche, A. Bhattacharya, G. S. Rohrer, and R. M. Suter. Importance of outliers: A three-dimensional study of coarsening in -phase iron. *Physical Review Materials*, 3(6):063611, June 2019.

[111] D. P. Field, P. B. Trivedi, S. I. Wright, and M. Kumar. Analysis of local orientation gradients in deformed single crystals. *Ultramicroscopy*, 103(1):33–39, April 2005.

[112] Luke N. Brewer, David P. Field, and Colin C. Merriman. Mapping and Assessing Plastic Deformation Using EBSD. In Adam J. Schwartz, Mukul Kumar, Brent L. Adams, and David P. Field, editors, *Electron Backscatter Diffraction in Materials Science*, pages 251–262. Springer US, Boston, MA, 2009.

[113] VP Yashnikov, HJ Bunge, and others. Group-theoretical approach to reduced orientation spaces for crystallographic textures. *Texture, Stress, and Microstructure*, 23:201–219, 1995. Publisher: Hindawi.

[114] VP Yashnikov, HJ Bunge, and others. Geometrical Foundations of Texture Analysis. Geodesic Curves and Motions in the group Space of Three-Dimensional Rotations. *Texture, Stress, and Microstructure*, 30:1–42, 1997. Publisher: Hindawi.

[115] V. P. Yashnikov. Group-theoretical principles of texture analysis of polycrystalline materials. *Bulletin of the Russian Academy of Sciences: Physics*, 71(12):1737–1747, December 2007.

[116] Robert Krakow, Robbie J. Bennett, Duncan N. Johnstone, Zoja Vukmanovic, Wilberth Solano-Alvarez, Steven J. Lainé, Joshua F. Einsle, Paul A. Midgley, Catherine M. F. Rae, and Ralf Hielscher. On three-dimensional misorientation spaces. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2206):20170274, October 2017. Publisher: Royal Society.

[117] Poorya Mianjy and Raman Arora. Stochastic PCA with ℓ_2 and ℓ_1 Regularization. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3531–3539. PMLR, July 2018. ISSN: 2640-3498.

[118] Roy Mitz and Yoel Shkolnisky. ROIPCA: An online memory-restricted PCA algorithm based on rank-one updates, June 2023. arXiv:1911.11049 [cs, stat].

[119] Alexander B Kyatkin and Gregory S Chirikjian. Pattern Matching as a Correlation on the Discrete Motion Group. *Computer Vision and Image Understanding*, 74(1):22–35, April 1999.

[120] Xinhua Zhang and Lance R. Williams. Similarity Equivariant Linear Transformation of Joint Orientation-Scale Space Representations, March 2022. arXiv:2203.06786 [cs].

[121] Erik M Franken. Enhancement of crossing elongated structures in images. 2008.

[122] Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

[123] R. Arnold and P. E. Jupp. Statistics of orthogonal axial frames. *Biometrika*, 100(3):571–586, September 2013.

[124] Morris W. Hirsch. The Imbedding of Bounding Manifolds in Euclidean Space. *The Annals of Mathematics*, 74(3):494, November 1961.

[125] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. pages 5745–5753, 2019.

[126] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas Guibas. Vector Neurons: A General Framework for SO(3)-Equivariant Networks, April 2021. arXiv:2104.12229 [cs].

[127] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) Equivariant Representations with Spherical CNNs, September 2018. arXiv:1711.06721 [cs].