



AFRL-AFOSR-VA-TR-2021-0062

LEGOMAT: Locally Extracted Globally Organized Microstructure Models using Markov Random Fields

Sundararaghavan, Veera
REGENTS OF THE UNIVERSITY OF MICHIGAN
503 THOMPSON ST
ANN ARBOR, MI, 48109
USA

06/27/2021
Final Technical Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
Air Force Office of Scientific Research
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 27-06-2021		2. REPORT TYPE Final		3. DATES COVERED (From - To) 01 Jan 1918 - 31 Dec 2020	
4. TITLE AND SUBTITLE LEGOMAT: Locally Extracted Globally Organized Microstructure Models using Markov Random Fields				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER FA9550-18-1-0091	
				5c. PROGRAM ELEMENT NUMBER 61102F	
6. AUTHOR(S) Veera Sundararaghavan				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) REGENTS OF THE UNIVERSITY OF MICHIGAN 503 THOMPSON ST ANN ARBOR, MI 48109 USA				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AF Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/AFOSR RTB1	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-AFOSR-VA-TR-2021-0062	
12. DISTRIBUTION/AVAILABILITY STATEMENT A Distribution Unlimited: PB Public Release					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The problem of microstructure reconstruction is deeply intertwined with the theories of microstructure quantification. Currently available methods for microstructure synthesis such as geometry based (Voronoi models), physically based (Phase field models) or feature-based (Simulated annealing) methods run into various difficulties when modeling real complexities of microstructures that include non-equilibrium features, non-convex grains, twins, second phases and cell structures that arise from thermomechanical processing. These features play an important role in the properties and performance of modern aerospace alloys. Further, microstructures are stochastic and lead to location-specific variability in material properties. In this proposal, we delve into a mathematical model that is expected to provide a better alternative for microstructure synthesis: Markov random fields. In the report, theory and software for building 3D microstructural maps of engineering components through inference from 2D measurements is proposed					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			ALI SAYIR
U	U	U	UU	224	19b. TELEPHONE NUMBER (Include area code) 426-7236

LEGOMAT: Locally Extracted Globally Organized Microstructural Material Models

by

Veera Sundararaghavan

To fulfill the requirements of
Air Force Office of Scientific Research Materials for Extreme Environments Program
(Grant No. FA9550-18-1-0091) Final report
(Aerospace Engineering)
in the University of Michigan
2021

Student Contributors:

Sid Srivastava, Ph.D.(2020)
Iman Javaheri, Ph.D. (ongoing)

Prof. Veera Sundararaghavan
veeras@umich.edu

“A good idea has a way of becoming simpler and solving problems other than that for which it was intended”– Robert Tarjan

ACKNOWLEDGEMENTS

I acknowledge the Air Force Office of Scientific Research Materials for Extreme Environments Program (Grant No. FA9550-18-1-0091) for financial support. I would like to express my sincere gratitude towards my program manager, Dr. Ali Sayir, for support and encouragement to pursue challenging ideas. Dr. Sayir manages projects in several exciting areas of research, and understands the importance of microstructure quantification and analysis as a unifying force in all these areas. He thus maintains a portfolio of cutting-edge research at the intersection of microstructure, computation and mathematics, of which this project is a part. Dr. Sayir's suggestions and questions in review meetings have not only refined this work but have also given me new ideas for my future research in this area.

The work acknowledging this project is published in these papers:

1. I. Javaheri, V Sundararaghavan, Polycrystalline Microstructure Reconstruction Using Markov Random Fields and Histogram Matching, Computer Aided Design, Volume 120, 102806 (2020)
2. S. Srivastava, M. Yaghoobi, V. Sundararaghavan, A graph-theoretic approach for multiscale modeling and prediction of crack propagation in polycrystalline materials, Engineering Fracture Mechanics, Volume 241, January 2021, 107406
3. S. Srivastava, V. Sundararaghavan. Graph Coloring Approach to Mesh Generation in Multiphase Media with Smooth Boundaries, AIAA Journal, Vol. 58, No. 1, 2020.

4. S. Ganesan, I. Javaheri, V. Sundararaghavan. Constrained Voronoi Models For Interpreting Surface Microstructural Measurements, Mechanics of Materials, in review, 2021.
5. S. Sundar, V. Sundararaghavan, Database development and exploration of process-microstructure relationships using variational autoencoders, Materials Today Communications, Volume 25, December 2020, 101201
6. S. Srivastava, V. Sundararaghavan. Box algorithm for the solution of differential equations on a quantum annealer, Physical Review A, 99, 052355(1-10) (2019)
7. I. Javaheri, V. Sundararaghavan., Large-Scale Synthesis of Microstructures Manufactured by Laser Powder-Bed Fusion Using Markov Random Fields, Computer-Aided Design, in review, 2021.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	viii
LIST OF TABLES	xv
LIST OF APPENDICES	xvi
ABSTRACT	xvii
CHAPTER	
I. Markov random Fields for Microstructure Reconstruction	1
1.1 Introduction	2
1.2 Methodology	5
1.2.1 Generation of 3D Unit Cells - MRF Sampling Algorithm	5
1.2.2 Large-Scale Synthesis - LEGOMAT Embedding Algorithm	8
1.3 Results and Discussion	11
1.4 Conclusions	16
II. LEGOMAT: Integration of MRF models for large scale microstructure synthesis	25
2.1 Introduction	25
2.2 Methodology	29
2.2.1 Sampling Algorithm	29
2.2.2 Histogram Matching Algorithm	33
2.2.3 Computational Cost vs. Window Size	36
2.3 Results and Discussion	39
2.3.1 Example 1: Validation of 3D Grain Size	39

2.3.2	Example 2: 3D Reconstruction Based on EBSD Data	43
2.3.3	Example 3: Reconstruction of Polarized Light Micrograph	48
2.3.4	Example 4: Limitation of Markov Random Field Modeling	50
2.4	Conclusions	51
III. Graph Coloring Method for Conformal Mesh Generation . .		53
3.1	Background	54
3.2	Problem Formulation	56
3.3	Results and Discussion	60
3.4	Conclusion	66
IV. Graph Coloring Method for Modeling Microstructural Fracture		68
4.1	Background	69
4.2	Mathematical Framework	73
4.2.1	Graph representation of Microscopic crack	75
4.2.2	Graph based modeling of Energy release	77
4.2.3	Modeling surface energy	78
4.3	Method	83
4.3.1	Problem Setup	83
4.3.2	Computational Procedure	84
4.3.3	Computational complexity	86
4.4	Application to examples	87
4.4.1	Fracture in a homogeneous medium	88
4.4.2	Cracks in Polycrystalline material	94
4.5	Conclusion	100
V. Graph Coloring Method for Modeling Evolving Microstructures		102
5.1	Problem formulation	104
5.1.1	Temporal evolution: First order dynamics	106
5.2	Estimation of Length	109
5.2.1	Embedding graphs in Regular 2D grids	110
5.2.2	Numerical approximation to Cauchy-Crofton relation	113
5.3	Computation with Quadtree structure	116
5.3.1	Pairwise formulation for pixel mesh	117
5.3.2	Pairwise formulation for Quadtree mesh	118
5.4	Simulation Results	119
5.5	Extensions	121
5.5.1	Modelling anisotropic surface energies for non-equiaxed microstructure	121

5.5.2	Modelling 3D grains for denoising experimental Microstructure image	123
5.6	Conclusion	126
VI.	Constrained Voronoi Models For Interpreting Surface Microstructural Measurements	128
6.1	Introduction	129
6.2	Methodology	132
6.2.1	2D Voronoi Diagram	132
6.2.2	Convex 2D Microstructure	135
6.2.3	Obtaining Voronoi Generators for 2D Convex Microstructures	136
6.2.4	Construction of 3D Microstructure Using Voronoi Generators	139
6.2.5	Convexification of 3D Microstructure	140
6.3	Results and Discussion	143
6.4	Conclusions	154
VII.	Conclusions	155
7.1	Summary	155
7.2	Applications of proposed methodologies	157
APPENDICES		161
B.1	Conformal mesh generation	173
B.2	Modeling microstructural fracture	178
B.3	Multiphase flow	182
BIBLIOGRAPHY		187

LIST OF FIGURES

Figure

- | | | |
|-----|---|----|
| 1.1 | <i>LEGOMAT approach: full-scale component imaging built by using a single site-specific MRF unit cell. A small-scale texture of the AM specimen is extracted through EBSD imaging of three orthogonal planes. The 3D microstructure is then reconstructed via the MRF technique. The laser path, along with other processing parameters such as layer thickness and hatch spacing (imposed by the user) are then used to embed the crystallographic elements from MRF unit cell onto the CAD model. The BD, SD, and TD axes here represent the Building, Scanning, and Tangential Directions, respectively.</i> | 18 |
| 1.2 | <i>MRF unit cell reconstruction schematic: (a) the neighborhoods of the voxel v in the slices orthogonal to the $i = \{x, y, z\}$ axes, denoted as $\mathbf{V}_v^i$, are shown. (b) The windows in the input 2D micrographs shown in dotted lines are denoted by $\mathbf{S}_v^i$. These windows closely resemble the neighborhoods of the orthogonal cross sections of voxel v, $\mathbf{V}_v^i$. . .</i> | 19 |
| 1.3 | <i>Histogram matching algorithm: given the reference (2D) and the target (3D) images, the color histograms are obtained by binning the colors into 255 discrete intervals. Cumulative Distribution Functions (CDFs) of the two color histograms, denoted as $H_{2D}(\cdot)$ and $H_{3D}(\cdot)$ are then computed and normalized. The color level G_p in the synthesized 3D image is switched for each RGB channel independently. The new color G_q has the same cumulative distribution function as in input 2D exemplars.</i> | 19 |
| 1.4 | <i>Illustration of LEGOMAT embedding process: (a) Inputting the geometrical mesh along with 3D solid microstructure, (b) manually drawing the local vector fields, representing the preferred crystal growth orientation, (c) setting up the grain scaling by embedding larger grains farther from the laser path, (d) result of LEGOMAT's automatic embedding procedure, along with snapshots of three orthogonal cross-sections in the realized structure.</i> | 20 |

1.5	<i>LEGOMAT optimization schematic: the optimization process minimizes the difference vectors $\mathbf{d}_p$, $\mathbf{d}_q$, and $\mathbf{d}_r$ between microstructure coordinate axes $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and the transformed tensor axes $(\Phi(\mathbf{P}), \Phi(\mathbf{Q}), \Phi(\mathbf{R}))$ for each tetrahedron.</i>	21
1.6	<i>Illustration of Orientation Distribution Function (ODF): (a) the Rodrigues fundamental region for cubic symmetry, (b) a discretized ODF with independent nodes colored blue, (c) colormap showing the RGB values of independent nodes in the fundamental region.</i>	21
1.7	<i>Illustration of 2D/3D images: (a) set of orthogonal experimental images of AM 316 L stainless steel microstructure, (b) 3D MRF reconstructed model, and (c) sectional images of the reconstructed model.</i>	22
1.8	<i>Texture comparison of $(1, 1, 1)$, $(1, 1, 0)$, and $(1, 0, 0)$ pole figures for: (a) the orthogonal planar images against (b) the synthesized 3D microstructure.</i>	22
1.9	<i>Illustration of 2D/3D spatial distribution of grain boundary misorientation angles for: (a) set of orthogonal experimental images of AM microstructure, (b) 3D MRF reconstructed model, and (c) sectional images of reconstructed model.</i>	23
1.10	<i>Comparison of Probability Distribution Function (PDF) of grain boundary misorientation angles of the 3D synthesized microstructure against experimental imaging. Here, the probability densities of the raw misorientation angles are computed over 18 bins using a log-normal fit. The maximum unique misorientation angle possible due to cubic symmetry is 62.8°.</i>	23
1.11	<i>Illustration of LEGOMAT embedding for simulating microstructure (right) for (a) two laser scan lines across a single layer and (b) two layers with different hatch spacing. The laser paths and tensor fields for each case are shown in the left.</i>	24
2.1	<i>(a) The neighborhoods of the voxel v in the slices orthogonal to the x, y, and z axes are shown. (b) The windows in the input 2D micrograph shown in dotted lines are denoted by $\mathbf{S}_v^i$ for $i = x, y$, and z. These windows closely resemble the neighborhoods of the voxel, v.</i>	31
2.2	<i>Histogram matching algorithm: the color G_i in the synthesized 3D image is switched for each channel independently. The new color G_j has the same cumulative distribution function as in the 2D images.</i>	34
2.3	<i>Effect of histogram matching: without histogram matching (as illustrated on the first row), the color space continuously shrinks and phase information is lost after first few iterations; contrarily, with histogram matching (as shown on the bottom), the color space remains consistent with the input reference image throughout the simulation.</i>	35
2.4	<i>Effect of window size: none of the sections in 3D images look identical to the sample image, but the image generated with window size of 11 has statistical correlation function similar to that of input image.</i>	37

2.5	<i>An experimental 2D polycrystalline microstructure (as shown on the left) is used as an exemplar for 3D reconstruction. The 3D MRF reconstructed model is depicted on the right.</i>	40
2.6	<i>Schematic explaining the Saltykov formula: (a) Grains are assumed as spheres with a random section seen in the 2D image. (b) Probability of a given 3D grain size R given a 2D section radius r can be obtained by a change of variables. (c) In the case where multiple 3D grain sizes exist, the Saltykov estimate takes into account the probability that a plane can cut a sphere of radius R. Hence, the spheres of larger radii are more likely to intersect the section plane.</i>	41
2.7	<i>(a) Comparison of the grain size distribution (PDF) of the 3D microstructure along with the Saltykov prediction. (b) Comparison of cumulative distribution function (CDF) of 3D grain sizes with Saltykov prediction.</i>	42
2.8	<i>Several sectional images of the 3D reconstructed model along diagonal (shown on the left) and orthogonal (shown on the right) directions are illustrated. The grain sizes and shapes along these sections nearly resemble the 2D reference image.</i>	43
2.9	<i>(a) Comparison of probability density distributions of the 3D grain sizes along oblique and orthogonal sections. (b) Comparison of cumulative distribution function of the 3D microstructure across oblique and orthogonal directions.</i>	44
2.10	<i>(a) Rodrigues fundamental region for cubic symmetry. (b) A coarse discretization with independent nodes colored blue. (c) A finer discretization.</i>	44
2.11	<i>The colormap shows the RGB values of individual nodes in the face-centered cubic (FCC) fundamental region. The pixels RGB values in the raw 2D orthogonal EBSD images are mapped to the closest node (in Euclidean norm sense) using the colormap. The orientation distribution functions of all three input images are similar, after adjusting the relative alignments of the detectors with respect to the sample.</i>	45
2.12	<i>(a) Experimental 2D polycrystalline microstructures imaged in orthogonal planes. (b) 3D MRF reconstructed model. (c) Sectional images of the reconstructed microstructure.</i>	46
2.13	<i>Comparison of the ODFs and various pole figures of the 2D sample images against the synthesized 3D microstructure.</i>	47
2.14	<i>Texture sequence analysis is shown for MRF algorithm. Each reconstruction uses the random slices of the previous 3D model (shown on its left) as its input exemplars. Here, the orientation distributions of each reconstruction exceptionally match with the original exemplars' ODF.</i>	48
2.15	<i>(a) Experimental 2D polycrystalline microstructure. (b) 3D MRF reconstructed model. (c) Sectional images of the reconstructed microstructure.</i>	49

2.16	(left) Four predominant colors (purple (P), blue (B), orange (O), yellow (Y)) as marked on the 2D and 3D microstructures. (right) Neighbor statistics of the colors; the x-axis denotes the color of the chosen grain, and the bars indicate the probability of occurrence of colored neighbor grains. There are two bars for each neighbor color, one for the 2D image and another for the 3D reconstructed microstructure.	50
2.17	This example is specifically selected to illustrate the shortcomings of the MRF algorithm, as presented in this chapter. The varying morphology of the given microstructure along the z direction, can not be captured by only one single image orthogonal to that axis. As a result, the MRF algorithm can not be used to properly reconstruct such a 3D model. Here, the slices on the right hand side correspond to $z = 0, 64, \text{ and } 128$, respectively.	51
3.1	Multiscale design of turbine blades for aircraft engines.	55
3.2	Construction of the graph $G=(V,E)$ from a finite element mesh. . .	58
3.3	An illustration of labeling procedure: (a) Microstructure image (b) Pixel-based labeling (c) Smoothened mesh using Multiway K-Cut based labeling.	61
3.4	Grain identification in a mesh using a 50x50 pixel image: (a) Pixel-based labeling (b) Multiway K-Cut based labeling.	61
3.5	Dependence of Graph partitioning on the data term in energy. . . .	62
3.6	Comparative study of Finite element simulation results for a bicrystal in plane strain (linear elastic) using Pixel-based (quad mesh), Graph-based (triangle mesh with smoothing) and Conformal mesh	62
3.7	Grain identification for elongated polycrystals using (a) Euclidean metric (b) using Riemannian metric	63
3.8	Illustration of various morphologies based on the metric tensor, $\mathbf{D}$.	63
3.9	Grain identification in 3D meshes using a 64x64x64 voxel base image: (a) Vertex position based labeling (b) Multiway K-Cut based labeling	64
3.10	Mesh generation for woven fiber composite (a) Voxelated mesh with $10^2 \times 10^2 \times 10^2$ elements (b) Tetrahedral mesh with multiway K-Cut based labeling (c) Comparison of fiber volume and surface area of various voxelated meshes and a tetrahedral (smoothened) mesh .	64
3.11	Time complexity analysis of the mesh generation procedure.	65
4.1	Experimental image of a fatigue crack in WE43 Mg alloy (HCP) adapted from [1]. Transgranular fracture followed the basal trace (red line) in grains A-F. Intergranular fracture occurred between grain pairs (A,B), (G,H) and (G,I).	71
4.2	A portion of the FE mesh (in blue) with the dual graph, $\mathcal{G}$ (in yellow) embedded in the mesh. Only the connections across the crack (in red) contribute to the cost. The energy of the connection between elements A and B is determined by the length and the normal of the interface edge.	76

4.3	An illustration of the 3D surface energy density of a cubic material with $\gamma_0 = 1J/m^2$ and $\delta = 2$ in the direction normal to the cleavage surface.	80
4.4	An illustration for the dihedral angle of a 2D crack.	81
4.5	2D surface energy density of a cubic material with $\gamma_0 = 1J/mm^2$ and $\delta = 2$. Each row represents a specific crystal orientation in the specimen frame of reference, and the polar plots are evaluated with respect to cleavage normals. 2D surface energy densities are evaluated for a specimen with unit thickness. The dihedral angle is plotted as $\psi_D + 90^\circ$ for positivity considerations in the polar plot.	82
4.6	(a) A tensile test specimen for mixed mode failure (b) The definition of the macroscopic crack angle from the microscopic crack path. . .	83
4.7	Flowchart of the procedure for crack estimation.	86
4.8	Run-time Analysis: Computation time is estimated for each process for different mesh sizes. A sample set of 20 meshes is used for each mesh size. Error bars represent the standard deviation in the computation time for each sample set.	88
4.9	Mesh convergence analysis: $d(C_{\text{Numerical}}, C_{\text{Analytical}})$ is estimated for different mesh sizes. A sample set of 20 meshes is used for each mesh size. Error bars represent the standard deviation in $d(C_{\text{Numerical}}, C_{\text{Analytical}})$ for each sample set.	90
4.10	The crack path predicted in mixed-mode failure of an isotropic media. The black line represents the numerically simulated crack in the semi-circular region. The overlaid yellow line represents the corresponding macroscopic crack direction.	90
4.11	The crack path predicted in the mode-I failure of a homogeneous media with a preferred cleavage direction of 45° . (a) Polar plot of the surface energy density for different values of δ . (b) Numerically simulated crack where the black line represents the microscopic crack, and the overlaid yellow line represents the corresponding macroscopic crack direction.	92
4.12	Polar plot for γ_0/γ with respect to the crack propagation angle, $\theta' = \theta - 90^\circ$ for a bi-directional surface energy with $\delta = 1$ and the crystal orientation (a) $\omega = 30^\circ$, (b) $\omega = 10^\circ$. The grey area represent the forbidden regions (c) The lower region of the cracked surface for $\omega = 30^\circ$ (top) and $\omega = 10^\circ$ (bottom)	94
4.13	Effect of grain boundary on crack propagation	96
4.14	Effect of $\bar{r}$ on crack path	98
4.15	Dynamic crack propagation in a polycrystalline material	99
5.1	Illustration of $dist(\Gamma, \Gamma_0)$ function where the initial curve Γ_0 evolves to Γ	108
5.2	(a) Illustration for Cauchy crofton relations: $\text{length}(\Gamma) \propto$ number of intersections with lines, (b) Parameterization of a line, L , in terms of ρ and ϕ , and (c) Set, Ξ , of all lines in $\mathbb{R}^2$ represented in the (ρ, ϕ) -coordinates	110

5.3	Connections in a stencil. The red region denotes the stencil, $\mathcal{N}_1$. The stencil, $\mathcal{N}_{\sqrt{2}}$ is constructed by adding connections in the green region to $\mathcal{N}_1$. The neighborhood, $\mathcal{N}_{\sqrt{5}}$ is constructed by adding the nodes in the blue region to the neighborhood, $\mathcal{N}_{\sqrt{2}}$	113
5.4	(a) Illustration shows the relationship of a pixel in parent layer and its children (b) Comparison of two circular grains represented using Pixel Mesh and Quadtree mesh with similar number of active nodes. The pixelated mesh has a discretization of 38×38 while the Quadtree mesh has 4 layers with the initial layer of size 10×10	117
5.5	Simulation results for a pill shaped grain.	120
5.6	A 4-point junction splitting into 2 3-point junctions. All grains have no bulk energy density and surface energy between each pair of grain is also same.	120
5.7	simulation of a polycrystalline microstructure	121
5.8	Simulation of an initially circular grain with surface energy estimated using Eq (5.18) with $D_{11} = D_{22} = 1, D_{12} = D_{21} = 0.75$	124
5.9	Initial and final image of a microstructure estimated using Eq (5.18) with $D_{11} = D_{22} = 2.5, D_{12} = D_{21} = -1.5$	124
5.10	Denoising 3D microstructure data as a graph labeling problem. (a) Noisy raw data from experiments (b) Microstructure after smoothening procedure using $\mathcal{N}_1$ (c) Microstructure after smoothening procedure using $\mathcal{N}_{\sqrt{6}}$	126
6.1	(a) Voronoi diagram with the generators and Voronoi points, (b) geometrical method to obtain the generator points from the Voronoi diagram.	134
6.2	Method to obtain the 2D convex microstructure: (a) the original experimental microstructure (grain A represents a twinned region, which is an internal facet of the parent grain), (b) the convex hulls of individual grains, (c) the resulting convex microstructure after dividing the intersecting regions among the individual grains.	136
6.3	Method to eliminate the intersecting regions of the convex hulls: (a) grains 1, 2, and 3 with their overlapping convex hulls, (b) Step 2 is followed for grains 1 and 2, (c) Step 2 is followed for grain 3.	137
6.4	Method to obtain the generator points from a convex microstructure: (a) rays $r_1, r_2, \dots, r_6$ do not converge to a single point, therefore the grain A is represented by six Voronoi generators $A_1, A_2, \dots, A_6$, (b) the Voronoi regions of the grain is indicated by dashed lines.	138
6.5	In the case of slender grains, Voronoi regions corresponding to some of the generator points can extrude into the neighboring grains across the long edges. (a) Voronoi regions of slender grains A and B extrude into grains B and C as indicated by the boxed regions. (b) Additional generator points are placed along the midpoint of edges $\overline{RS}$ and $\overline{TU}$ to bound the Voronoi regions within the grain.	139

6.6	Schematic diagram showing the construction of 3D microstructure from the given surface observation and EBSD map. Sections of the EBSD map are randomly chosen and stacked below the microstructure using their centroid as Voronoi generators.	141
6.7	Method to convexify the 3D microstructure: (a) the intersecting points of the convex hulls of the grains are removed; (b) the voxels within the gap between the grains are reassigned to its nearest grain using a linear least-square solver, the non-convex region of the partition is indicated.	142
6.8	Convexification of 3D microstructure: (a) original microstructure generated by stacking up the Voronoi generators, (b) the convexified 3D microstructure.	143
6.9	The First three layers of the 3D microstructure before convexification are shown on the top. The same three layers of the microstructure after convexification procedure are depicted below. Here, each color represents a different grain ID.	143
6.10	Three microstructures considered in this chapter are (a) convexified columnar microstructure with 46 grains, (b) variant 1 with 340 grains, and (c) variant 2 with 367 grains. Here, different stacks of EBSD images are used to build the variants 1 and 2. Additionally, for each microstructure, the perturbation of generator points along z -direction are distinct due to it being assigned randomly.	144
6.11	The recalculated pole figures for 2D EBSD data as well as 3D variants 1 and 2 in (0001), (10 $\bar{1}$ 0), and (10 $\bar{1}$ 1) directions are shown, respectively.	145
6.12	Experimental and simulated stress-strain curves for (a) tension (b) compression	148
6.13	Comparison of principal strain ϵ_1 from (a) SEM-DIC experiment, (b) convexified columnar microstructure, (c) 3D variant 1, and (d) 3D variant 2 are illustrated.	149
6.14	Comparison of relative activity of the various slip systems using quadrature point data from CPFE model for columnar microstructure, 3D variants 1 and 2. The m_{basal} denotes the Schmid factor for the basal slip system.	150
6.15	Comparison of the relative activity of the various slip systems using quadrature point data from CPFE model for 3D variant 2 along the depth of the sample.	152
7.1	(a,b) Transverse cracks under tension normal to fibers in a polymer matrix composite and intergranular stress corrosion cracking of Inconel (c,d) Graph cuts solution:green line indicates the computed crack trajectory.	158
7.2	3D fracture modelled using the graph coloring method proposed in this thesis.	159
7.3	(left) Synthetic microstructure from MRF algorithm. (right) Use of graph cut segmentation to identify the grain interfaces (shown in black)	160

LIST OF TABLES

Table

1.1	Benefits and shortcomings of existing AM microstructure modeling techniques in literature.	5
2.1	Breakdown of computational cost associated with highest resolution ($128 \times 128 \times 128$) level of the MRF reconstruction. For reference, all reconstructions are executed using a single processor and 4 GB of memory.	38
4.1	Crack propagation angle in mixed-mode failure	91
4.2	Crack propagation angle in mixed-mode failure	93
4.3	Crack propagation angle in mixed-mode failure	99
6.1	Hardening parameters for simulating stress-strain curves.	148

LIST OF APPENDICES

Appendix

A.	Theorems and Propositions	162
B.	Pseudo-Codes	173

ABSTRACT

The problem of microstructure reconstruction is deeply intertwined with the theories of microstructure quantification. Currently available methods for microstructure synthesis such as geometry based (Voronoi models), physically based (Phase field models) or feature-based (Simulated annealing) methods run into various difficulties when modeling real complexities of microstructures that include non-equilibrium features, non-convex grains, twins, second phases and cell structures that arise from thermomechanical processing. These features play an important role in the properties and performance of modern aerospace alloys. Further, microstructures are stochastic and lead to location-specific variability in material properties. In this proposal, we delve into a mathematical model that is expected to provide a better alternative for microstructure synthesis: Markov random fields.

In the report, theory and software for building 3D microstructural maps of engineering components through inference from 2D measurements is proposed. The approach is based on a Markov random field reconstruction approach developed by the PI. The method was extended to predict 3D microstructural map of large specimens using novel spatial sampling and reconstruction strategies. The first two chapters of this report present (i) a Markov random field algorithm that will allow 3D microstructure inference from 2D experimental data. (ii) Synthesis of microstructures into a CAD geometry of the component by merging the synthesized 3D microstructures. Because the method builds global models based on locally extracted data, we term the project LEGOMAT: Locally Extracted Globally Organized Microstructure Models through

Autonomy.

Computational methods are rapidly emerging as an essential tool for understanding and solving complex engineering problems, which complement the traditional tools of experimentation and theory. When considered in a discrete computational setting, many engineering problems can be reduced to a graph coloring problem. However, graph techniques are under-utilized in materials science. The computational techniques developed in the later chapters allow the representation of surface parameters of microstructures, such as perimeter and area, using discrete variables in a graph. Results from integral geometry, specifically Cauchy-Crofton relations, are used to estimate these scalars via submodular functions. With this framework, several quantities important to engineering applications can be represented in graph-based algorithms. These include the surface energy of cracks for fracture prediction, grain boundary energy to model microstructure evolution, and surface area estimates (of grains and fibers) for generating conformal meshes. Combinatorial optimization problems for these applications are presented in chapters 3–5. In particular, the methods provide a path for physics-based segmentation of synthesized microstructures, allowing inclusion of grain boundary properties and stereology, an area that should be the basis of future work.

CHAPTER I

Markov random Fields for Microstructure Reconstruction

An image-based framework is developed and examined for microstructure synthesis over large length-scales. Here, a set of three 2D surface micrographs of an Additively-Manufactured (AM) 316 L stainless steel specimen are first obtained experimentally from Electron Backscatter Diffraction (EBSD) and subsequently converted to a 3D representative unit cell using the Markov Random Field (MRF) technique. Features such as orientation and grain boundary misorientation distribution functions are used to validate the reconstructed 3D microstructure against experimental EBSD images. The variations of 3D microstructures during the Selective Laser Melting (SLM) process are then modeled by merging a single reconstructed 3D microstructure in an engineering-scale geometry via a patch-based optimization process that maps every finite element within the geometrical domain to a microstructural mesh in the unit cell. This new approach provides a workflow for reconstructing real-time global maps of AM microstructures by embedding site-specific images along user-specified laser paths while accounting for hatch spacing, layer thicknesses, and scan directions. The results are specifically useful for visualization of process-microstructure relationships in additive manufacturing processes.

1.1 Introduction

With the emergence of the paradigm of Integrated Computational Materials Engineering (ICME) [2], multiscale design optimization approaches for tailoring engineering properties of Additively-Manufactured (AM) structures through controlled processing parameters [3, 4, 5, 6, 7] are of great interest to the design and additive manufacturing communities. Such multi-level simulations often involve solving the process-microstructure-dependent properties to fully define the governing equations for the macro-scale analyses. This requires models beyond the grain-scale level that can accurately capture the underlying morphology and texture of the materials [8]. The 2D microstructures are conventionally characterized through experimental instrumentations [9, 10, 11, 12]. However, sectional information obtained from empirical characterization techniques is generally insufficient for defining inherent microstructural parameters (e.g., connectivity, grain size/shape, and boundary networks, etc.) [13, 14]. Yet, these parameters are a critical aspect of the materials design and can strongly affect the capability of the structural systems to perform in their respective loading environments [15]. As a result, there has been a growing desire to develop accurate measurements, allowing for the direct acquisition of full 3D microstructural information [16, 17].

The need to precisely characterize the 3D microstructures has led to the development of several experimental methods, which are mainly variants of two major procedures: serial sectioning [18, 19, 20], and High-Energy Diffraction Microscopy (HEDM) [21, 22, 23, 24]. Serial sectioning is a destructive approach in which layers of controlled thickness are sequentially polished away with high accuracy. This enables metallographic etching and captures sectional images of a material's microstructure. Post-processing is followed to render these 2D images into a solid 3D model. Contrary to serial sectioning, HEDM is a non-destructive technique that uses high-energy X-rays to probe a relatively bulk specimen. The diffraction pattern in this approach

determines the crystallographic orientation of the AM microstructure. Acquiring 3D microstructures with the above experimental techniques is generally a prodigious process and requires the brilliance of expensive empirical scanning devices. Additionally, these procedures uncover microstructural information over a relatively small volume of materials at a time. Hence, a large number of specimens must be scanned and seamlessly merged to generate an engineering-scale Computer-Aided Design (CAD) model. Such limitations reduce the general applicability of the experimental procedures, and so urging the development of robust computational reconstruction strategies.

Current numerical methods for the prediction of AM microstructure distributions are mainly physics-based techniques that are requiring extremely large computational effort. As indicated in Table 1.1, amongst phase-field [25, 26], kinetic Monte Carlo [27], and cellular automata [28, 29] simulations, the phase-field modeling techniques are often considered the most accurate approaches that is adequately capturing the solute concentration, precipitates, and dendrites shape [30]. However, the advantage of using phase-field techniques is obscured by the required computational power, limiting the predictions to atomistic and continuum-length scales. Less costly than phase-field, cellular automata enables the scalability for larger domain sizes. However, the accuracy can diminish considerably with increasing the cell size [31]. Additionally, the kinetic Monte Carlo simulations (e.g., SPPARKS) [27, 32] allow large-scale prediction but have difficulty predicting the texture distributions and reconstructing the complex components such as non-equilibrium grain structures during the additive manufacturing processes. To overcome such shortcomings, there have been considerable efforts towards the development of statistical-based models to generate representative microstructures. For instance, works based on feature-matching algorithms using correlation functions [33] or Markov Random Field (MRF) methods [34, 35, 36] can simulate microstructural features efficiently and are significantly less expensive compared to the physics-based models.

Modern data-driven microstructure reconstruction methods such as MRFs can employ snapshots of conventional 2D prototypes to rapidly infer diverse groups of microstructures for new processing modalities such as hatch spacing, layer thickness, scan velocity, and effective laser energy density. Consequently, here in this chapter, a geometrical framework based on MRF technique [36] is used for the real-time description of microstructure distribution of AM materials by combining material flow fields that capture modeling variations in directionality, anisotropy, and grain sizes. Fundamental steps include (i) experimental acquisition of orthogonal scans of 2D microstructures, (ii) MRFs for 3D reconstruction of microstructures, and (iii) embedding the realized microstructures into a virtual specimen (a digital twin). As a result, the method builds global models based on locally extracted images, warranting the term LEGOMAT: Locally-Extracted Globally-Organized Microstructure Models. The first version of LEGOMAT, as presented in this chapter, is able to embed a single measured microstructure following the material flow path while allowing the user to input parameters such as hatch spacing, layer thickness, and the scan direction. The LEGOMAT algorithm reconstructs virtual AM specimens with microstructural information by mapping every finite element within the CAD model to a microstructural domain. The algorithm utilized an iterative graph-based convergence criterion, minimizing the difference between the tensor fields associated with embedding microstructures to the specified local parameters in the CAD geometry. As a result, such an approach can be used for rapid reconstruction of 3D maps of microstructures with billions of grains at the component-scale with user-defined embedding modalities.

The structure of the chapter is as follows. The MRF algorithm for reconstructing 3D images from three 2D orthogonal micrographs is briefly described in Section 1.2.1. The LEGOMAT algorithm to embed the MRF microstructures is reviewed next in Section 1.2.2. In the following, the optimization processes in correlation with material flow paths are described. Last, numerical and experimental results along with statisti-

Table 1.1: Benefits and shortcomings of existing AM microstructure modeling techniques in literature.

Method	Computational Cost	Benefits	Challenges
Phase-Field prediction	Extremely high	Physics-based	Small-scale
Cellular Automata on cell size	High	Texture prediction	Accuracy depends
Kinetic Monte Carlo (e.g., SPPARKS) prediction	Intermediate Texture prediction	Allows large-scale	
LEGOMAT prediction dependent	Real-time Experimental data-	Allows large-scale	

cal comparisons for verification of crystal orientation distribution and grain boundary misorientation angles of the proposed methodology are presented in Section 1.3.

1.2 Methodology

In this section, the mathematical formulation and implementation for reconstructing large-scale 3D AM microstructures is discussed. The overall process, as illustrated in Fig. 1.1, utilizes two separate algorithms. First outlined in Section 1.2.1, the MRF theory, from previously-published work [36] by authors, is summarized. There, the 2D microscopy imaging is leveraged to reconstruct a representative 3D microstructural unit cell using three orthogonal planar exemplars. Next, Section 1.2.2 describes the large-scale synthesis process for embedding site-specific unit cells through a optimization procedure, merging microstructures into a CAD model.

1.2.1 Generation of 3D Unit Cells - MRF Sampling Algorithm

The input to the MRF algorithm consists of three orthogonal planar Electron Backscatter Diffraction (EBSD) exemplars of AM microstructure, whereas the output is a representative 3D solid model of the respective microstructure. The overall process

closely follows the original procedure from [36]. In the following, the key ingredients of the process are described briefly. Let $\mathbf{S}^i$ for $i = \{x, y, z\}$ denote a set of three orthogonal experimentally-obtained slices of microstructures along the respective x , y , and z directions. The symbol $\mathbf{V}$ indicates the synthesized 3D microstructure, with $\mathbf{V}_v$ representing the Red-Green-Blue (RGB) coloring of voxel v . In the realm of microstructures, RGB channels (as a triplet) are used for the representation of the phase information (e.g., grain size/shape, crystal orientations, chemical compositions, boundary networks, etc.). Based on the Markovian assumption, the Probability Density Function (PDF) of a voxel given the states of its spatial neighborhood is independent of the entire dataset. Thus, only neighbors over a relatively small user-specified window around the voxel v are examined throughout the sampling process. As seen in Fig. 1.2(a), the vector denoting the spatial neighborhood of voxel v in the slice orthogonal to the i axis is denoted as $\mathbf{V}_v^i$. Additionally, let $\mathbf{S}^{i,w}$ denote a window of the same size in its respective 2D micrograph, $\mathbf{S}^i$. Using the Markovian properties, the most likely RGB triplet of the voxel v can be estimated by identifying center pixel values of $\mathbf{S}^{i,w}$ in the planar micrographs that best resemble the corresponding cross sections of $\mathbf{V}_v^i$. These windows, as shown in Fig. 1.2(b), are denoted by $\mathbf{S}_v^i$, where $i = \{x, y, z\}$.

Subsequently, let the value $\mathbf{V}_{v,u}^i$ denote the RGB coloring of voxel u in the neighborhood of $\mathbf{V}_v^i$. Let the values $\mathbf{S}_{v,u}^i$ and $\mathbf{S}_u^{i,w}$, respectively denote the RGB triplets of pixel u in the windows $\mathbf{S}_v^i$ and $\mathbf{S}^{i,w}$. Consequently, the best-matching neighborhood of voxel v , $\mathbf{S}_v^i$, in each orthogonal planer image is selected by solving the following minimization problem:

$$\mathbf{S}_v^i = \arg \min_{\mathbf{S}^{i,w}} \sum_u \omega_{v,u}^i \|\mathbf{V}_{v,u}^i - \mathbf{S}_u^{i,w}\|_2^2 \quad (1.1)$$

where $\|\cdot\|_2$ represents the L_2 norm, and $\omega_{v,u}^i$ denotes a per-pixel weighting factor, preserving the short-range correlations of the nearby pixels/voxels.

The process outlined in Eq. 1.1 consists of an exhaustive search that compares all the windows in the input 2D exemplars, $\mathbf{S}^{i,w}$, to the corresponding neighborhood of voxel v , $\mathbf{V}_v^i$, and identifies windows, $\mathbf{S}_v^i$, that lead to a minimum weighted squared Euclidean distance. Generally the center pixel values in $\mathbf{S}_v^i$ are composed of distinct RGB triplets. Yet, an optimal value of $\mathbf{V}_v$ needs to be inferred by weighting colors pertaining to location v in not only the matching windows of voxel v but also its surroundings, as follows:

$$\mathbf{V}_v = \left(\sum_{i \in \{x,y,z\}} \sum_u \omega_{u,v}^i \mathbf{S}_{u,v}^i \right) / \left(\sum_{i \in \{x,y,z\}} \sum_u \omega_{u,v}^i \right) \quad (1.2)$$

note that the subscripts u and v are switched in Eq. 1.2, compared to Eq. 1.1. This implies that the optimal color of the voxel v , $\mathbf{V}_v$, is the weighted-average of the colors at locations corresponding to voxel v in the best-matching windows of voxels u found within the synthesized 3D microstructure. The RGB channels here are averaged independently in Eq. 1.2. Since $\mathbf{V}_v$ is continuously changing after each step, the set of closest input neighborhoods $\mathbf{S}_v^i$ varies accordingly. Hence, the above two steps are repeated until convergence; that is until the set of $\mathbf{S}_v^i$ remain unchanged.

The color space associated with micrograph images is typically discrete and range bound. Consequently, the averaging performed in Eq. 1.2, always tends to shrink the color levels. For instance, the color level 0 has the tendency to increase, due to it being averaged with all the color levels that are greater or equal to 0. However, the assumption in MRF reconstruction remains that the two datasets (i.e., three orthogonal planar images and solid synthesized model) have the same Cumulative Distribution Function (CDF) of color densities. As a result, given the reference exemplars and the target synthesized images, the algorithm normalizes the color histograms at the end of each iteration by first binning the colors into 255 discrete intervals. Next, histogram matching, as demonstrated in Fig. 1.3 is applied independently for each color channel such that the color density of synthesized model closely matches with

the planar exemplars [36].

Although the methodology presented here is not tied to any specific image resolution, all the input exemplars are initially resampled to 128×128 pixels prior to the 3D reconstruction. Then, a random RGB triplet from the input planar exemplars is assigned to each voxel v , as an initial condition. The reconstruction process is then carried out in a multi-resolution framework: starting with a coarse voxelated mesh while progressively interpolating the results to a finer resolution. The multi-resolution approach considerably decreases the convergence rate associated with the sampling algorithm. As such, the resolution levels (32^3 , 64^3 , and 128^3) are subsequently used for all presented MRF reconstruction examples. In the following section, the numerical process associated with the LEGOMAT for embedding the location-specific unit cell in an engineering-scale geometry is discussed.

1.2.2 Large-Scale Synthesis - LEGOMAT Embedding Algorithm

The inputs of the LEGOMAT algorithm consist of a geometrical CAD model and an exemplar solid 3D unit cell of an AM microstructure as outputted from the MRF reconstruction process. In return, the output is of a 3D CAD model, filled with patches of the microstructure from the exemplar unit cell. Here, the input CAD model is first converted to a tetrahedral mesh model via Delaunay tetrahedralization [37]. In the case of laser powder-bed technology, the solidification process results in the formation of highly-localized non-equilibrium microstructural components within the fusion zone [38, 39]. Preferred crystallographic growth directions in Selective Laser Melting (SLM) tend to align the columnar grains in the direction of increasing temperature. As a result on the BD-TD face, the columnar grains tend to grow perpendicular to the melt pool's solidification surface, while curving towards the laser direction [40, 41]. Accordingly, the alignment of such complex features in 3D geometrical space while using a single site-specific reference 3D snapshot, requires a tensor field (i.e., a set of

three orthogonal vector fields) underlying material growth orientations. Such a tensor field can systematically capture the preferred local crystallographic growth direction of columnar grains throughout the solidification process.

In the current implementation, as illustrated in Fig. 1.4(b), a volumetric vector field is specified over the CAD model, simply by manually drawing the material growth directions on the surface and cross-sections of the CAD model. The second direction of the tensor field is then chosen randomly upon embedding each microstructural element, and the third direction is set to be the cross product of the the two vectors. Alternatively, the volumetric tensor field can be inferred directly by the experimental measurements in the proximity of the fusion zone. However, such measurements are only possible to capture on external surfaces of the build and not at the interior locations. Thus, one can incorporate transient computational heat-transfer simulations to infer the spatial temperature fields that ultimately can inform the material flow directions [42, 43]. For interpolation of the user-specified vectors, a Laplacian smoothing is utilized [44] which has been previously employed in lapped texture synthesis methods [45, 46]. Once the material flow paths are sketched, the local scaling of the grain sizes, as seen in Fig. 1.4(c), is specified to incorporate size-varying crystals in the CAD model. Subsequently, the algorithm repeatedly pastes the microstructure from the unit cell onto the CAD model while aligning the embedding microstructures according to the local tensor field and specified scaling parameter. The result of the embedding process is shown in Fig. 1.4(d).

Contrary to the voxelated approach outlined in Section 1.2.1, a patch-based optimization technique is implemented here that uses a tetrahedral discretization to represent the solid geometrical CAD models. This formulation has certain benefits over voxel representation. The main advantage is it requires a substantially smaller number of elements to approximate the 3D CAD model, resulting in considerably fewer iterations (and less computational power) to complete its embedding task.

1.2.2.1 LEGOMAT Embedding Procedure

The embedding process consists of the following steps and follows the procedure outlined in Ref [46], that was originally developed for 2D surface embedding. In the process of volumetric embedding, a group of N tetrahedron elements ($T_i, i = 1, \dots, N$) is selected in the CAD model such that there are small variations in the user-defined tensor field across this group. The size of this group is also chosen such that it can be contained within the representative microstructure. The tetrahedron located at the center of this group is denoted as the seed tetrahedron, T_o . The embedding is based on an optimization process that computes the mapping of the vertices of these tetrahedra to locations in the microstructure that satisfies the user-defined tensor field. Let $\mathbf{v}_1, \dots, \mathbf{v}_4$ represent the four vertices of a tetrahedral element T_o in the CAD model, then the primary aim is to identify the linear mapping of these nodes to positions $\Phi(\mathbf{v}_1), \dots, \Phi(\mathbf{v}_4)$ in the representative microstructure.

Here, let $\tilde{\mathbf{P}}$, $\tilde{\mathbf{Q}}$, and $\tilde{\mathbf{R}}$ represent the set of three orthogonal vectors of the tensor associated with the tetrahedron, T . The barycentric coordinates $p_1, \dots, p_4$ on tetrahedral element T_o is used to represent $\tilde{\mathbf{P}}$ as follows:

$$\begin{cases} p_1 \mathbf{v}_1 + p_2 \mathbf{v}_2 + p_3 \mathbf{v}_3 + p_4 \mathbf{v}_4 = \tilde{\mathbf{P}} \\ p_1 + p_2 + p_3 + p_4 = 0 \end{cases} \quad (1.3)$$

Similarly, $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{R}}$ can be represented using $q_1, \dots, q_4$ and $r_1, \dots, r_4$, respectively. Since the mapping from $\mathbf{v}_1, \dots, \mathbf{v}_4$ to $\Phi(\mathbf{v}_1), \dots, \Phi(\mathbf{v}_4)$ is linear, the mapping of vector $\tilde{\mathbf{P}}$ in the CAD model to a vector $\Phi(\tilde{\mathbf{P}})$ in the microstructural space follows the same weighting:

$$\Phi(\tilde{\mathbf{P}}) = p_1 \Phi(\mathbf{v}_1) + p_2 \Phi(\mathbf{v}_2) + p_3 \Phi(\mathbf{v}_3) + p_4 \Phi(\mathbf{v}_4) \quad (1.4)$$

Similarly, $\Phi(\tilde{\mathbf{Q}})$ and $\Phi(\tilde{\mathbf{R}})$ can be obtained from $q_1, \dots, q_4$ and $r_1, \dots, r_4$, respectively. Ideally, when mapped into the microstructure, these vectors should be aligned with

the orthogonal x , y , and z coordinate vectors in the microstructure. However, to enforce microstructural continuity in the CAD model, small differences are permitted. Difference vectors $\mathbf{d}_p, \mathbf{d}_q, \mathbf{d}_r$ are defined for each tetrahedra in the group as follows:

$$\begin{cases} \mathbf{d}_p = \Phi(\tilde{\mathbf{P}}) - (1, 0, 0)^t \\ \mathbf{d}_q = \Phi(\tilde{\mathbf{Q}}) - (0, 1, 0)^t \\ \mathbf{d}_r = \Phi(\tilde{\mathbf{R}}) - (0, 0, 1)^t \end{cases} \quad (1.5)$$

The optimization problem is to find the vertex mapping $\Phi(\mathbf{v}_1), \dots, \Phi(\mathbf{v}_4)$ for all the tetrahedra T_i within the patch such that the least squares functional defined below is minimized:

$$\sum_{T_i} \|\mathbf{d}_1\|^2 + \|\mathbf{d}_2\|^2 + \|\mathbf{d}_3\|^2 \quad (1.6)$$

The minima is unique up to a translation vector and thus, an additional constraint is used to fix the center of the seed tetrahedra to the center of the microstructural volume:

$$\frac{\Phi(\mathbf{v}_1) + \Phi(\mathbf{v}_2) + \Phi(\mathbf{v}_3) + \Phi(\mathbf{v}_4)}{4} = (0.5, 0.5, 0.5)^t, \text{ with } \mathbf{v}_i \in T_0 \quad (1.7)$$

The optimized solution $\Phi(\mathbf{v}_j)$ to the constrained minimization problem can be efficiently obtained by solving a sparse linear system. The procedure is continued at another location with an unmapped tetrahedra and the process repeated until all the vertices of the tetrahedra in the CAD model are mapped to the microstructure.

1.3 Results and Discussion

In this section, the process parameters for manufacturing and image acquisition of EBSD sections of an AM 316 L stainless steel specimen is explained. Thereafter, the

results of reconstructing digital twins using a set of locally orthogonal planar images is highlighted. Here, the efficacy of the algorithm is also demonstrated by comparing the crystallographic orientations/misorientations of the synthetic and experimental images.

The 316 L stainless steel specimen was fabricated via an SLM 280HL machine, equipped with two 400 W Continuous Wave (CW) Ytterbium fiber laser beams with approximately 80 μm diameter at the focal point. The SLM 280HL machine is composed of a building platform with dimensions $280 \times 280 \times 350 \text{ mm}^3$. Prior to the operation, argon gas was flooded into the chamber to lower the oxygen level below 0.1%. The pre-alloyed 316 L stainless steel powder, with particle sizes between 30 – 50 μm , was used as the printing substance. A fully-dense cuboid of $10 \times 6 \times 10 \text{ mm}^3$ sample was fabricated for microscopic analysis, using manufacturer recommended guideline with core processing parameters of 200 W laser power, 800 mm/s scan speed, 30 μm layer thickness, and 120 μm hatch spacing. The AM volume was printed using a bi-directional scanning technique at which the laser beam moves across each layer in a zigzag pattern. The cuboidal sample was then sequentially polished through diamond suspensions of 9 μm , 6 μm , and 3 μm . This was followed by alumina suspensions of 1 μm and 0.05 μm , resulting in smooth faces. The EBSD analyses of three orthogonal $700 \times 700 \mu m^2$ faces of the cuboidal specimen were subsequently performed using a high-resolution Scanning Electron Microscopy (SEM) TESCAN MIRA3 at 30 kV voltage and 1.0 μm step size.

Once the experimental imaging was finalized, the raw Euler angles were mapped to nodes in a discretized Orientation Distribution Function (ODF). The ODF provides a probability distribution of the crystallographic textures by describing the Euler angles associated with each node in the Rodrigues space. Here, the crystal orientation is represented based on the unique association of a rotation axis, denoted as $\mathbf{n}$, and a counterclockwise angle, represented by θ , about $\mathbf{n}$. Afterwards, the Rodrigues'

parametrization, denoted as $\mathbf{r}$, can be computed as $\mathbf{r} = \mathbf{n} \tan(\frac{\theta}{2})$. Accordingly, a proper rotation relates the sample orientation to the crystal orientation. The lattice structure of the material of interest in this chapter (316 L stainless steel) is of cubic symmetry. Such a lattice structure reduces the ODF to a small subset, called the fundamental region, accounting for the cubic symmetry of the scanned material. The resulting fundamental Rodrigues region is illustrated in Fig. 1.6(a). For numerical analysis, the fundamental region can be discretized using a finite-element mesh. However, due to cubic symmetries, several of the nodes in the grid are equivalent. Hence, the ODF is represented using a smaller set of independent nodes colored blue, as shown in Fig. 1.6(b). Accordingly, each point in the EBSD dataset is mapped to its closest independent node in the discretized ODF space. The scanned EBSD images are then colored according to the nodal numbers. A simple choice for the colormap, as depicted in Fig. 1.6(c), is to apply the Rodrigues vector itself for each independent node as an RGB triplet. Subsequently, for MRF reconstruction, a subset of experimental EBSD sections of size $256 \times 256 \mu m$ are selected and resampled to 128×128 pixels. Following this, the texture colorings corresponding to BD-SD and BD-TD images were remapped such that they represent a similar color histogram as of SD-TD image. The SLM manufacturing processes tend to create large variations in the crystallographic texture across a specimen. As such, performing *Histogram Matching* can establish a consistent texture as images are taken at different locations within the specimen. Fig. 1.7(a) shows the orthogonal set of images for 316 L stainless steel specimen prior to the MRF reconstruction. The 3D EBSD image from the MRF algorithm is shown in Fig. 1.7(b). A number of different sections of the reconstructed model are shown in Fig. 1.7(c).

Following the reconstruction of the unit cell, each color level of the synthesized model is mapped to an independent node in the fundamental Rodrigues region using the colormap shown in Fig. 1.6(c). For every RGB triplet, four closest (in Euclidean

norm sense) colors in the discretized ODF space are identified. The voxel coloring remains unaltered, provided the sampled RGB triplet is within the specified threshold. Otherwise, the RGB coloring of the voxel is replaced randomly with one of the four closest independent nodes in the ODF fundamental region. As depicted in Fig. 1.8(a)-(b), it is found that the resulting pole figures of the MRF reconstructed microstructure closely resembles the measured texture distribution.

Once the RGB triplets are mapped through the discretized ODF colormap, the misorientation angles can be conveniently calculated. Hence, after the reconstructed image is fully segmented, the misorientation angles are computed slice by slice using eight-fold connectivity, where the crystal orientation (i.e., Euler angles) of every pixel is being compared against its adjoining neighbors (along the horizontal, vertical, or diagonal direction) on the cross-section. If the frequency of the most repeating orientation index in the 3×3 window is within a specified threshold, the two most dominating Euler angle triplets inside the window are then identified. Following this, the misorientation angle between the two most-frequently occurring orientation values is computed. The same process is followed for the orthogonal experimental images. Fig. 1.9(a)-(b) illustrates the spatial distribution of misorientation angles for experimental and synthesized EBSD images, respectively. The corresponding distribution of misorientation values of the sections in Fig. 1.7(c) are also depicted in Fig. 1.9(c). The probability densities of the raw misorientation angles for the set of three orthogonal images and cross-sections of the reconstructed model, as seen in Fig. 1.10 are then calculated over 18 bins using a log-normal fit. Because of the cubic symmetries of the measured specimen's lattice structure, the maximum unique misorientation angle possible is limited to 62.8° . As such, both the log-normal distributions were cut off at 62.8° , and the results were subsequently renormalized. Overall, a close agreement between the measured and reconstructed model is observed, with the 3D reconstruction predicts marginally a higher probability for misorientation angles ranging in [40

52].

Thereafter, LEGOMAT reconstruction technique is followed by building global models, as shown in Fig. 1.11, based on a single locally-extracted microstructure. The CAD models for the block geometries in Figs. 1.11(a)-(b) contain 24948 and 38366 tetrahedral finite elements, respectively. Here, the LEGOMAT approach is to embed the reconstructed 3D microstructure in the CAD geometry based on experimentally observed insights on AM microstructure formation as it is controlled by laser parameters (e.g., speed, power) and the laser path (e.g., hatch spacing, layer thickness). The effects of cooling rate and temperature gradient history on microstructural patterns during the solidification process have been well studied in literature [47, 40, 41]. Key components seen in the solidification process include (1) smaller grain growth near the laser track because of high-temperature profile at the solid-liquid interface [29], and (2) elongated grains away from the laser spot in-plane to the surface. The preferred growth direction of elongated grains can be controlled by heat flow direction, whereas the grain size decreases with increasing laser speed or decreasing laser power [48]. (3) On BD-TD face, the elongated grains tend to grow perpendicular to the melt pool's solidification surface, bending towards the laser track [29]. As a result of such geometrical principles, a natural way to describe grain formation after the solidification process is to align microstructures in the 3D geometrical space using a tensor field as shown in Fig. 1.11(left). The grain size field is represented using the same approach as shown in Fig. 1.4, with smaller grains across the laser path. This geometric reconstruction approach can systematically capture the preferred grain growth direction along the laser path as observed experimentally. Fig. 1.11(right) shows the final reconstructed outputs displaying the grain sizes and shapes that result from the LEGOMAT procedure.

1.4 Conclusions

In this chapter, an image-based framework was introduced for 3D microstructure synthesis of Additively-Manufactured (AM) materials over very large length-scales. The complete workflow from the 2D experimental EBSD images to a CAD model visualization is an important and novel contribution in this work. Here, a set of three Electron Backscatter Diffraction (EBSD) sections imaged on orthogonal planes of a 316 L stainless steel specimen from the Selective Laser Melting (SLM) process was first obtained via microscopy imaging. The microstructures are subsequently converted to a 3D representative unit cell using the Markov Random Field (MRF) modeling technique. The MRF algorithm reconstructs 3D images through matching 3D slices at different voxels to the experimental sectional micrographs while ensuring neighborhoods of voxels taken from the 2D micrographs have meshed together seamlessly in the 3D image. Thereafter, statistical analyses of textural features, such as Orientation Distribution Function (ODF) and grain boundary misorientation angles for the reconstructed 3D microstructures were carried out against the original 2D EBSD images. The results demonstrate that MRF can effectively provide 3D texture prediction of AM 316 L stainless steel microstructure. The variations of 3D microstructures during SLM at larger length-scales are then captured through user-specified processing parameters such as laser energy density (speed and power) as well as scanning path (hatch spacing and layer thickness). The LEGOMAT procedure embeds 3D microstructures in an engineering-scale Computer-Aided Design (CAD) geometry accordant with a material flow path, accounting for variations in grain size and grain shapes (equiaxed vs. elongated) in additive manufacturing. Here, the grain sizes and shapes are defined in terms of a tensor field based on known microstructure patterns seen from the experimental characterization technique. For instance, smaller grains are observed near the laser track, whereas elongated grains are mainly located away from the laser path in-plane to the surface. Hence, the preferred growth direc-

tion of elongated grains is controlled by laser parameters, where the size of elongated grains decreases with increasing laser speed or decreasing laser power. As a result, the LEGOMAT approach can provide in real-time an efficient methodology to describe such features in 3D geometrical CAD space via a user-defined tensor field capturing the preferred growth direction during the solidification process. Because the large-scale microstructure models are primarily based on geometry, the reconstructions are extremely robust and require minimal computational power when compared to existing phase-field simulations. Although the preliminary study presented in this chapter shows significant promise of the LEGOMAT approach, future work will focus on increasing the resolution of the reconstructions and development of methods to reduce seams that occur along the intersecting surfaces of tetrahedral groups in the CAD model. In the presented cases, these seams are not pronounced due to the highly stochastic nature of the additive microstructure. The approach is specifically useful as an efficient visualization tool to understand microstructure variability as a function of process parameters and identify possible weak links (fracture paths) at larger scales than what can be measured experimentally. The current work can benefit from machine learning methods to map location-specific crystallographic texture to the reconstructed 3D grain structure and to identify outliers in the reconstruction where additional experimental data can be used to improve upon the visualization.

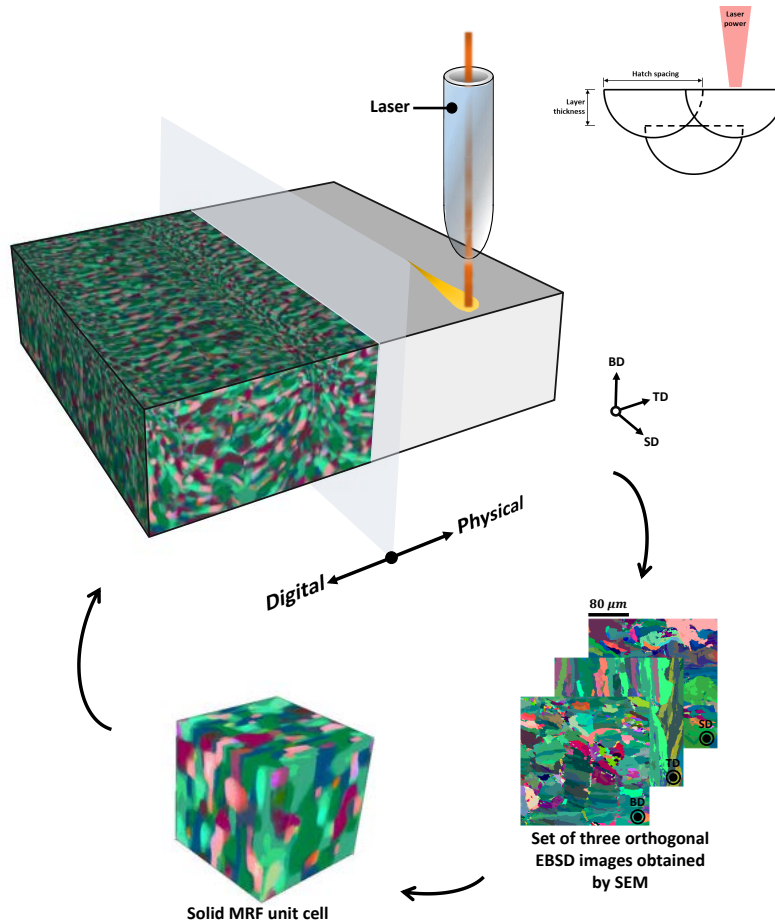


Figure 1.1: *LEGOMAT* approach: full-scale component imaging built by using a single site-specific MRF unit cell. A small-scale texture of the AM specimen is extracted through EBSD imaging of three orthogonal planes. The 3D microstructure is then reconstructed via the MRF technique. The laser path, along with other processing parameters such as layer thickness and hatch spacing (imposed by the user) are then used to embed the crystallographic elements from MRF unit cell onto the CAD model. The BD, SD, and TD axes here represent the Building, Scanning, and Tangential Directions, respectively.

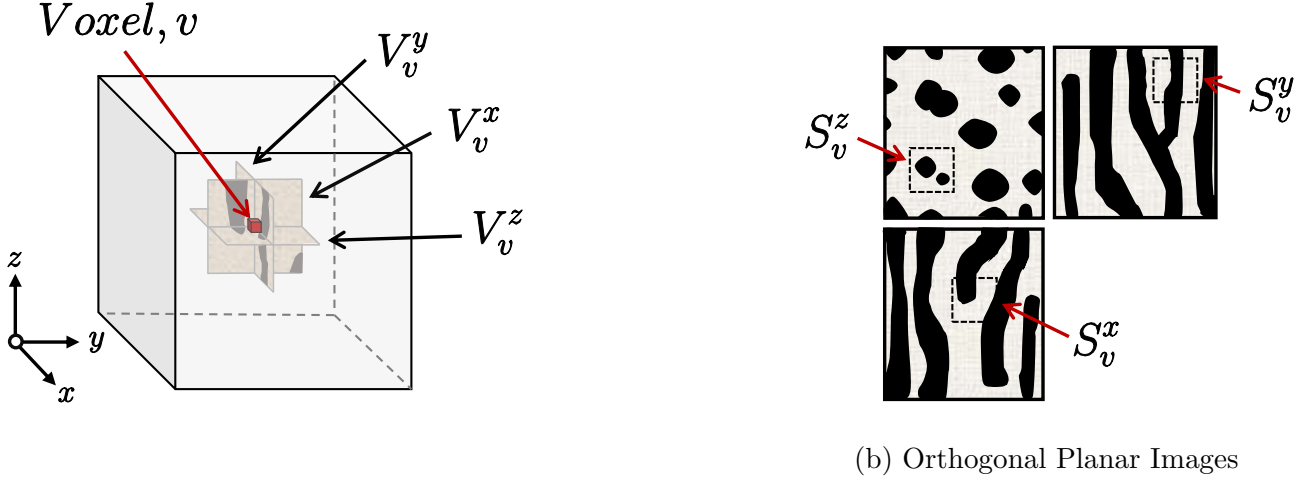


Figure 1.2: *MRF unit cell reconstruction schematic: (a) the neighborhoods of the voxel v in the slices orthogonal to the $i = \{x, y, z\}$ axes, denoted as V_v^i , are shown. (b) The windows in the input 2D micrographs shown in dotted lines are denoted by S_v^i . These windows closely resemble the neighborhoods of the orthogonal cross sections of voxel v , V_v^i .*

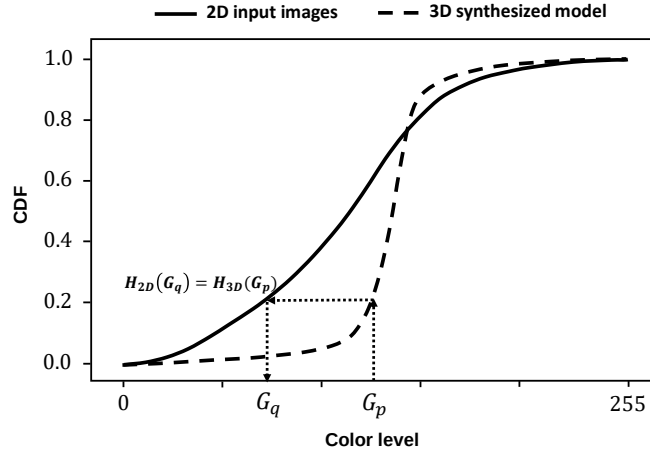


Figure 1.3: *Histogram matching algorithm: given the reference (2D) and the target (3D) images, the color histograms are obtained by binning the colors into 255 discrete intervals. Cumulative Distribution Functions (CDFs) of the two color histograms, denoted as $H_{2D}(\cdot)$ and $H_{3D}(\cdot)$ are then computed and normalized. The color level G_p in the synthesized 3D image is switched for each RGB channel independently. The new color G_q has the same cumulative distribution function as in input 2D exemplars.*

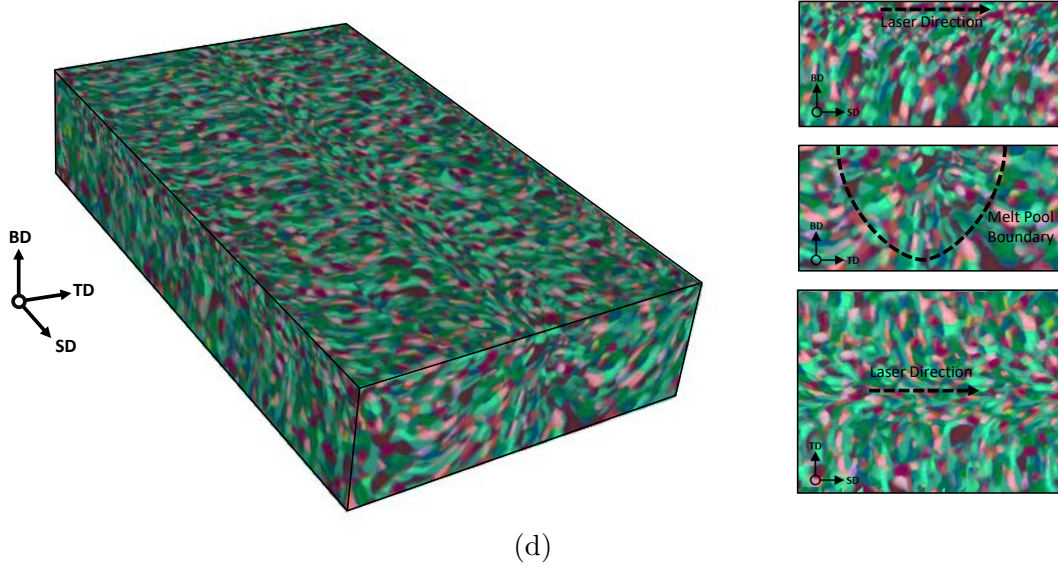
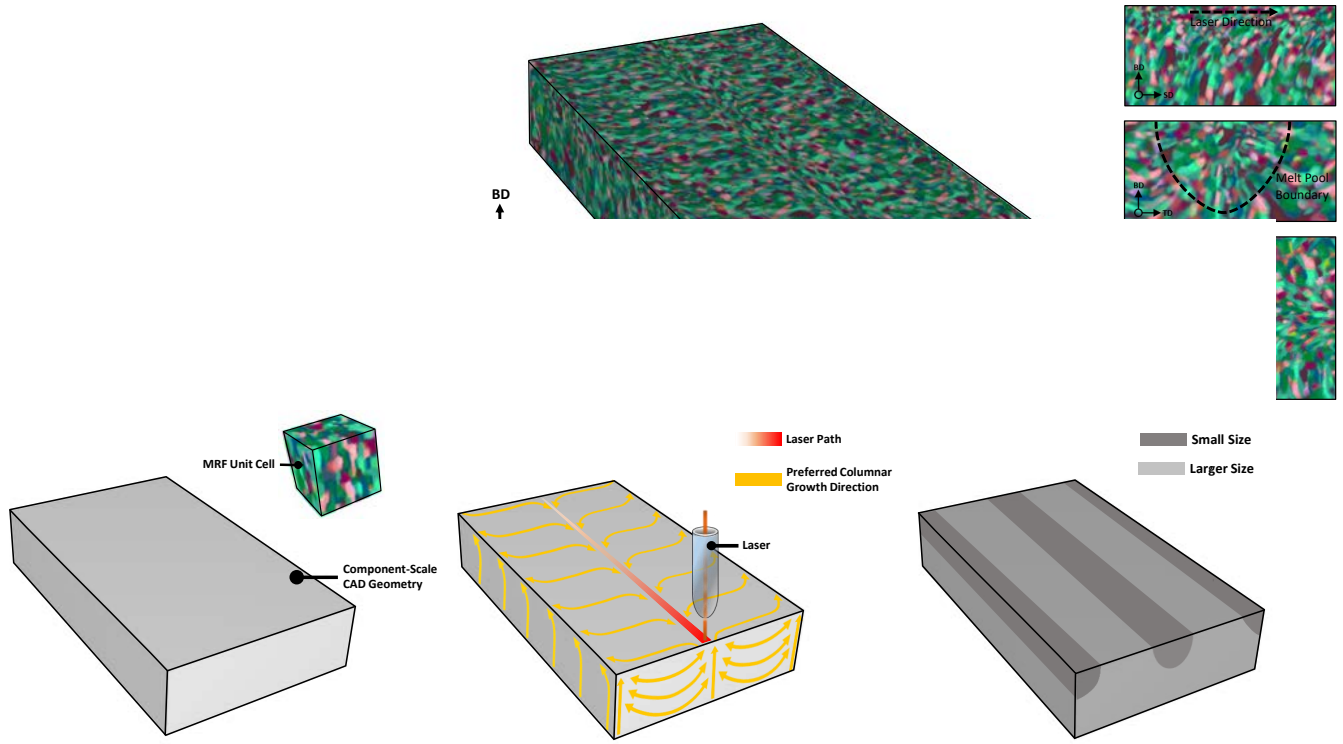


Figure 1.4: Illustration of LEGOMAT embedding process: (a) Inputting the geometrical mesh along with 3D solid microstructure, (b) manually drawing the local vector fields, representing the preferred crystal growth orientation, (c) setting up the grain scaling by embedding larger grains farther from the laser path, (d) result of LEGOMAT's automatic embedding procedure, along with snapshots of three orthogonal cross-sections in the realized structure.

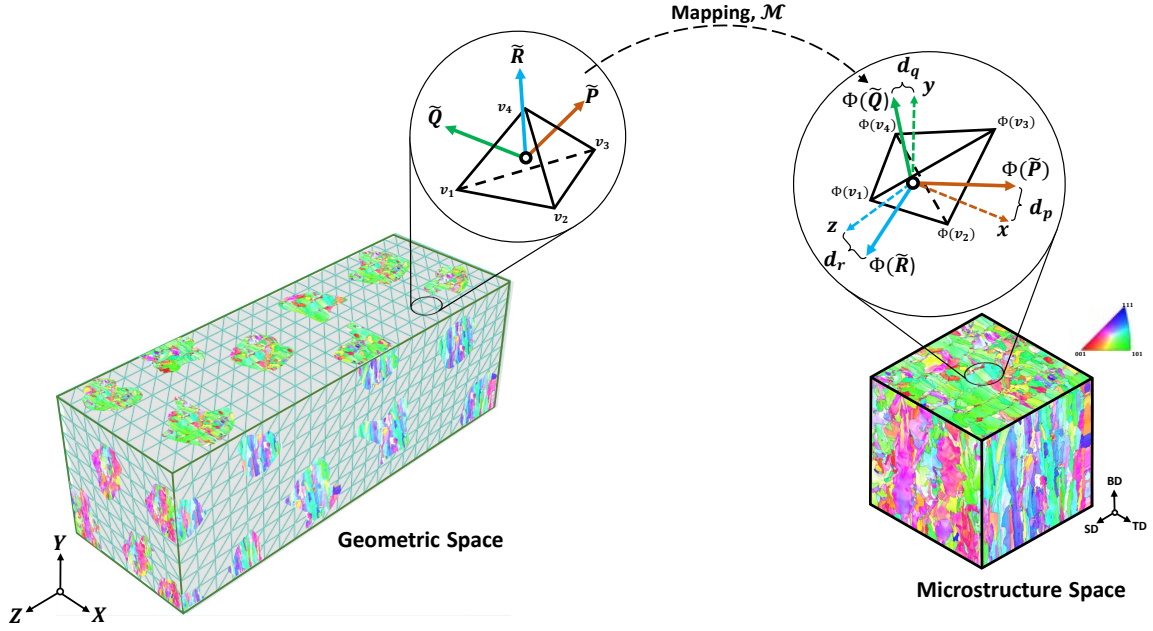


Figure 1.5: *LEGOMAT optimization schematic: the optimization process minimizes the difference vectors d_p , d_q , and d_r between microstructure coordinate axes (x, y, z) and the transformed tensor axes $(\Phi(P), \Phi(Q), \Phi(R))$ for each tetrahedron.*

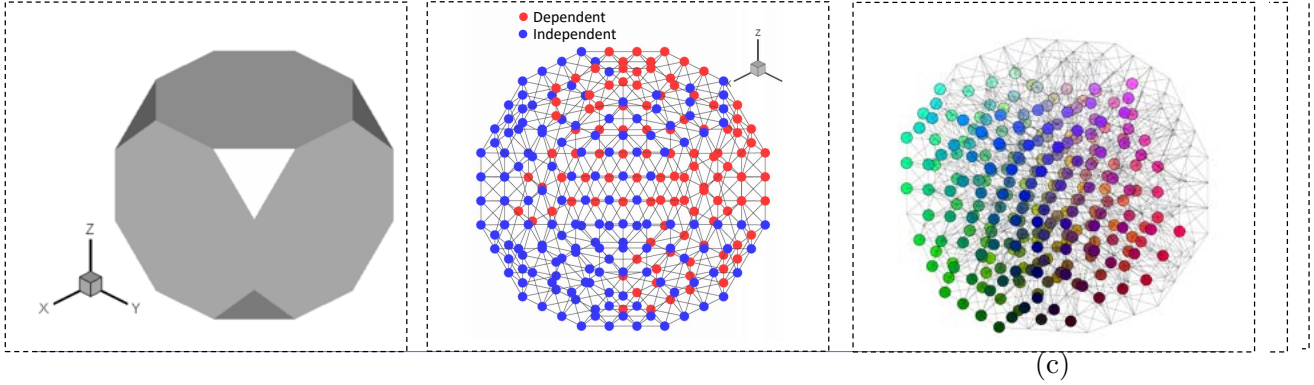


Figure 1.6: *Illustration of Orientation Distribution Function (ODF): (a) the Rodrigues fundamental region for cubic symmetry, (b) a discretized ODF with independent nodes colored blue, (c) colormap showing the RGB values of independent nodes in the fundamental region.*

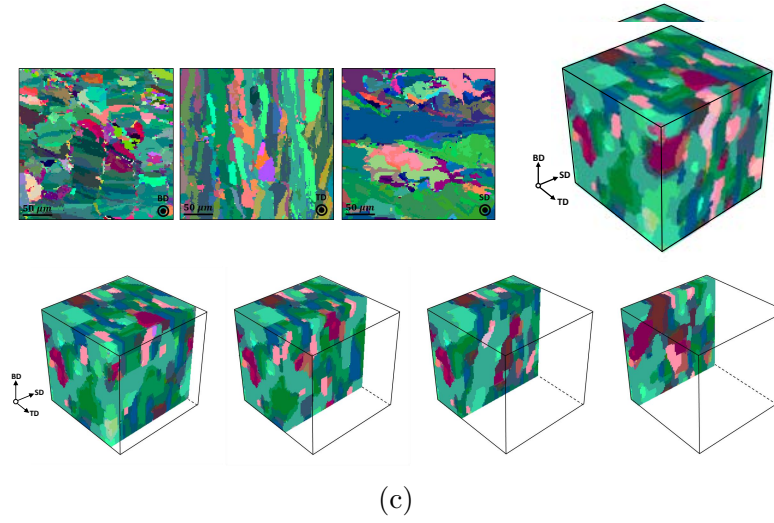


Figure 1.7: *Illustration of 2D/3D images: (a) set of orthogonal experimental images of AM 316 L stainless steel microstructure, (b) 3D MRF reconstructed model, and (c) sectional images of the reconstructed model.*

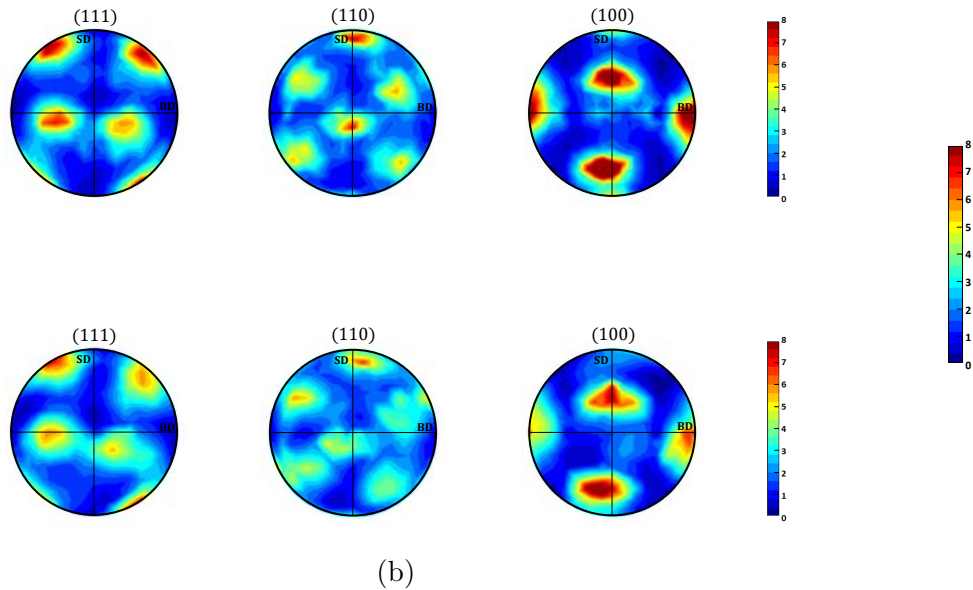


Figure 1.8: *Texture comparison of (1, 1, 1), (1, 1, 0), and (1, 0, 0) pole figures for: (a) the orthogonal planar images against (b) the synthesized 3D microstructure.*

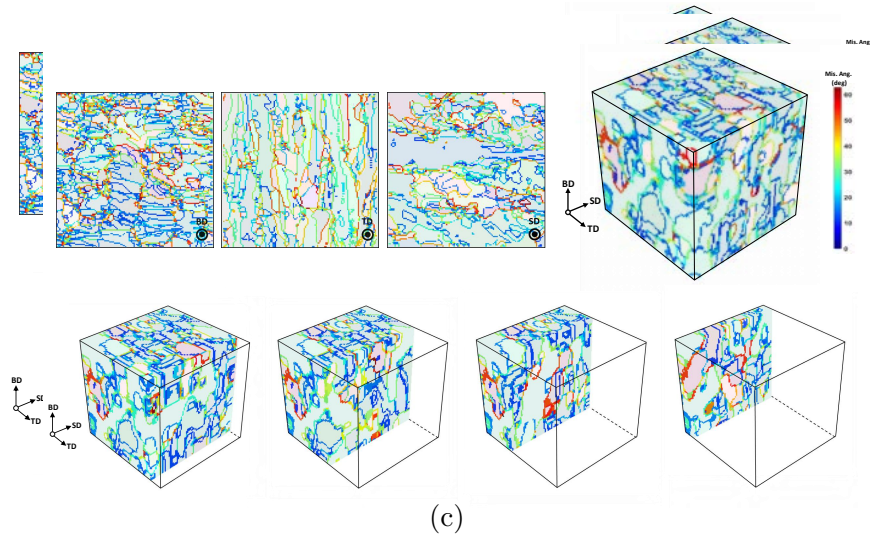


Figure 1.9: *Illustration of 2D/3D spatial distribution of grain boundary misorientation angles for: (a) set of orthogonal experimental images of AM microstructure, (b) 3D MRF reconstructed model, and (c) sectional images of reconstructed model.*

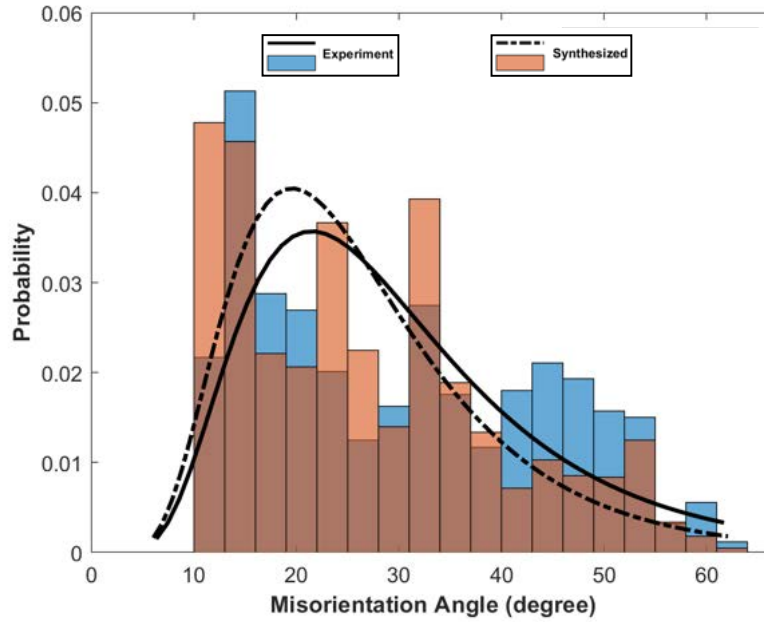
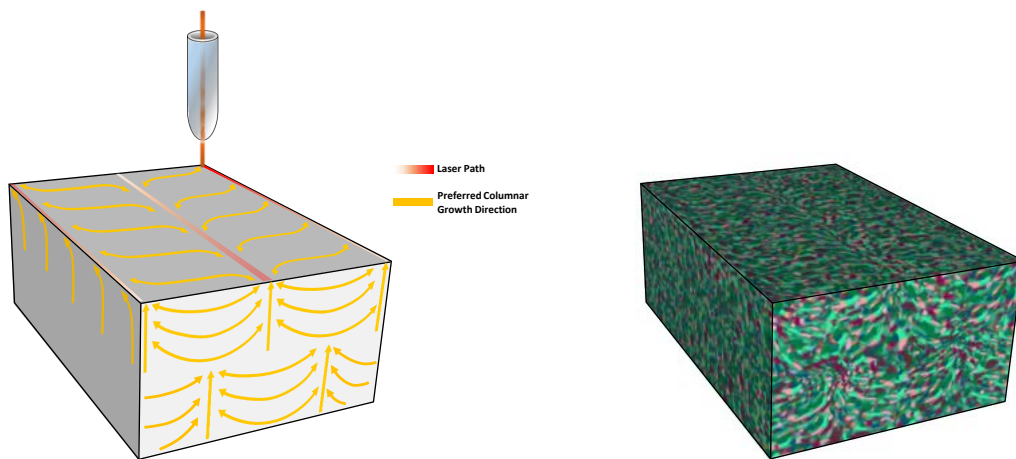
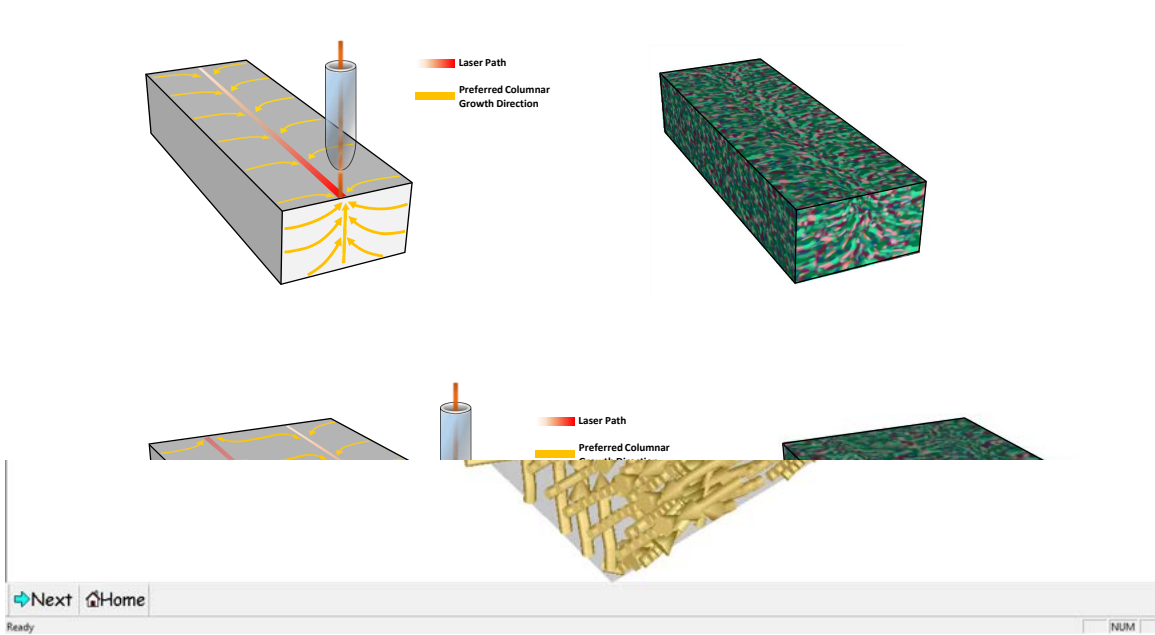


Figure 1.10: *Comparison of Probability Distribution Function (PDF) of grain boundary misorientation angles of the 3D synthesized microstructure against experimental imaging. Here, the probability densities of the raw misorientation angles are computed over 18 bins using a log-normal fit. The maximum unique misorientation angle possible due to cubic symmetry is 62.8° .*



(b)

Figure 1.11: Illustration of LEGOMAT embedding for simulating microstructure (right) for (a) two laser scan lines across a single layer and (b) two layers with different hatch spacing. The laser paths and tensor fields for each case are shown in the left.

CHAPTER II

LEGOMAT: Integration of MRF models for large scale microstructure synthesis

A new numerical method is presented for reconstructing three-dimensional (3D) microstructures from two-dimensional (2D) sections, imaged on orthogonal planes, by exploiting the complete red-green-blue (RGB) color space. The algorithm reconstructs 3D models through sampling voxel neighborhoods to representative 2D micrographs, based upon a Markovian assumption. The sampling is followed by an optimization procedure, ensuring smoothness across the orthogonal sections of the synthesized voxels. Previous 3D Markov random field (MRF) microstructure reconstruction techniques were restricted to traditional grayscale images only. This method now enables the use of the entire RGB spectrum, employing a histogram matching step. This chapter examines the algorithm's accurate representation of orientations and morphologies, encompassing a variety of micrographs from electron backscatter diffraction (EBSD) and polarized light microscopy.

2.1 Introduction

Many aspects of microstructures (e.g., size, shape, and connectivity of phases in a multiphase material) are not readily apparent from a two-dimensional (2D) sectional

measurement [14]. Yet, the majority of conventional microstructural characterization procedures are limited to only 2D observations [49, 50]. Three-dimensional (3D) microstructural information is essential to computational techniques, used for understanding structure–property relationships. As a result, advances in destructive metallographic methods have gained considerable attention for a 3D model generation. The serial sectioning method and its variants are now commonly used to derive 3D microstructural information from 2D sectional images [51, 52, 53]. In serial sectioning, layers of controlled thickness are repeatedly removed. This allows for image acquisition of each section, followed by reconstruction of the layers into a 3D representation of the microstructure. The 3D data collected via serial sectioning can then be used as the basis for computer simulations, such as crystal plasticity modeling of mechanical deformation [54, 55]. The drawback of these characterization techniques is that they are fairly laborious for routine engineering applications, due to the time and the effort involved. In response to these limitations several non-destructive experimental techniques for 3D microstructural information acquisition have recently been realized. X-ray based computed tomography (CT) methods are generally deficient in achieving the desired resolution for detailed analyses of many phenomena in metallic materials [56, 57, 58]. High-energy diffraction microscopy (HEDM) techniques can probe the microstructures at or below the grain-scale more accurately [59, 60, 61, 62]. However, a lack of general availability for high-energy beam sources restricts their common application in computational modeling development. For these reasons, new numerical techniques that can rapidly synthesize 3D microstructural models have gained significant interests in the materials community.

Advances in microscopy have allowed for data collection on the surfaces of a specimen at a rate of several hundred points per second [63]. As a result, the direct problem of capturing 2D sections using optical or micro-diffraction techniques is considerably more straightforward, compared to the aforementioned 3D acquisition

procedures. Current numerical techniques for computer-aided microstructure reconstruction methods, such as geometry-based (e.g., Voronoi tessellations) or feature-based (e.g., simulated annealing), often use 2D sectional information to generate 3D microstructures. Feature-based reconstruction algorithms have long been used to categorize various microstructural snapshots based on common sets of underlying features [64, 65, 66, 67, 68, 69, 70, 71, 72, 73]. Some of these features include marginal histograms [64], multi-resolution filter outputs (e.g., Gaussian and wavelet filters) [65, 66, 74], and point probability functions (e.g., autocorrelation function) [67, 68, 69, 70, 71, 72]. Although these methods are efficient at modeling the global features of the microstructure, they usually fail at capturing the local information in the form of per-pixel data, when reconstructing polycrystalline structures [66]. A popular geometry-based tool for 3D reconstructions of microstructures is DREAM.3D [75, 76, 77]. DREAM.3D extracts statistical information from images, such as grain morphology as well as orientational features. From such information it generates a full 3D geometrical model, capturing the overall crystalline structure. The super-ellipsoidal grain approach associated with DREAM.3D reconstruction tends to idealize the crystal morphology [78]. Hence, the above-mentioned numerical methods run into difficulties when modeling complexities of microstructures such as non-equilibrium grain structures, non-convex grains, twins, second phases, precipitates, and cell structures [79, 80]. Such features naturally arise from materials processing, playing an important role in their properties and performance. Accordingly, the need to develop robust numerical algorithms for 3D microstructure reconstruction is critical. Such algorithms can provide insight into the microstructure-property relationship of materials; ultimately advancing the development of smarter, multifunctional, and more sustainable structural systems.

It is experimentally observed that sufficiently large slices along a particular direction taken from a 3D polycrystalline microstructure generally ‘look alike’ [73, 81, 82, 83,

84]. This gives evidence to the notion of stationary probability distribution, which underlies microstructure formations [35, 85]. Consequently, the inverse problem of synthesizing 3D polycrystalline structures from 2D sectional images taken along the orthogonal directions is of specific interest to this chapter. Here, the 2D input sections are in the form of pixels that contain red–green–blue (RGB) color values representing different constituent phases or orientations. The outcome of the inverse problem is a 3D solid microstructural model made of voxels (i.e., volumetric pixels) that are colored consistently with the 2D orthogonal input images, such that any arbitrary x , y , or z slice of the 3D model is ‘similar’ to its corresponding 2D input micrograph. The measurement of similarities between the slices of the synthesized 3D microstructure and the given 2D sectional images is based on a least–square distance (i.e., Euclidean–norm) cost function. Minimizing the cost function ensures that the local neighborhood of any voxel on a 2D slice taken along the x , y , or z direction through the synthesized 3D microstructure is representative of some neighborhood in the given 2D micrograph, taken along the same direction. This approach is consistent with the Markov random field (MRF) model of texture synthesis [86, 87, 88] that recently has been successfully applied for microstructure reconstruction [85, 89, 90, 91, 92]. The underlying assumption states that the probability density of each pixel within the microstructure can be fully defined, based on a limited number of its surrounding pixels, not necessarily all the pixels in the micrograph [35, 85]. To estimate an unknown voxel in the 3D synthesized model, small windows in the 2D micrographs are first identified as being the best–match to the unknown voxel’s neighborhood. Once identified, the matching windows are sampled. The state of the unknown voxel is then determined by the coloring of the center pixels in the best–matching windows. Previous works in 3D microstructure reconstruction using MRF have shown that such an approach can effectively capture microstructural features [35]. However, the results were restricted to grayscale coloring only. This work has laid the foundation for

the use of the entire red–green–blue (RGB) color spectrum, through the exploitation of an image histogram matching step (also known as histogram specification). In the image processing community, this step is often performed to achieve consistent intensity (e.g., illumination or contrast) across two datasets (i.e., the reference and the target images) [93, 94]. In this chapter, the efficacy of this new approach to model 3D features in the polycrystalline structure using orthogonal 2D image datasets is examined. Using the MRF algorithm, it is shown that both global features such as grain size, texture distribution, and grain neighborhood correlations as well as localized information such as precipitates and grain boundary networks closely resemble those of the experimental images.

2.2 Methodology

The objective here is to first summarize the sampling methodology, used in Reference [35]. The histogram matching extension, which provides the ability to reconstruct color images, is described next. Following this, the computational costs associated with MRF modeling and its relation with sampling window size is discussed in detail.

2.2.1 Sampling Algorithm

In the following discussion, let $\mathbf{S}^x$, $\mathbf{S}^y$, and $\mathbf{S}^z$ denote the set of orthogonal slices of the microstructure along their respective x , y , and z directions. The symbol $\mathbf{V}$ indicates the synthesized 3D microstructure. Whereas, $\mathbf{V}_v$ represents the color of voxel v in the solid microstructure. In the previous work, G distinct color levels in the range $\{0, 1, \dots, G - 1\}$ were used to represent the microstructural data. In the current algorithm, three layers of color in the red, green, and blue channels (as a triplet) are allowed for representation of the phase information. The exploitation of RGB scheme allows for significant flexibility in the choice of the microstructural colormap, enabling direct visualization of colors throughout the reconstruction.

The following algorithm presented here is the same as in Reference [35], except that the averaging step is performed on each color channel separately. For pedagogy, let us consider how one of the color channels in the 3D microstructure is reconstructed. The hypothesis in the sampling approach is that the color assigned to any voxel follows a Markovian property. That is, the probability density function (PDF) of a voxel given the states of its spatial neighborhood is independent of the entire dataset. Thus, here the neighbors are chosen over relatively small user-assigned windows around the voxel v . The vectors denoting the spatial neighborhood of voxel v in the slices orthogonal to the x , y , and z axes are denoted as $\mathbf{V}_v^x$, $\mathbf{V}_v^y$, and $\mathbf{V}_v^z$, respectively (see Figure 6.1(a)). Let $\mathbf{S}^{x,w}$, $\mathbf{S}^{y,w}$, and $\mathbf{S}^{z,w}$ denote a window of the same size in the input 2D micrographs. In order to find the coloring of an unknown voxel v given the neighbor voxels in the x -plane, one needs to compute the conditional probability density of $\mathbf{V}_v$ given colorings of x -plane neighbors of v . Explicit construction of such a probability density is often computationally intractable. An alternative approach is to estimate the most likely RGB value of the voxel v by identifying a window $\mathbf{S}^{x,w}$ that is most similar to $\mathbf{V}_v^x$ in the 2D input micrograph. This window is denoted by $\mathbf{S}_v^x$ (see Figure 6.1(b)). Similarly, matching windows to the y and the z plane neighborhoods of voxel v are found in the corresponding 2D sectional images (denoted as $\mathbf{S}_v^y$ and $\mathbf{S}_v^z$, respectively). Each of these matching windows $\mathbf{S}_v^x$, $\mathbf{S}_v^y$, and $\mathbf{S}_v^z$ may have different RGB coloring for the center pixel. Thus, an optimization methodology is employed to effectively merge these disparate values and to identify a unique coloring for the voxel v . The optimization approach is described next.

Let the value $\mathbf{V}_{v,u}^x$ denote the RGB coloring of voxel u in the neighborhood $\mathbf{V}_v^x$. Let the values $\mathbf{S}_{v,u}^x$ and $\mathbf{S}_u^{x,w}$, respectively denote the RGB color of pixel u in the windows $\mathbf{S}_v^x$ and $\mathbf{S}^{x,w}$. Consequently, the 3D microstructure can be synthesized by posing the problem as a Euclidean distance (L^2) minimization of the energy function (also known as the cost function), as formulated below:

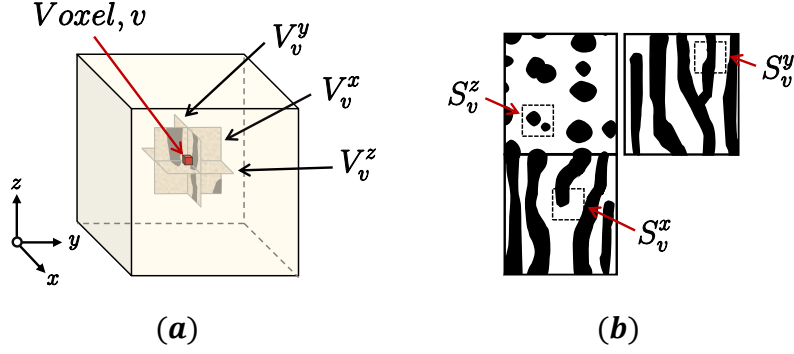


Figure 2.1: (a) The neighborhoods of the voxel v in the slices orthogonal to the x , y , and z axes are shown. (b) The windows in the input 2D micrograph shown in dotted lines are denoted by $\mathbf{S}_v^i$ for $i = x, y$, and z . These windows closely resemble the neighborhoods of the voxel, v .

$$E(\mathbf{V}) = \sum_{i \in \{x, y, z\}} \sum_v \sum_u \omega_{v,u}^i \|\mathbf{V}_{v,u}^i - \mathbf{S}_{v,u}^i\|^2 \quad (2.1)$$

Here, $\omega_{v,u}^i$ denotes a per-pixel weighting factor. In order to preserve the short-range correlations of the microstructure as much as possible, a Gaussian weighting is used such that the weights for nearby pixels are taken to be higher than pixels farther away. For instance, if the pixel u along the x -plane is placed at a relative location (y_u, z_u) (in lattice units) with respect to the voxel v , then $\omega_{v,u}^x$ is given as:

$$\omega_{v,u}^x = \frac{\exp\left(-\frac{y_u^2 + z_u^2}{2\sigma^2}\right)}{\sum_j \exp\left(-\frac{y_j^2 + z_j^2}{2\sigma^2}\right)} \quad (2.2)$$

The summation in the denominator is taken over all the pixels in $\mathbf{S}^{x,w}$. Here, the standard deviation σ , is taken to be $0.16 \times w$, where w refers to the window size [89]. Similarly, $\omega_{v,u}^y$ and $\omega_{v,u}^z$, respectively denoting per-pixel weighting factors in $\mathbf{S}^{y,w}$ and $\mathbf{S}^{z,w}$, can be computed.

The optimization of the energy function $E(\mathbf{V})$, is carried out in two steps: (i) the

searching step, and (ii) the expectation step. In the first step, the energy function is minimized with respect to $\mathbf{S}_v^i$. Here, the assumption is that the most likely sample from the conditional probability distribution of the center voxel in the 3D model along a particular direction is the center pixel of the best-matching window in an experimentally-obtained 2D slice on the corresponding plane. Hence, the best-matching neighborhood of voxel v along the x -plane is selected by solving the following minimization problem:

$$\mathbf{S}_v^x = \arg \min_{\mathbf{S}^{x,w}} \sum_u \omega_{v,u}^x \|\mathbf{V}_{v,u}^x - \mathbf{S}_u^{x,w}\|^2 \quad (2.3)$$

This is an exhaustive search that compares all the windows in the input 2D micrograph to the corresponding x -slice neighborhood of voxel v and identifies a window that leads to a minimum weighted squared distance.

Thus, for each voxel v , a set of three best-matching neighborhoods within the experimental images are identified. Generally the center pixel values in each of these neighborhoods are composed of different RGB colorings. Yet, a unique value of v needs to be found by weighting colors pertaining to location v in not only the matching windows of voxel v but also its surroundings. This is done in the second step of the optimization procedure. Here, the optimal color of voxel v is computed by setting the derivative of the energy function to zero, with respect to $\mathbf{V}_v$. This leads to the following weighted-average expression for the color of voxel v :

$$\mathbf{V}_v = \left(\sum_{i \in \{x,y,z\}} \sum_u \omega_{u,v}^i \mathbf{S}_{u,v}^i \right) / \left(\sum_{i \in \{x,y,z\}} \sum_u \omega_{u,v}^i \right) \quad (2.4)$$

Note that the subscripts u and v are switched in the above expression, compared to Equation 2.1. This implies that the optimal color of the voxel v is the weighted-average of the colors at locations corresponding to voxel v in the best-matching windows of voxels u (i.e. $\mathbf{S}_u^i$) found within the synthesized 3D solid microstructure.

The color channels are averaged independently in the equation above. Since $\mathbf{V}_v$ is continuously changing after each step, the set of closest input neighborhoods $\mathbf{S}_v^i$ varies accordingly. Hence, the above two steps are repeated until convergence; that is until the set $\mathbf{S}_v^i$ remains unchanged.

2.2.2 Histogram Matching Algorithm

The primary issue with the above equations is the assumption that the color space is continuous. Such an assumption allows for the partial derivatives of the energy function to be obtained. However, the color space is typically discrete and range bound. Consequently, the averaging performed in Equation 2.4, always tends to shrink the color levels. For example the color level 0 always tends to increase, since it is repeatedly averaged with all the color levels that are greater or equal to 0. Similarly the maximum color level G of each RGB channel (typically equals to 255) has a tendency to decrease, due to it being averaged with other color levels that are always smaller or equal to G . However, the general assumption in MRF reconstruction remains, that the two datasets (i.e., input images and solid synthesized model) have the same range of color levels. To solve this issue, histogram matching is applied after each iteration, allowing for the color levels to be appropriately stretched back to values from 0 to G at each channel.

The stretching process is based on the cumulative distribution function (CDF) of colors in the input micrographs. That means, the 2D micrographs are considered as the reference images, whereas the 3D solid model's color histogram is to be matched following each iteration. Here, all three orthogonal images are lumped together in order to generate the reference cumulative probability distribution. In current implementation, the histogram matching algorithm is applied separately, for each color channel. Given the reference (2D) and the target (3D) images, the color histograms are obtained by binning the pixels into G discrete intervals. Subsequently, CDFs of

the two color histograms H_{2D} and H_{3D} are computed and normalized. For each voxel (with RGB color value $\mathbf{V}_v$) in the 3D synthesized model, the color level X in the 2D image is found such that $H_{2D}(X) = H_{3D}(\mathbf{V}_v)$. Accordingly, the color level associated

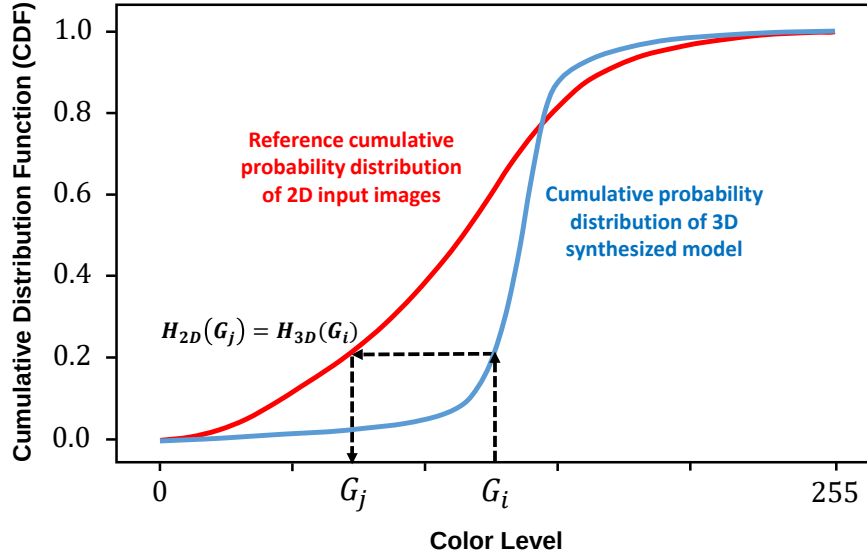


Figure 2.2: *Histogram matching algorithm: the color G_i in the synthesized 3D image is switched for each channel independently. The new color G_j has the same cumulative distribution function as in the 2D images.*

The application of histogram matching using an RGB image is illustrated in Figure 6.3. The first reconstruction (shown on the top of Figure 6.3) is based on the algorithm presented in Reference [35]. Note, that phase information (e.g., grain size, shape, orientation, boundary networks, etc.) is lost within the first few iterations. Though with the current implementation (using histogram matching) the volume fraction of each color level is preserved with the input images. The result of which is the reconstruction of a representative 3D microstructure. In Figure 6.3, the same 2D micrograph is considered as the reference image for both reconstruction schemes. The snapshots of the 3D solid models at iterations 0, 7, 21, and 35 are shown, respectively. Although the general approach presented in this work is not tied to a specific input

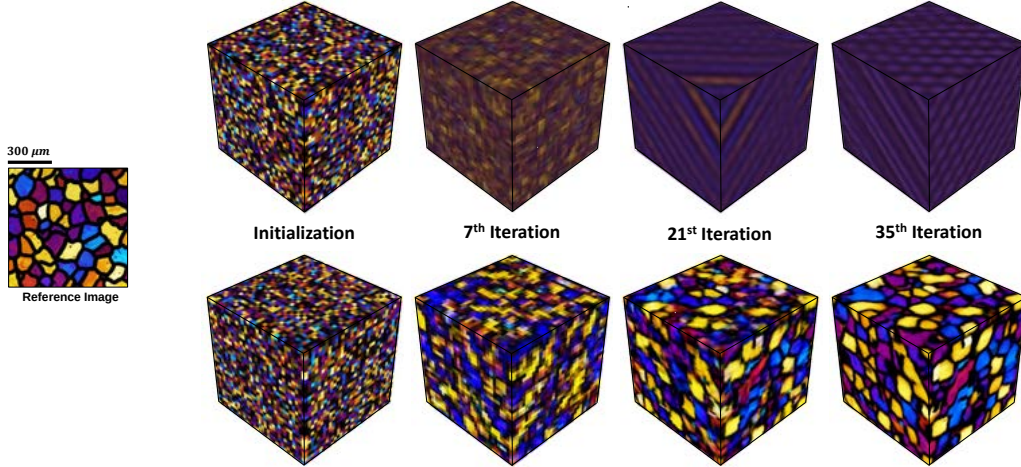


Figure 2.3: *Effect of histogram matching: without histogram matching (as illustrated on the first row), the color space continuously shrinks and phase information is lost after first few iterations; contrarily, with histogram matching (as shown on the bottom), the color space remains consistent with the input reference image throughout the simulation.*

resolution, all the input images are resampled to 128×128 pixels prior to MRF reconstruction. As a starting condition for 3D reconstruction, a random RGB color from the input 2D images is assigned to each voxel v . The reconstruction process is then carried out in a multi-resolution (or multi-grid) fashion [35]: starting with a coarse voxel mesh while progressively interpolating the results to a finer mesh once the coarser 3D image has converged to a local optimum. The multi-resolution approach drastically increases the convergence rate to an optimum cost function. As such, three resolution levels (32^3 , 64^3 , and 128^3) are used for the following three numerical examples. Respective window sizes of 7, 9, and 11 are chosen in the three resolution levels. The pseudocode in Algorithm 1 summarizes the sampling and reconstruction schemes:

Algorithm 1 Pseudocode for 3D MRF reconstruction of size N^3 .

Require: Orthogonal 2D images of size N^2 as well as the sampling window size for each level

Ensure: Optimum cost function given in Equation 2.1

Resize 2D images to L^2 , where $L = N/4$

Initialize a random 3D microstructure of size L^3 , where $L = N/4$

Initialize level index: $level \leftarrow 1$

while level < 4 **do**

while level unchanged **do** Use Equation 2.3 to identify the best matching windows in orthogonal 2D images (*Searching Step*)

 Use Equation 2.4 to compute the expected voxel channel values (*Expectation Step*)

for each channel **do** Compute CDFs of 2D images (H_{2D}) and 3D solid model (H_{3D})

 Find color level X in the 2D images such that $H_{2D}(X) = H_{3D}(\mathbf{V}_v)$

 Replace voxel channel color with color level X (*Histogram Matching Step*)

end for Update image size: $L \leftarrow 2L$

 Resize 3D synthesized model to L^3 and 2D images to L^2

 Update level index: $level \leftarrow level + 1$

2.2.3 Computational Cost vs. Window Size

Although there are several free parameters throughout the reconstruction strategy, window size is chosen to be the only adjustable variable for simplicity. Window size plays an important role in the MRF reconstruction model. At window sizes much smaller than the correlation lengths, false matches lead to inaccurate reconstructions.

Contrarily, at very large window sizes, not enough matching windows can be identified. Additionally, high window sizes require extensive computational power for both sampling and reconstruction. Between these two extremes, there is an ideal window size that needs to be realized through numerical trials. Normally, the window size should be larger than the average microstructural features, especially at low-resolution levels. This allows the algorithm to capture global attributes (e.g., grain boundary network, inter-connectivity, size distribution, etc.) in the polycrystalline microstructure at early iterations while reconstructing finer details (e.g., precipitate, twin, etc.) at later resolution levels. Figure 6.4 depicts the effect of window size (within the highest resolution level, 128^3) on the quality of synthesized 3D images. Note that only odd values are allowed for the window size, so that the sampled neighborhood is symmetric around the center voxel.

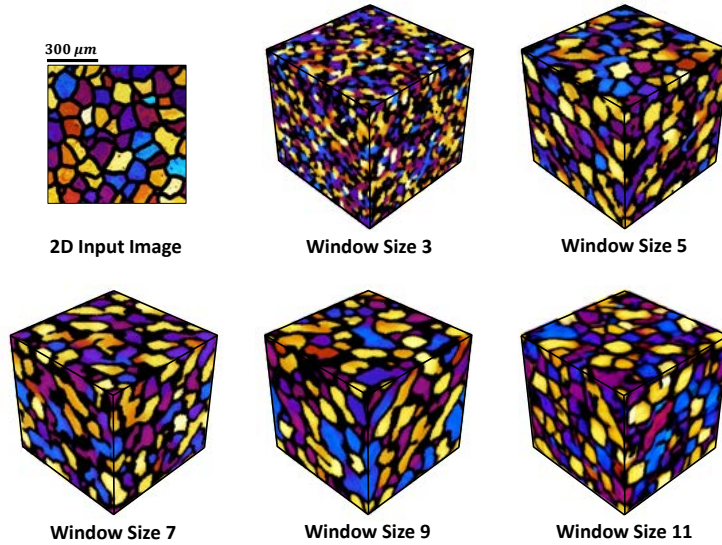


Figure 2.4: *Effect of window size: none of the sections in 3D images look identical to the sample image, but the image generated with window size of 11 has statistical correlation function similar to that of input image.*

The computational cost to obtain 3D solid models is directly related to the voxel resolution. In other words, higher voxel resolutions (128^3 vs. 64^3) require more computational power to perform searching and expectation steps. Table 6.1 provides a

Table 2.1: Breakdown of computational cost associated with highest resolution ($128 \times 128 \times 128$) level of the MRF reconstruction. For reference, all reconstructions are executed using a single processor and 4 GB of memory.

	Relative Time (%) Searching Step	Relative Time (%) Expectation Step	Average Time (min)
Window Size 3	65.05	34.95	1.69
Window Size 5	71.94	28.06	5.53
Window Size 7	74.31	25.69	11.09
Window Size 9	78.80	21.20	20.37
Window Size 11	81.31	18.69	32.25

breakdown of the computational cost as a function of the sampling window size associated with the highest resolution level. The processing times here correspond to the solid models shown in Figure 6.4. Note, simulation times generally vary drastically depending on iteration number as well as the features being sampled. As shown in Table 6.1, most of the computational burden is related to identifying the best-matching windows in the orthogonal input images (i.e., searching step). This is expected as the searching stage consists of an exhaustive approach that compares all the windows in the input micrographs, in order to identify a neighborhood that leads to a minimum weighted squared distance, formulated in Equation 2.3. Furthermore, note that the computational burden associated with performing the histogram matching step is typically related to the 3D voxel resolution, and it is independent of the sampling window size. Given this reason, histogram matching is excluded from Table 6.1. At the highest grid resolution, the typical time required to perform histogram matching is minimal (usually fewer than 5 seconds) when compared to expectation and searching steps.

In light of the above discussion regarding computational burden caused by different window sizes, it is expected that mapping sample images to lower resolutions would significantly improve the overall computational efficiency of the current algorithm. Note, the reconstruction scheme presented in this work is not tied to a specific input

resolution. Thus, any 2D image resolution can be reconstructed, if needed. The implication is that the user should specify a sufficient pixel resolution that would capture the phenomena of interest.

2.3 Results and Discussion

In the example below, the MRF algorithm is examined by comparing grain sizes for a previously studied sample in Reference [35] using the Saltykov method. Following this, the reconstruction procedure is applied to an orthogonal set of EBSD images and polarized light microscope data of an Aluminum alloy. Thereafter, the restrictions of current MRF microstructural modeling is highlighted.

2.3.1 Example 1: Validation of 3D Grain Size

Here, a polycrystal grayscale micrograph is chosen. The image is reconstructed in the RGB color space using the present algorithm. In this example, the ability of MRF to accurately model 3D grain size distribution as predicted by the stereological formula is studied. Such validation has not been done previously for 3D MRF algorithms. Grain size distribution is essential for simulating mechanical properties in polycrystals via Hall–Petch models [95]. The Saltykov approximation method provides a closed-form expression for the 3D grain sizes, given a 2D observation. The assumption is that the grains are approximately equiaxed, which is reasonable given that only one micrograph is used for all three orthogonal sections. The 3D reconstruction is shown in Figure 6.5.

The Saltykov formula is used to unfold the 2D apparent grain-size distribution into the actual 3D PDF ($F(R)$), as shown in Figure 6.6(a). The results are then compared against the 3D grain size distribution of the MRF reconstruction. The assumption in the Saltykov formula is that the reconstructed volume and the 2D images are of unit volume and unit area, respectively. The derivation is based on Figure 6.6(b),

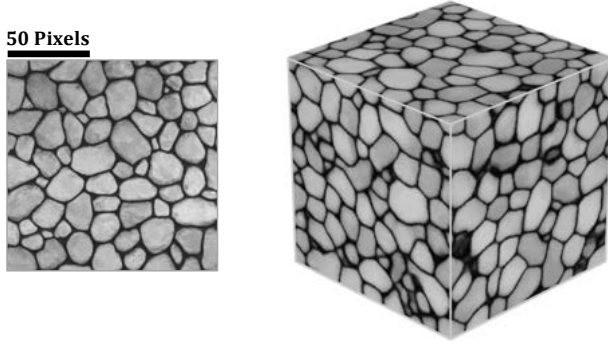


Figure 2.5: *An experimental 2D polycrystalline microstructure (as shown on the left) is used as an exemplar for 3D reconstruction. The 3D MRF reconstructed model is depicted on the right.*

where $z = \sqrt{R^2 - r^2}$ corresponds to a random cut location across a sphere of radius R with r being its apparent 2D grain size. Since the probability of z follows a uniform distribution (i.e., $P(z) = \frac{1}{R}$), the probability of r can be computed in a closed-form by the change of variables, as follows:

$$P_r(r) = P(z) \left| \frac{dz}{dr} \right| = \frac{r}{R\sqrt{R^2 - r^2}} \quad (2.5)$$

Integrating this expression leads to:

$$P(r_1 < r < r_2) = \int_{r_1}^{r_2} P_r(r) dr = \frac{1}{R} \left(\sqrt{R^2 - r_1^2} - \sqrt{R^2 - r_2^2} \right) \quad (2.6)$$

Where R corresponds to the 3D grain size, r_1 and r_2 are lower and upper limits of the interval for 2D grain size, and $P(r_1 < r < r_2)$ is the probability to cut the section within the defined interval.

The probability that a plane cuts a sphere of radius R is $2R$ (per unit length). The likelihood of cutting such spheres not only depends on their numbers but also on their sizes, since larger spheres are more likely to intersect a random section plane, as shown in Figure 6.6(c). Also, the likelihood of a sphere of radius R in the unit volume is $F(R)\Delta$. Here $\Delta = (R_{max} - R_{min})/k$ denotes the size of each bin, where k

equals the total number of bins. Hence, the probability a 2D section of radius r can be numerically written by taking a product of the equation above with $2RF(R)\Delta$ and by summing it over all bins (Saltykov formula):

$$P(r_1 < r < r_2) = 2\Delta \sum_i \left(\sqrt{R_i^2 - r_1^2} - \sqrt{R_i^2 - r_2^2} \right) F(R_i) \quad (2.7)$$

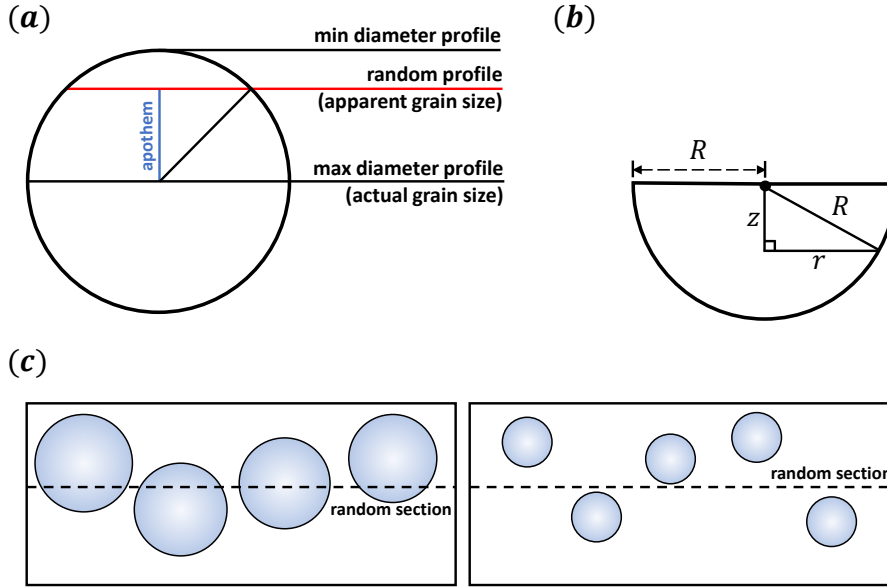


Figure 2.6: *Schematic explaining the Saltykov formula: (a) Grains are assumed as spheres with a random section seen in the 2D image. (b) Probability of a given 3D grain size R given a 2D section radius r can be obtained by a change of variables. (c) In the case where multiple 3D grain sizes exist, the Saltykov estimate takes into account the probability that a plane can cut a sphere of radius R . Hence, the spheres of larger radii are more likely to intersect the section plane.*

For the Saltykov formula, the same number of bins ($k = 24$) in the 2D and 3D case is used such that the above set of equations (one equation for each 2D bin) can be exactly inverted to obtain $F(R)$ [96]. In this example, the 2D grain statistics are captured in the following manner. The incomplete grains on the outer-edge adjacent to the borders of the image are removed. For every inner grain, the maximum possible diameter at all its boundary points is computed in pixel units. The PDF of the raw diameter data is then calculated over 24 bins using a log-normal fit. The result of

the Saltykov algorithm on this data is plotted as solid points in Figure 6.7. To obtain the 3D grain size distribution, edge detection is first performed on the solid MRF volume, followed by segmentation of grains. In the segmentation algorithm, grains are described by the number of connected faces (six-fold connectivity) of a similar color. To obtain the grain size distribution, the same process is followed as that used in the 2D image. That is the external grains are first eliminated, and for each inner grain, the maximum diameter for every boundary voxel is obtained. As illustrated in Figure 6.7, a close agreement of the Saltykov estimation and the sampled grain size distribution from the MRF reconstruction is observed. Deviations from Saltykov estimates are seen at the tails of the distribution, where the 3D reconstruction marginally predicts

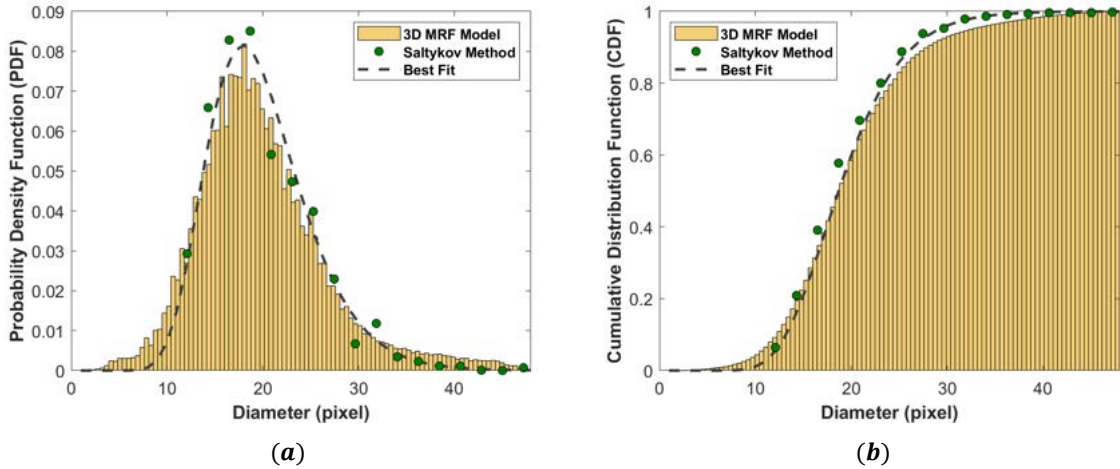


Figure 2.7: (a) Comparison of the grain size distribution (PDF) of the 3D microstructure along with the Saltykov prediction. (b) Comparison of cumulative distribution function (CDF) of 3D grain sizes with Saltykov prediction.

For the 3D model presented here, one reference image is supplied as an exemplar for all three orthogonal directions. Therefore, the model is assumed to be an equiaxed microstructure. This implies that slices taken from the diagonal directions should exhibit similar statistics as orthogonal planes in the solid 3D model. Hence, for comparison, the grain size distributions of these sections are studied. The process is

carried out for total of 15 orthogonal planes (five in each direction) and 18 oblique sections (three in every face-diagonal direction) across the 3D microstructure. A number of these sectional images are shown in Figure 6.8.

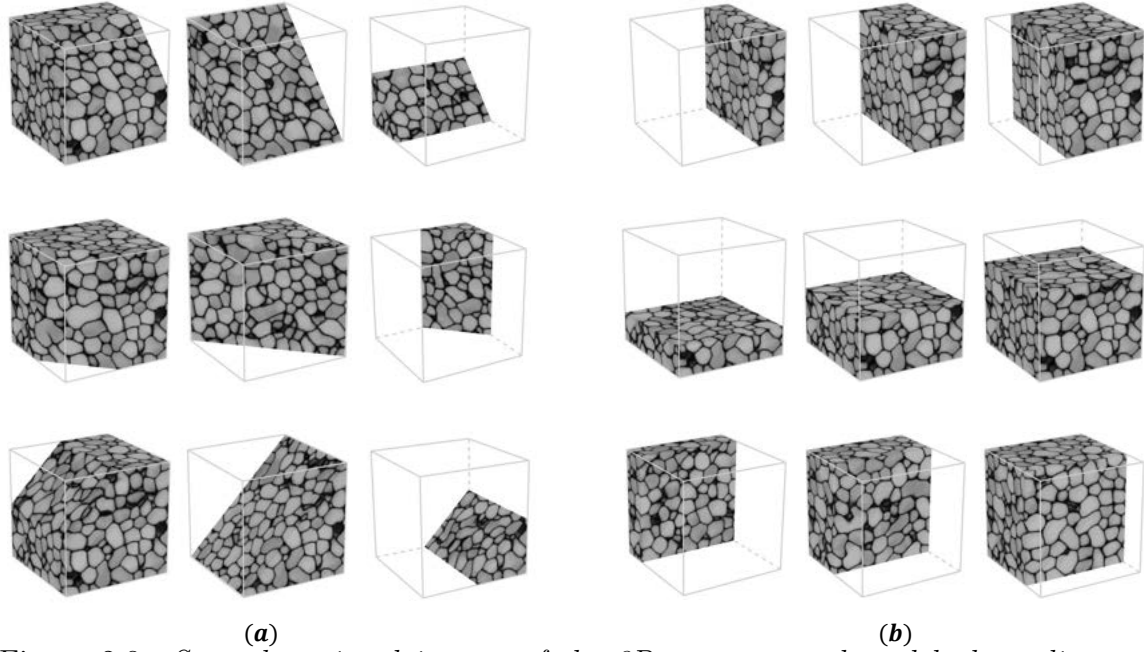


Figure 2.8: *Several sectional images of the 3D reconstructed model along diagonal (shown on the left) and orthogonal (shown on the right) directions are illustrated. The grain sizes and shapes along these sections nearly resemble the 2D reference image.*

To acquire the sectional grain statistics, same process as described for 2D input image, is employed. The partial grains adjacent to the borders are eliminated, and for each inner grain, the maximum diameter at every boundary node is calculated using pixel units. The probability density of diameters for both orthogonal and diagonal sections are then computed over 50 bins. As illustrated in Figure 6.9, a close trend between the two data is observed.

2.3.2 Example 2: 3D Reconstruction Based on EBSD Data

The following EBSD images, shown in Figure 6.11, correspond to an Al-Li alloy after forging to 2.5-inch thickness. Here, three orthogonal sections are characterized and used for MRF reconstruction. Experimentally, EBSD data is obtained using a

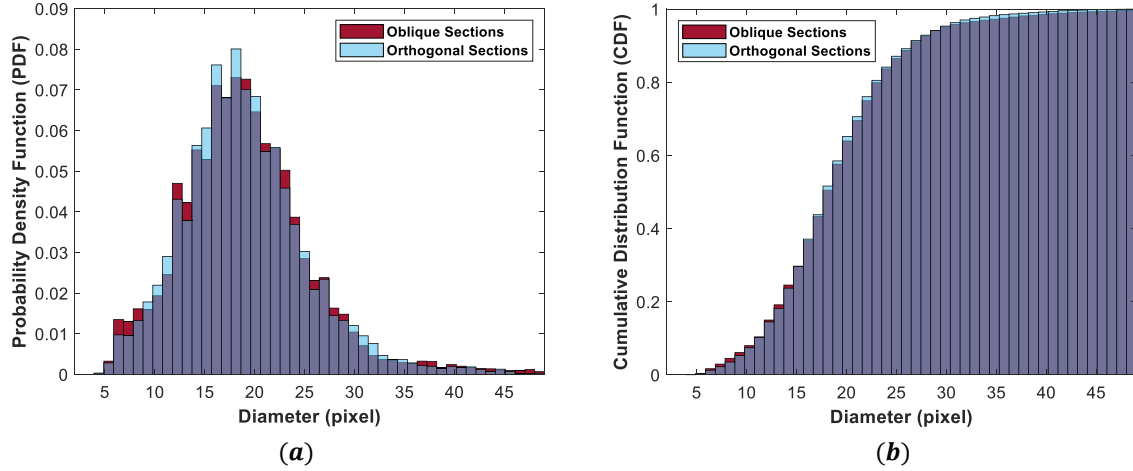


Figure 2.9: (a) Comparison of probability density distributions of the 3D grain sizes along oblique and orthogonal sections. (b) Comparison of cumulative distribution function of the 3D microstructure across oblique and orthogonal directions.

scanning electron microscope (SEM) equipped with detectors normal to each plane. To ensure a consistent MRF sampling across all EBSD sections, the orientations need

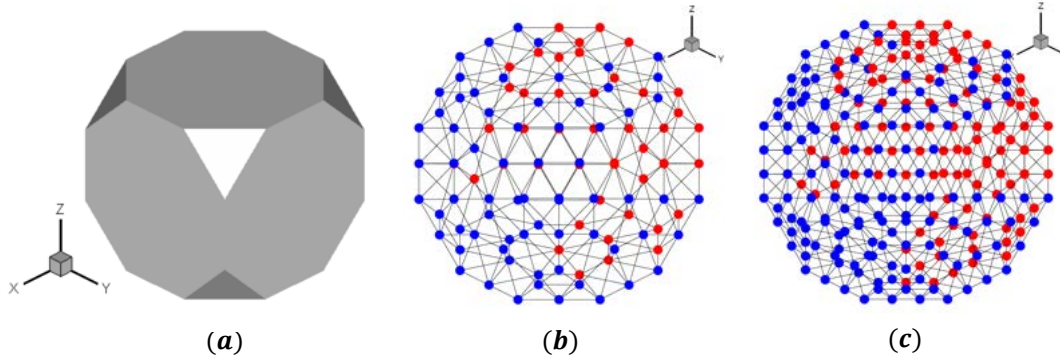


Figure 2.10: (a) Rodrigues fundamental region for cubic symmetry. (b) A coarse discretization with independent nodes colored blue. (c) A finer discretization.

Here, the raw Euler angle data is mapped to nodes in a discretized orientation distribution function (ODF). The ODF is a one-point probability function. It describes the volume fractions of crystals as a function of orientation. The orientation is represented here using an axis-angle format initially proposed by Rodrigues. The format

is based on the unique association of orientation with an axis of rotation, $\mathbf{n}$, and a counterclockwise rotation angle, θ about $\mathbf{n}$ [97, 98]. The Rodrigues' parametrization, $\mathbf{r}$, is scaled as $\mathbf{r} = \mathbf{n} \cdot \tan(\frac{\theta}{2})$. A proper rotation, $\mathbf{R}$, relates the sample orientation to the crystal orientation. Here, the example material (Al–Li alloy) lattice is of cubic symmetry. The 3D orientation space can be reduced to a small subset called the fundamental region that accounts for the cubic symmetry. The resulting Rodrigues fundamental region is shown in Figure 6.10(a). For numerical analysis, the fundamental region is discretized using a finite–element mesh. The symmetry of the ODF is an additional constraint that must be properly considered. Due to cubic symmetries, several of the nodes in the grid are equivalent. Hence, the ODF is represented using a smaller set of independent nodes, as shown in Figure 6.10(b), (c).

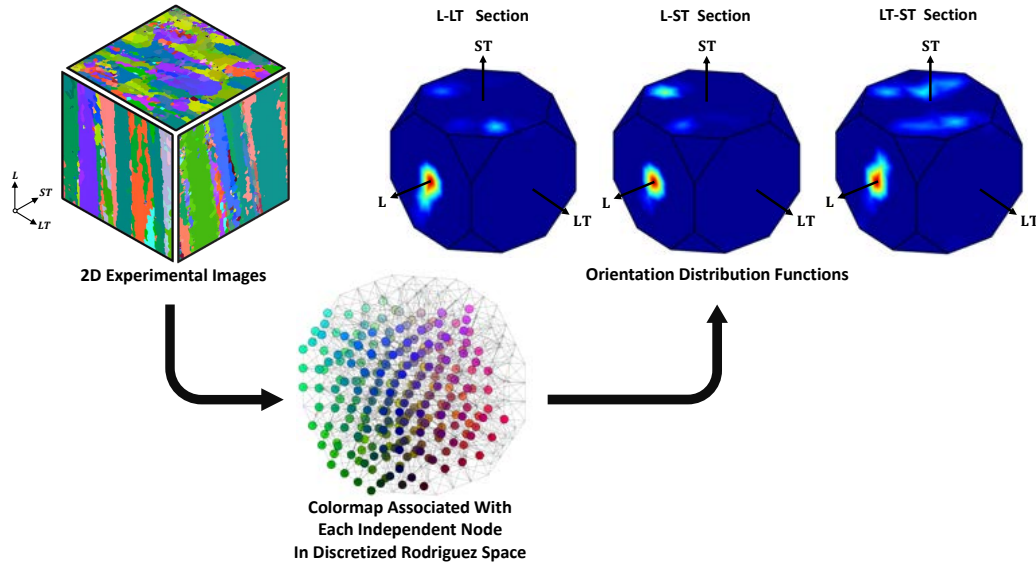


Figure 2.11: *The colormap shows the RGB values of individual nodes in the face-centered cubic (FCC) fundamental region. The pixels RGB values in the raw 2D orthogonal EBSD images are mapped to the closest node (in Euclidean norm sense) using the colormap. The orientation distribution functions of all three input images are similar, after adjusting the relative alignments of the detectors with respect to the sample.*

Each EBSD data point is mapped to the closest independent node in the ODF. If the orientation is close to a non-independent node, the node that is the symmetry

equivalent to the node is chosen. The 2D EBSD images are then colored according to the nodal numbers. A simple choice for the colormap is to apply the Rodrigues vector itself as an RGB triplet, after normalization of each Rodrigues vector component to a range of $[0, 255]$. Figure 6.11 shows the orthogonal set of images from the rolled sample; grains are elongated in the rolling direction and shortened in the transverse directions. The grains are colored based on a colormap in Figure 6.11, depicting the color of each independent node in the ODF. Here, three 700×700 EBSD sections are chosen and resampled to 128×128 images for the reconstruction. The ODFs for all three sections are comparable after correcting for the relative orientation of the camera with respect to the sample, as shown in Figure 6.11. The 3D reconstruction from the MRF algorithm is shown in Figure 6.13(b). A number of different sections of the reconstructed model are shown in Figure 6.13(c).

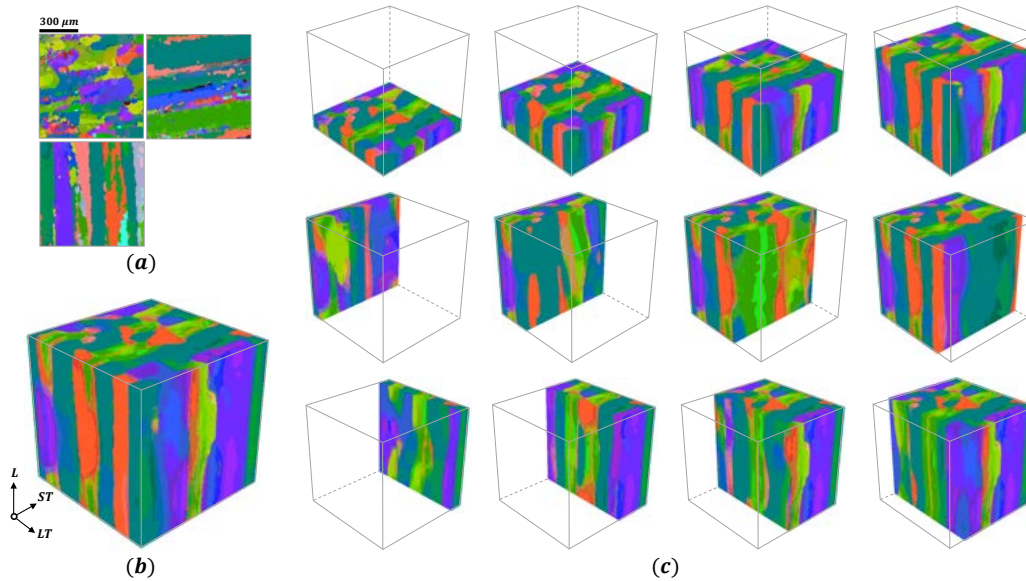


Figure 2.12: (a) *Experimental 2D polycrystalline microstructures imaged in orthogonal planes.* (b) *3D MRF reconstructed model.* (c) *Sectional images of the reconstructed microstructure.*

In this example, the averaging algorithm tends to smoothen the noise, and it supplies an overall smoother reconstruction compared to the input 2D images which consist of a significant amount of noise. Since the averaging step in the MRF algorithm introduces

deviation of colors from the original 2D colors, an additional step is employed to remap the colors to the independent nodes of the ODF through the colormap shown in Figure 6.11. The top four closest colors in the ODF that closely match the voxel color are found first. If it is a match within a threshold error, the voxel color remains unaltered. Otherwise, the closest colors (in Euclidean norm sense) to the colormap are identified. Of the colors identified as best matching, one randomly picked to recolor the voxel. It is found that the resulting ODF and pole figures of the MRF reconstructed microstructure matches exceptionally well to the measured ODF, as shown in Figure 6.14.

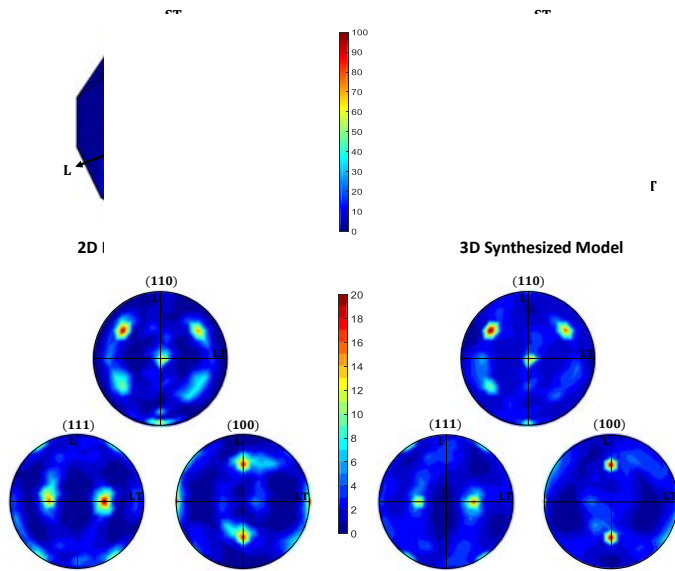


Figure 2.13: *Comparison of the ODFs and various pole figures of the 2D sample images against the synthesized 3D microstructure.*

In addition, a new analysis is performed, where the reconstruction algorithm is run sequentially, using random slices of previous 3D model as the input exemplars for the next synthesis. Such an analysis exemplifies the ability of 3D MRF algorithm to preserve texture statistics, generate distinct equivalent microstructures, and most importantly demonstrates how errors can be accumulated after every reconstruction. An example for sequential reconstruction is shown in Figure 6.15. These sequences reveal that our 3D reconstruction strategy succeeds in preserving the original exem-

plars' orientation distribution. Despite all, due to the averaging step, at some point deviations in statistics are expected to creep into reconstructions. This only happens after a significantly larger number of reconstructions.

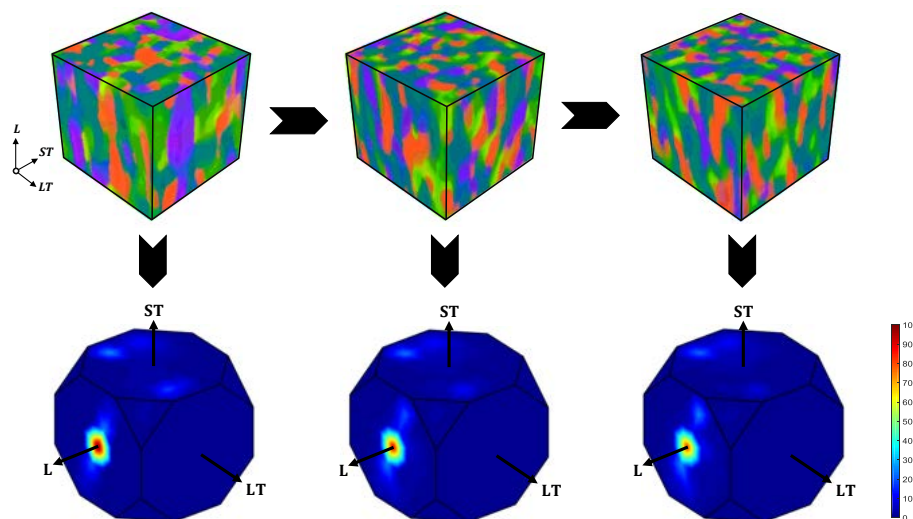


Figure 2.14: *Texture sequence analysis is shown for MRF algorithm. Each reconstruction uses the random slices of the previous 3D model (shown on its left) as its input exemplars. Here, the orientation distributions of each reconstruction exceptionally match with the original exemplars' ODF.*

2.3.3 Example 3: Reconstruction of Polarized Light Micrograph

The third example is an optical micrograph of an Aluminum alloy from Reference [99]. In this example, the microstructure is first etched with Barker's reagent. Next, an electric field is applied to produce a thick oxide layer over the Aluminum grains. When exposed to the cross-polarized light, interference in these oxide layers produces colors that depend on the grain orientation. For reconstruction purposes, the image is preprocessed such that all grain boundaries are accentuated to demarcate the grains clearly. The 2D microstructure is depicted in Figure 6.12(a). Only one sectional representation is taken, and the other sections are assumed to be identical (similar to Example 1). The microstructure is then reconstructed using MRF algorithm, and several sections of the 3D synthesized model in Figure 6.12(b) is indicated in Figure

6.12(c).

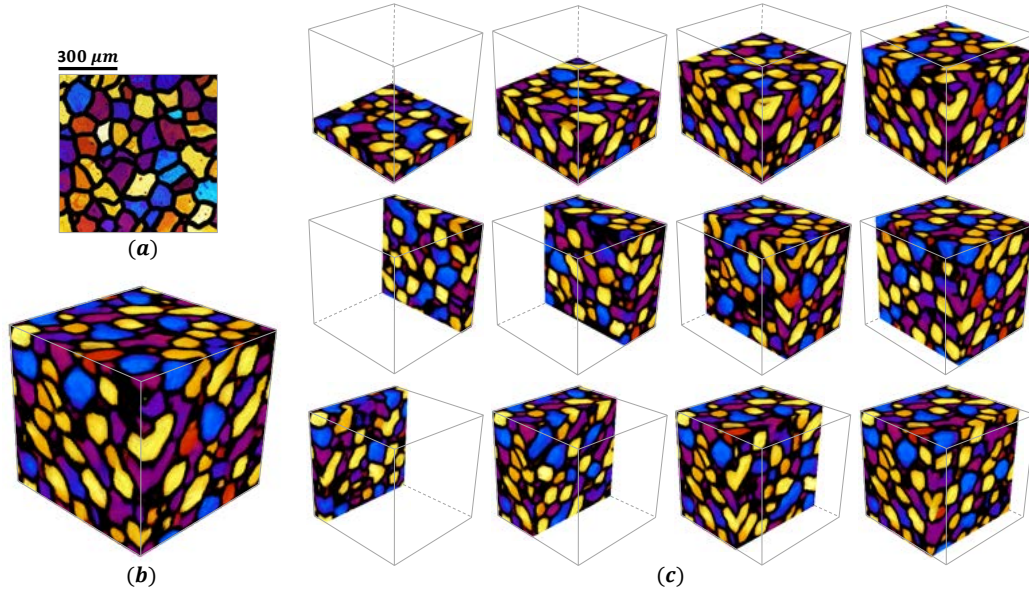


Figure 2.15: (a) *Experimental 2D polycrystalline microstructure.* (b) *3D MRF reconstructed model.* (c) *Sectional images of the reconstructed microstructure.*

The goal here is to examine the nearest neighbor–grain correlation, which is a critical descriptor relating mechanical properties [100]. For this purpose, the grains are classified in terms of four predominant colors observed in the image: purple (P), blue (B), orange (O), yellow (Y). The 2D and 3D microstructures with grains marked by this classification are shown in Figure 2.16 (left). For all yellow grains, the number of blue neighbors, as well as all the other neighbors (purple, orange, and yellow), are counted. The same process is repeated for the remaining colors. Similar data is gathered from five slices in each direction (total of 15 slices) across the 3D microstructure. The neighbor statistics of the colors are compared in Figure 2.16 (right). In this figure, the x -axis denotes the color of the chosen grain, while the color bars indicate the probability of occurrence of a yellow, orange, purple, or blue neighbor for that specific grain. There are two bars for each neighbor color, one for the 2D microstructure (located on the left hand side) and another for the 3D reconstructed model (located on the right hand side) as indicated by the legend. The range bar

on the tip of the 3D bars shows the maximum and minimum values of the neighbor color percentages along the x , y , and z planes. The numbers on the bars indicate the relative percentage difference between the 2D and 3D statistics. The maximum error between the 2D and 3D neighbor statistics is only 3.32% which is associated with having a purple grain neighboring another purple grain. This example indicates that the MRF algorithm is exceptional in maintaining the nearest grain neighborhoods in the reconstructed image, as is also observed in the two-point short-range correlation study in Reference [35].

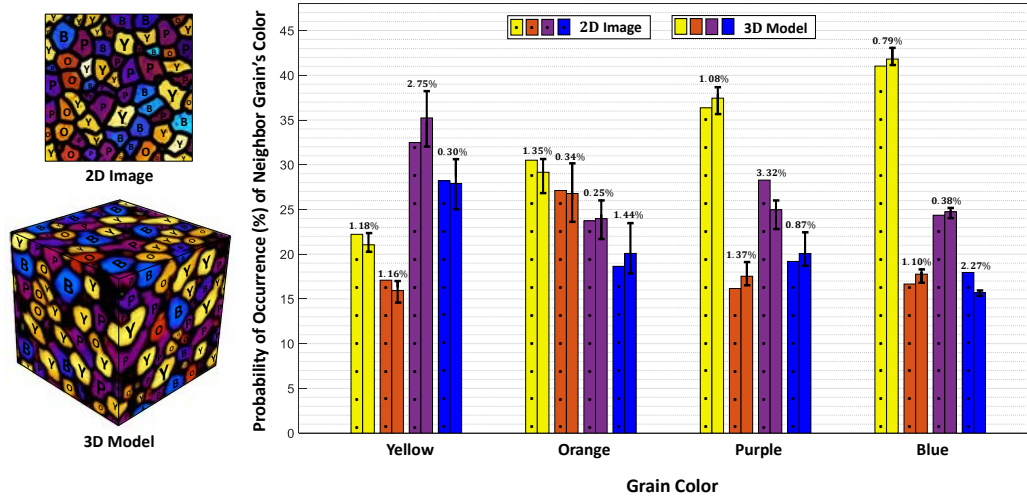


Figure 2.16: (left) Four predominant colors (purple (P), blue (B), orange (O), yellow (Y)) as marked on the 2D and 3D microstructures. (right) Neighbor statistics of the colors; the x -axis denotes the color of the chosen grain, and the bars indicate the probability of occurrence of colored neighbor grains. There are two bars for each neighbor color, one for the 2D image and another for the 3D reconstructed microstructure.

2.3.4 Example 4: Limitation of Markov Random Field Modeling

The presumption behind MRF reconstruction methodology is that given surface images are representative (i.e., having similar grain morphology and textural information) to every slice in their respective directions. This requires the user to carefully select volumetric regions in which one single image can adequately describe the entire dataset along that particular axis. As shown in previous examples, this approach

is efficient in capturing a wide variety of microstructural features. However, there are samples for which above requirement is insufficient. Figure 2.17 demonstrates a synthesized 3D microstructure with a non-uniform morphology. Here, grains are densely populated at $z = 128$ and scarcely generated on $z = 0$ slice. Due to the nature of the given microstructure, no single image can thoroughly capture the varying morphology along the z direction. As a result, to reconstruct an equivalent 3D model using MRF, one needs to take into account the statistics of additional orthogonal or oblique sections in their exemplars [92].

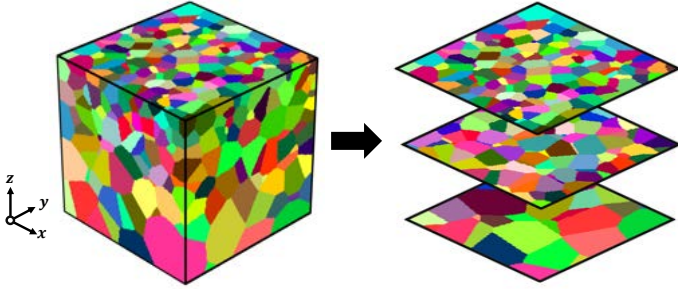


Figure 2.17: *This example is specifically selected to illustrate the shortcomings of the MRF algorithm, as presented in this chapter. The varying morphology of the given microstructure along the z direction, can not be captured by only one single image orthogonal to that axis. As a result, the MRF algorithm can not be used to properly reconstruct such a 3D model. Here, the slices on the right hand side correspond to $z = 0$, 64, and 128, respectively.*

2.4 Conclusions

A numerical method for reconstructing diverse 3D multi-color microstructures from 2D micrographs imaged on orthogonal planes is presented. The algorithm reconstructs 3D solid models by matching orthogonal sections of the synthesized microstructure to the representative 2D micrographs through an iterative minimization procedure. The principal contribution in this chapter is the extension of the algorithm to true color (RGB) images via implementing a histogram matching step after every iteration. Such an extension is essential for accurate representation of descriptors such as

orientation distribution functions. In this chapter, the new algorithm is applied for a wide variety of morphologies, particularly equiaxed and elongated grains. Through the reconstruction of optical and EBSD datasets, it is shown that the MRF approach is able to precisely capture crucial features of polycrystalline microstructure, such as grain size, orientation distribution, and grain neighborhood correlation. The method can be useful to rapidly create a library of 3D microstructures from 2D micrographs for modeling and simulation purposes.

CHAPTER III

Graph Coloring Method for Conformal Mesh Generation

This chapter provides a novel approach for mesh generation for materials which have distinct spatial components with a smooth boundary between them. Experimental data is used in pixel/voxel format to label elements in a generic finite element mesh of a representative volume element (RVE). The basis of this approach is a novel Potts energy formulation to allow integer optimization on the dual of the Finite Element (FE) mesh. The Potts energy can be decomposed into two terms: the field energy/data cost and the interaction energy/smoothing cost. The field term is used to represent the likelihood of a grain label on an element based on the experimental voxel data. The interaction term encodes a prior on this labeling, in particular, it is used for smoothening the phase boundary. Energy minimization of this system leads to a multi-way cut problem which is solved using Graph-cuts. This methodology allows capturing smooth boundaries in materials with non-equiaxed morphologies. Applications to polycrystalline microstructures and woven composites are presented. The extension to non-equiaxed morphologies is presented using the Riemannian distance measure. This procedure allows re-usability of a FE mesh by adaptively assigning pixel/voxel information to elements while preserving important features like the phase boundary surface length/area.

3.1 Background

Structural metallic materials used in the aerospace industry (eg. Aluminum 2000 and 7000 series, advanced titanium alloys) are composed of aggregates of crystals. With the emergence of the paradigm of integrated computational materials engineering (ICME) [2], multi-scale design/optimization approaches for tailoring engineering properties of materials through controlled microstructure [101, 102] are of great interest. A multiscale approach for the design of turbine blades is presented in Fig 3.1 to illustrate stress variation in the macroscopic and microscopic scale. Such a simulation involves solving for microstructure-dependent properties for macro-scale analysis, which in turn requires meshing at the microstructural level that captures the grain size and shape features [103]. Meshing of 3D microstructures to conserve such grain features is of immense interest as these play an important role in processes such as localization [104] and fracture [105].

Computational tools, like finite element methods, are almost ubiquitously present in the form of both commercial and private software. In the realm of polycrystalline materials, numerical procedures generally require the knowledge of the microstructure which is often provided through experimental methods such as Electron Back Scatter Diffraction (EBSD). EBSD provides the spatial distribution of crystallographic orientations of different grains and can be used to estimate quantities like average misorientation. A complete review of experimental and computational EBSD techniques are provided in [106]. From the computational perspective, these experimental images offer information about the spatial distribution of phases in a pixelated format for 2D or voxelated format for 3D. This data also provides a convenient form of a mesh which consists of uniform quadrilateral/hexahedral elements. An open source package is developed in [77] for generation of voxelated representations of serial sectioned EBSD maps. However, voxelation leads to a stepped, block-like representation of otherwise smooth boundaries. This affects the quality of computational simulation

through the introduction of spurious stresses [107]. There are other drawbacks in using structured meshes of quadrilateral/hexahedral elements for 2D/3D elements as well, for example, the boundary length/area does not converge to the correct value with refinement.

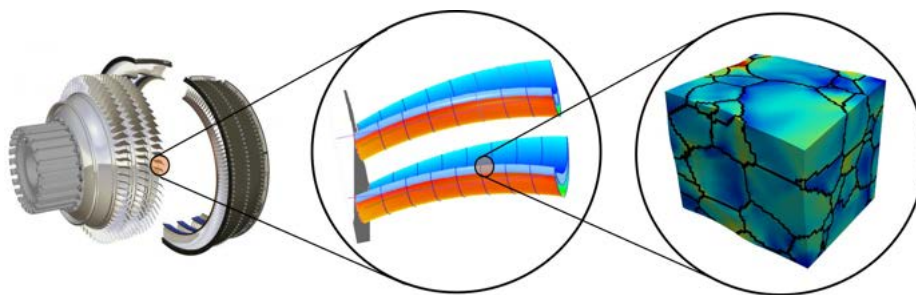


Figure 3.1: Multiscale design of turbine blades for aircraft engines.

Unstructured grids where element surfaces conform with the phase boundaries offer a better approximation of the spatial morphology. An unstructured meshing scheme which is widely employed, generates objects in the lowest dimension and then reuses these objects as seeds for generating higher order objects. More precisely, nodes (0-dimensional objects) are generated and are connected using lines (1-dimensional objects) where more points are sampled. Faces (2-dimensional objects) are created by joining these lines and are meshed using triangles with edge points as seeds. Then, these faces are connected to create a grain structure (3-dimensional objects) and are meshed with tetrahedrons using face triangles as seeds. This procedure allows for a smoother phase boundary and improves the estimation of quantities like Phase Boundary Energy. This procedure is also implemented for mesh generation in polycrystalline materials in an open source software package developed in [108]. However, this procedure can be computationally expensive as it requires re-meshing for every new experimental image.

A new technique for optimal partitioning of an arbitrary unstructured mesh into different phases is proposed in this chapter. In practice, partitioning the mesh means assigning an index to each element where elements with similar indexes form a single

phase. Thus, any experimental image can be superimposed on a generic unstructured FE mesh by effectively partitioning the mesh based on voxelated data. Two conditions are expected to be satisfied. Firstly, the phase index assignment to the finite element mesh should be as close as possible to the experimental voxel data. Secondly, the phase index assignments should be such that the boundaries are as smooth as possible. Seminal contribution in the development of the Graph partitioning algorithm are made in [109], [110] and [111]. Spectral methods (for instance, see [112]) are often used. Recent advances in this field have been reviewed in [113]. Improved results are achieved with global methods, such as multilevel approaches which operate in stages with refinement. METIS and KaHIP are good open source graph partitioners which utilize multi-level methods and are developed in [114] and [115], respectively. Since GP is often studied as a combinatorial optimization problem, the methods introduced in Chapter VI are also widely employed.

The overall objective of this chapter is to provide an effective way of labeling the elements of a mesh into the representative phases such that the spatial distribution of the phase is comparable to the experimental data while maintaining the smoothness in the boundaries. This is done by formulating a graph coloring problem on a graph embedding on the dual of an FE mesh. A Potts energy is formulated where the minimization of data term preserves the spatial phase data and the smooth term penalizes the rough behavior of phase boundaries. The resultant energy is solved using Graph-Cut algorithm. The above steps formalize the theory of embedding pixelated/voxelated data into an user-generated unstructured grid. The generalization to non-equiaxed morphologies of component phases is presented using a Riemannian metric. Application of the method to fiber composites is also studied.

3.2 Problem Formulation

The formulation for both 2D and 3D mesh generation is presented in this section.

Definition III.1 (Finite Element (FE) Mesh). Given a closed bounded polyhedral domain $\Omega \in \mathbb{R}^3$, one can associate a finite partition $\mathcal{P}_h$ of Ω in polyhedrons P_i (indexed with i) such that

$$\Omega = \bigcup_{P_i \in \mathcal{P}_h} P_i$$

such that:

1. $\text{int}(P_i) \neq \emptyset \forall P_i \in \mathcal{P}_h$
2. $\text{int}(P_i) \cap \text{int}(P_j) = \emptyset \forall P_i, P_j \in \mathcal{P}_h \text{ s.t. } P_i \neq P_j$
3. if $F = P_i \cap P_j \neq \emptyset$ for some $P_i, P_j \in \mathcal{P}_h$ and $P_i \neq P_j$, then F is either a whole face (polygon), a whole edge (line segment) or a node (point) of the polygons P_i and P_j
4. $h = \max_{P_i \in \mathcal{P}} h_i$ where h_i denotes the longest euclidean distance between two points of P_i .

where $\text{int}(\cdot)$ denotes the interior. The partition $\mathcal{P}_h$ is called the *mesh* of Ω

Remarks: In the case of 2D polygonal domain, $\Omega \in \mathbb{R}^2$, this definition can be extended by considering the partitions of polygons, P_i and modifying condition (3) to consider only edges and corners.

The mesh for multiphase materials is defined by assigning a phase to each element of the partition. This assignment or labeling, as it is referred to in the rest of the chapter, is done using an undirected graph, $\mathcal{G}$. The graph, $\mathcal{G}$, is embedded in the mesh, $\mathcal{P}_h$, by placing a single vertex, v_i , in the interior of each P_i . Without loss of generality, the location of v_i can be determined as the centroid of the polygon P_i . It ensures that $v_i \in \text{int}(P_i)$ when P_i is convex (generally true for all elements of FE meshes). A connection is introduced between two vertices v_i and v_j if and only if P_i and P_j share a common face. In case of 2D meshes, this connection is introduced if and only if P_i and P_j share a common edge. Furthermore, each connection is

endowed with a weight determined by the area and the unit normal of the shared face. Consequently in 2D meshes, the weight is determined by the length and the unit normal of the shared edge. In particular, if $\vec{n}$ and Δs are the unit normal and the area (length) of the face (edge), respectively, then weight w_i of the connection e_i is given as $w_i = \gamma(\vec{n})\Delta s$. The function γ is restricted to be strictly positive and symmetric i.e. $\gamma > 0$ and $\gamma(\vec{n}) = \gamma(-\vec{n})$. This construction is illustrated in Fig 3.2 where a connection between 2 representative elements **A** and **B** is introduced with a weight determined by the unit normal $\vec{n}$ and length of the edge Δs .

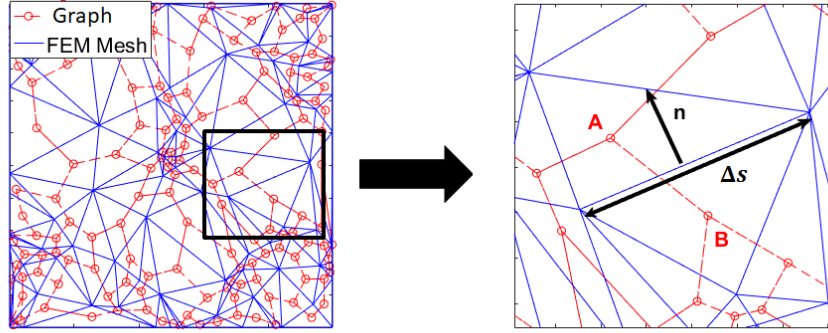


Figure 3.2: Construction of the graph $G=(V,E)$ from a finite element mesh.

In this work, labels ($\mathcal{L}$) are identified as different phases. For a particular choice of labels, and the corresponding multiway cut E' , a specific form of Potts energy is chosen:

$$E(\mathbf{S}) = \sum_{i=1}^{N_V} U_i(s_i) + \sum_{k=1}^{N_C} w_k \left(1 - \delta_{s_{\pi(k,1)}, s_{\pi(k,2)}} \right) \quad (3.1)$$

where $w_k \in \mathbb{R}^+$ and δ is the Kronecker delta function. The first term in this form is often referred to as the data term while the second term is referred to as the smooth term. The image information is encoded in the data term using a phenomenological form. Let I be a function which evaluates the label (phase information) from the image data (pixel/voxel). For instance, $I(v_i) = r$ means v_i belongs to the pixel region with phase index r . Additionally, let the function $Vol(P_i)$ evaluate the area/volume

of the i^{th} element in case of 2D/3D meshes with M representing the mode of the distribution. As an ansatz, U_i assumes the form of Eq (3.2) where $\alpha \in \mathbb{R}^+$ is a control parameter. It is worth observing that the minimization of this term leads to preservation of phase structure of the input data.

$$U_i(l) = \begin{cases} 0 & \text{if } I(v_i) = l \\ \alpha(1 - e^{-\frac{Vol(P_i)}{M}}) & \text{if } I(v_i) \neq l \end{cases} \quad (3.2)$$

The smooth term, as the name suggests, penalizes any roughness of phase boundaries. In this work, it is specialized to the form presented in Eq (3.3) where $\vec{n}_k$ and Δs_k represent the unit normal and length of the edge shared between the mesh elements $\pi(k, 1)$ and $\pi(k, 2)$. The action of the second term can be better understood by taking $\gamma(\vec{n}) = 1$ for all $\vec{n}$. In this special case, the second term reduces to the total length of the boundary between each component of the k-way partition, i.e. length of the phase boundary. Minimization of this term leads to the minimization of the phase boundary length which renders a smoothening effect. The total energy is minimized using the alpha expansion method discussed in [116] by solving the equivalent Multiway K-Cut problem. The data structure and the pseudocode for implementing this method is presented in the Appendix B.1.

$$w_k(f) = \gamma(\vec{n})\Delta s_k \quad (3.3)$$

Simultaneous minimization of the two energy terms gives rise to Pareto optimal solutions i.e. solutions where both terms cannot be simultaneously decreased any further. This means there exists a labeling such that any change in it either leads to deviation from the input image or introduces roughness in grain boundary. The solution is selected based on the choice of α in Eq (3.2). This phenomenon is illustrated in the next section by means of an example.

3.3 Results and Discussion

The salient features of the algorithm are illustrated using two examples. In the first example, polycrystalline materials are used with the grains representing the different phases of the material. Next, a case study of woven fibers composite is considered where the fiber and the matrix are identified as the two distinct phases. In polycrystalline materials, the smooth boundary is an artifact of microstructure evolution which is governed by surface energy minimization. According to the theory discussed in the previous section, the naive labeling based on the pixel location results in rough grain boundaries. The energy minimization problem is not required to estimate this labeling. In fact, it can be uniquely determined by the relation, $s_i = I(v_i)$, which was treated as the basis for formulating the data cost. This grain boundary is smoothened by minimization of energy form given in Eq (3.1). This process is illustrated in Fig 3.3 for a microstructure image with 200x200 pixels. A triangular mesh with 30,625 nodes and 60,547 elements is, first, labeled based on pixel positions. The boundaries are then smoothened by solving the Multiway K-Cut problem on the embedded graph containing 90,470 connections with the data parameter chosen as $\alpha = 0.002$ and the metric function chosen as $\gamma(\vec{n}) = 1$. The same procedure can be carried out for low-resolution images where pixel-based labeling may result in a very rough grain boundary. Grain identification for mesh elements using pixel-based and Multiway K-Cut based is presented for a low-resolution image of a pixel size of 50×50 in Fig3.4. It is observed that the smoothening effect is less prominent (in comparison to the high-resolution image) due to the pixelated nature of the base image.

The dependence of partitioning on the data term is shown in Fig 3.5. The Grain Boundary length ratio (i.e. the non- dimensional length of the grain boundary with respect to the unit cell) is calculated with respect to labelings for different data cost. Three regions are observed based on the value of the α defined in Eq (3.2). The first region corresponds to the case when higher weight is given to the minimization

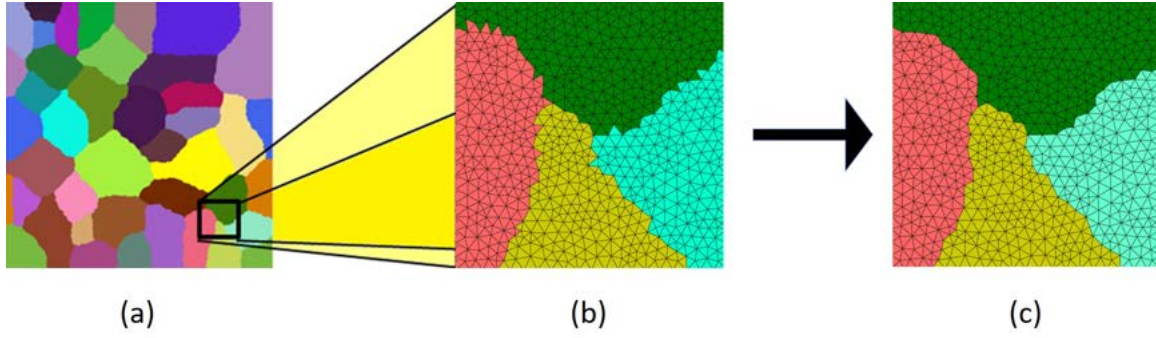


Figure 3.3: An illustration of labeling procedure: (a) Microstructure image (b) Pixel-based labeling (c) Smoothed mesh using Multiway K-Cut based labeling.



Figure 3.4: Grain identification in a mesh using a 50x50 pixel image: (a) Pixel-based labeling (b) Multiway K-Cut based labeling.

of length (low value of α). In this case, the addition of data cost is preferred over the addition of surface energy. On the contrary, labeling is governed by data cost in the third region. Therefore, this region is plagued with rough boundaries and an overall higher estimate of grain boundary length/area. Practical meshes with smooth boundaries and a minimal loss of grain structure are offered in the second region.

A comparison of the performance of this mesh with that of a uniform mesh with pixel-based labeling and an exact mesh (conformal mesh) is presented. A bicrystal with an incident angle of 60° between isotropic components is studied in uniaxial loading. For simplicity, the finite element simulations are presented for linear-elastic deformation assuming plane strain. The distribution of Von-mises stress is presented for the different meshes in Fig 3.6. The grain boundary is exactly mapped by the

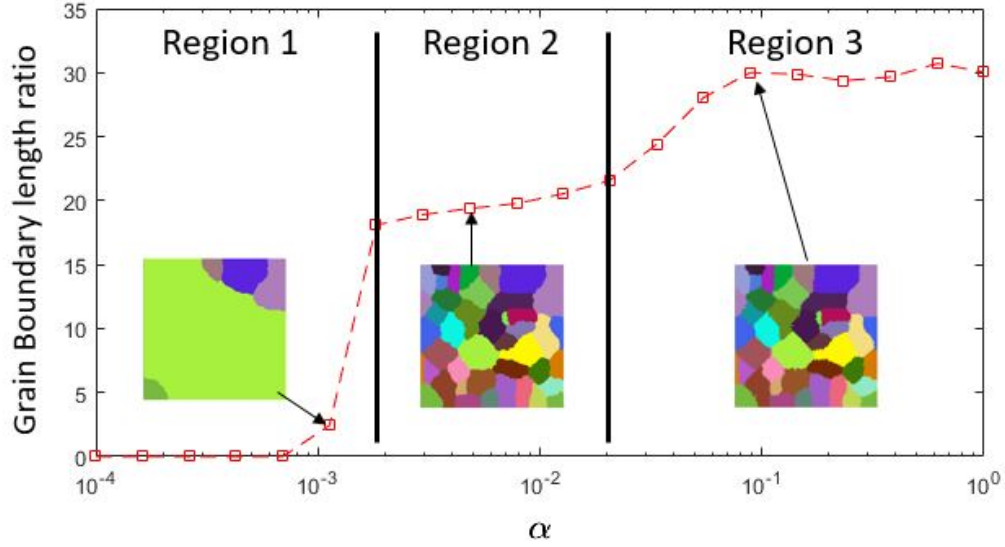


Figure 3.5: Dependence of Graph partitioning on the data term in energy.

edges of the conformal mesh. It captures all the localized data as well as the jump in stresses. In comparison, the pixel-based mesh has a diffused solution near the grain boundary and local features like stress concentrations are not well captured. Performance of graph-labeled mesh is very similar to the conformal mesh in terms of capturing discontinuity and localized stresses.

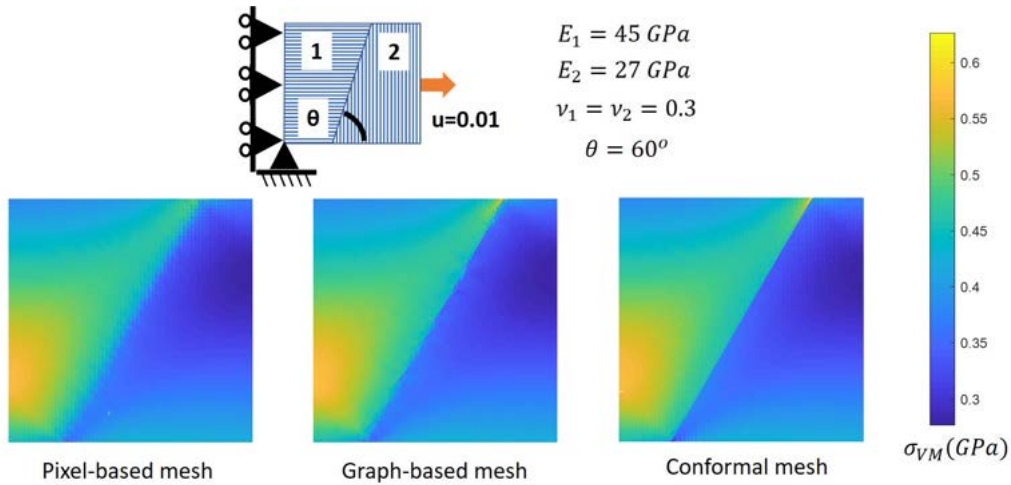


Figure 3.6: Comparative study of Finite element simulation results for a bicrystal in plane strain (linear elastic) using Pixel-based (quad mesh), Graph-based (triangle mesh with smoothing) and Conformal mesh

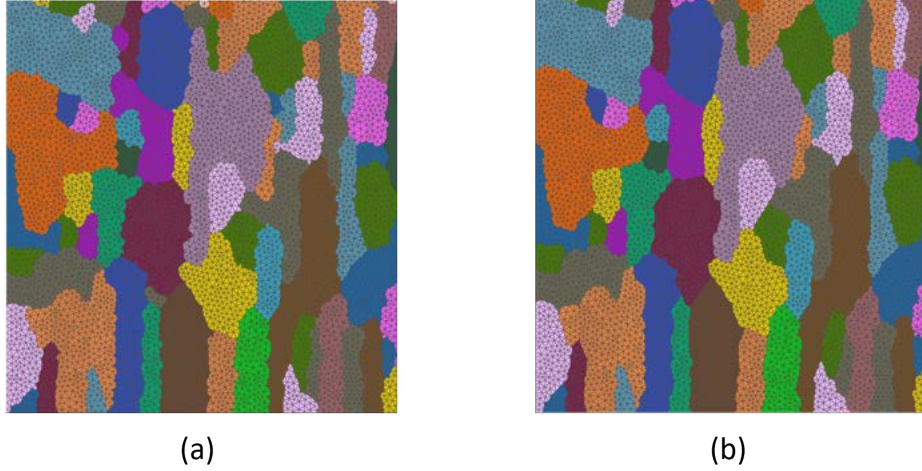


Figure 3.7: Grain identification for elongated polycrystals using (a) Euclidean metric (b) using Riemannian metric

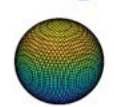
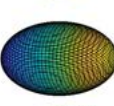
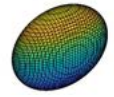
Grain type	$\mathbf{D}$	Iso-surface (Simplified Grain Shape)
Isotropic 2D grain	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	
Elongated 2D grain	$\begin{bmatrix} 0.2 & 0 \\ 0 & 1 \end{bmatrix}$	
Oriented 2D grain	$\begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$	
Isotropic 3D grain	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Elongated 3D grain	$\begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Pellet-like 3D grain	$\begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$	

Figure 3.8: Illustration of various morphologies based on the metric tensor, $\mathbf{D}$

Additive manufacturing techniques, like, selective laser melting (SLM), electron beam melting (EBM) and shaped metal deposition (SMD) often results in microstructure with elongated grains [117, 118, 119]. These structures can be better captured using the characteristic alignment of the grains. Motivated by the elliptical shape of grains, a non-euclidean distance measure can be used to favor or penalize certain directions.

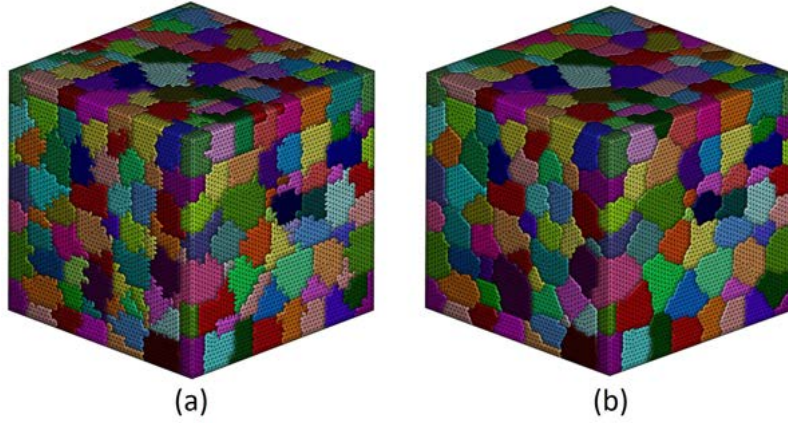


Figure 3.9: Grain identification in 3D meshes using a $64 \times 64 \times 64$ voxel base image: (a) Vertex position based labeling (b) Multiway K-Cut based labeling

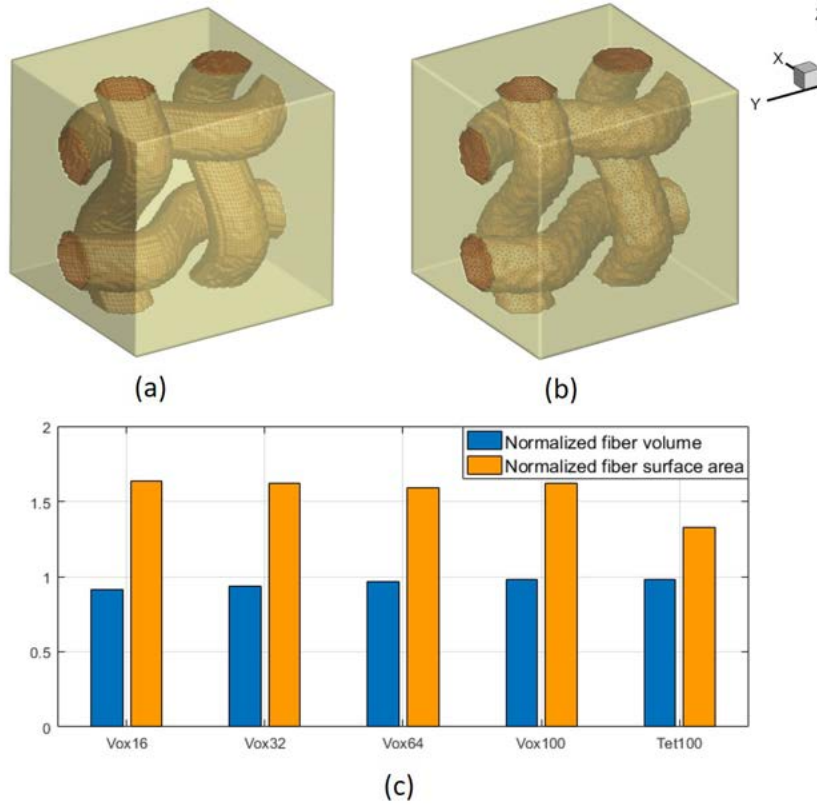


Figure 3.10: Mesh generation for woven fiber composite (a) Voxlated mesh with $10^2 \times 10^2 \times 10^2$ elements (b) Tetrahedral mesh with multiway K-Cut based labeling (c) Comparison of fiber volume and surface area of various voxelated meshes and a tetrahedral (smoothened) mesh

A useful class of such distance measures is a Riemannian norm. For this purpose, the

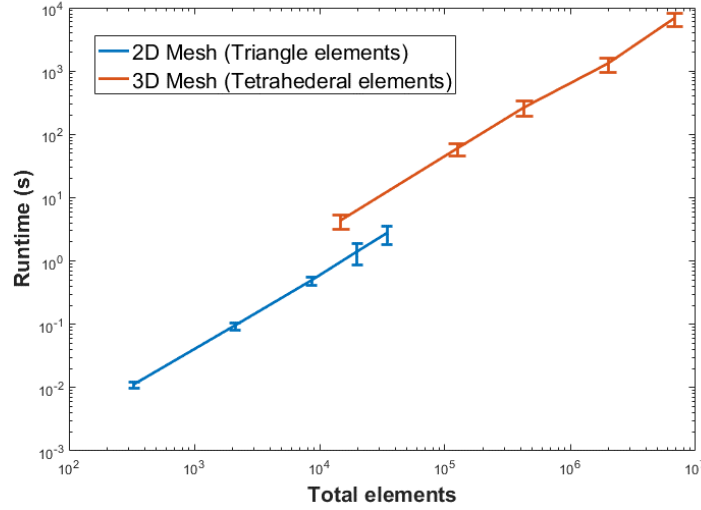


Figure 3.11: Time complexity analysis of the mesh generation procedure.

function γ can be specialized as Eq (3.4).

$$\gamma(v) = \sqrt{v^T \cdot D \cdot v} \quad (3.4)$$

where D is a positive definite matrix. In general, polycrystalline materials with non-equiaxed morphology show smoother grain boundaries in the normal/transverse direction (ND/TD). This directional smoothing is achieved by penalizing the length of a line element in an appropriate direction using the metric presented in Eq (3.4). An example of elongated microstructures is presented in Fig 3.7. It is observed that a metric with $D_{11} = 1$, $D_{22} = 10$ and $D_{12} = D_{21} = 0$, minimizes the kinks in the normal direction. This procedure can be specialized to many other kinds of morphologies using the matrix, D . The iso-surfaces for the distance function, $\gamma(\vec{n})$ for different choices of matrix D are presented as an illustration in Fig 3.8. Based on the theory of Wulff construction, the iso-surface represents the grain shape ([120]).

As suggested in section 3.2, this procedure can be easily extended to 3D meshes. For the purpose of illustration, a $64 \times 64 \times 64$ voxelated image of a microstructure is superimposed on an FE mesh with tetrahedral elements. The sample solution is presented

in Fig 3.9 for a mesh with approximately 3×10^5 elements. This smoothening procedure is also illustrated for a woven fiber composite system. A voxelated mesh and a tetrahedral (smooth) mesh with approximately 10^6 nodes are presented in Fig 3.10. A comparative study of convergence in volume and surface area is also presented in Fig 3.10(c). Different voxelated meshes are used with 'Vox $\langle n \rangle$ ' representing a mesh with $(n + 1)^3$ nodes. The fiber composite with Vox100 mesh is shown in Fig 3.10(a) with the tetrahedral mesh (Tet100) in Fig 3.10(b). The normalization is done with respect to the analytical values of respective quantities. It is observed that the estimate of fiber volume is almost consistent in all the meshes but the estimate of fiber surface area is improved in the tetrahedral mesh.

The time complexity of the algorithm is experimentally studied for 2D and 3D test cases. Ten instances of meshes with a similar number of elements are labeled using graph partitioning. The total average time for graph partitioning using alpha expansion is calculated for the 2D triangle and 3D tetrahedral elements. An almost linear trend is observed in Fig 3.11 (log-log scale is shown but note that the simulation time scales linearly with number of elements). This is faster in comparison to the experimental results presented in [121]. Faster heuristics are seen for the 2D mesh in comparison to 3D for the same number of elements. This is due to the fact that the embedded graph in the 2D case has a planar topology and Graphcut algorithm is known to be more efficient in these cases.

3.4 Conclusion

Based on the ICME paradigm, there has been increasing interest in the use of microstructure modeling tools for optimization of aerospace materials manufacturing processes. Experimental evaluation of the spatial distribution of multiphase materials provides images in a voxelated/pixelated format. However, such a format does not accurately capture features like grain surface area and curvature that are important

for modeling fracture. In addition, finite element simulation on voxelated microstructures may lead to the prediction of spurious stresses due to the stepped, block-like representation of otherwise smooth boundaries. In this chapter, a theory is developed for conversion of as-measured voxelated microstructure to an unstructured grid with a smoother representation of boundaries. This problem is solved using Graph Partitioning (GP) theory discussed in Chapter VI. This procedure smoothens the grain boundaries using a Potts energy model while optimally preserving the grain label information of the initial experimental data. The trade-off between smoothening and data preservation is controlled using a user-controlled parameter. The performance is tested for both equiaxed and non-equiaxed morphologies using Riemannian metric measures. It is observed that smoother grain boundaries in the normal/transverse direction (ND/TD) can be realized using this approach. The methodology has an efficient run-time and is easily extendable to 3D structures like woven fiber composites and 3D polycrystalline material as demonstrated in this work.

CHAPTER IV

Graph Coloring Method for Modeling Microstructural Fracture

A multiscale graph theory-based approach is introduced here to predict the microscale crack path in polycrystalline materials. The crack path is represented as the boundary of the partition of a geometric graph. The partitioning is carried out by optimizing an Ising-type hamiltonian. The hamiltonian parameters are chosen such that each partition cost is the same as the energy of the corresponding crack. The interplay of the loading conditions on the specimen and the microstructure of the material near the crack tip determines the crack growth angle in polycrystalline materials. Two different length scales of macro and micro are incorporated for the crack path by defining the crack total energy as the summation of macroscopic energy release and microscopic surface energy. The former term represents the macroscopically favorable crack growth angle by directing the crack to propagate from the crack tip along the direction of the maximum energy release. The latter term guides the microscopic crack path along the macroscopically preferred direction. At this scale, the crack path naturally accommodates both the intergranular and transgranular fractures. In the case of intergranular fracture, the crack propagates along the grain boundaries, while in the case of transgranular fracture, the crack propagates along the crystallographic cleavage planes. The mixed-mode fracture in a thin foil specimen is studied, and

the effect of the dihedral angle of the 2D crack is included in defining the effective surface energy. The model is validated using the analytical results for mixed-mode fracture in an isotropic medium and mode-I fracture in a medium with a preferred crack direction. The proposed method can be used to design materials microstructure with optimal fracture resistance.

4.1 Background

Traditionally, the two main types of fractures considered at the microscale are brittle and ductile. In brittle failure, the metal fractures with little to no plastic deformation, generating surfaces with bright cleaved crystalline facets. The crystallographic planes of the microstructure play an essential role in the mechanism of fracture. For instance, at low temperatures, BCC-Iron (Body centered cubic) cleaves on the $\{001\}$ family of planes and HCP-Zinc (Hexagonal close packed) cleaves on the basal plane, while FCC (Face-centered cubic) materials are usually immune to such failure. In contrast, fracture in ductile failure is preceded by significant plastic deformation, which redistributes the stress concentration ahead of the crack tip, resulting in different fracture mechanisms. The same kind of material can undergo ductile or brittle depending on the external conditions. One of the earliest pioneering studies of transition from brittle to ductile was carried out by Kelly, Tyson, and Cottrell [122]. They proposed that the ratio of the largest tensile stress and the largest shear stress close to the crack tip governs the crack behavior. A brittle fracture occurs when this ratio exceeds the ratio of ideal cleavage stress and the ideal shear stress.

There are two popular approaches in the literature for the numerical simulation of fracture. The first one is based on cohesive zone models ([123, 124, 125, 126]). These methods have proven to be extremely useful in predicting both the nucleation and crack propagation; however, they are plagued by numerical instabilities. These problems have been mitigated, to a certain degree, using either more stable but computa-

tionally extensive load path-following procedures (e.g., Riks method and arc-length method) or by introducing artificial viscosity to the cohesive zone law [127] leading to inaccuracies in load prediction. The second approach is based on Phase-field methods, where cracks are represented using diffuse damage parameters. The evolution of these damage parameters can be modeled using a Ginzburg-Landau type potential ([128, 129]) or using gradient damage theory [130]. There is also the issue of mesh dependence in numerical crack path prediction. It has been shown, for instance, in [131], one of the major reasons for mesh dependence is the stress concentration at the tip. Consequently, mesh-dependent objects like element size near the crack tip can influence the direction of crack growth. These problems are usually addressed using regularization approaches to limit the sensitivity of damage evolution to the stress concentration. Most phase-field studies (for e.g. [130, 132]) use regularization by considering a diffused crack. Moreover, many FEM techniques now allow generating sophisticated crack paths like the node enrichment FEM (e.g. X-FEM)[133] and elemental enrichment FEM (e.g. E-FEM) [134, 135, 136, 137].

One of the major drawbacks of these approaches is that the solution may not lead to a global minimum of energy due to the problem's non-convex nature. This aspect of the global and local solutions can be better understood in the variational model of quasistatic crack evolution proposed by Francfort and Marigo[138]. Consider the following energy of a body with a given crack, Γ , and external load, U :

$$E(\Gamma, U) = E_d(\Gamma, U) + \int_{\Gamma} 2\gamma ds$$

where E_d is the bulk energy of the body, and γ is the surface energy density of the cracked surface. The crack path is estimated using an evolution law where the load is monotonically increased, and the crack progresses in the direction of a maximum decrease in energy. However, this evolution does not need to lead to a global minimizer

of energy. In fact, it is mentioned in [138] that these evolution laws are dependent on the load history and have meaningful continuum limits only in certain conditions. At the same time, the reader is cautioned that this not an endorsement of global energy minimization over local growth law in terms of thermodynamic arguments. The global energy minimization is a postulate which seems crucial in understanding the effect of inhomogeneity of surface energy in polycrystalline materials. As an example, an experimental image for crack propagation in WE43 Mg alloy specimen presented in [1] (see Fig 4.1) revealed a crack path which did not propagate completely in the basal direction. Transgranular fracture path in Grain E shows an unusual bending before hitting the grain boundary. A possible explanation is that the intergranular fracture between the grains G and H lowers the overall energy of the crack and therefore influences the crack path in the transgranular region of grain E and F. In comparison, the numerical study of grain boundary effect on the crack path presented in [132] shows this upstream influence at a much smaller length scale. Such results motivate the search for global minimizers of total energy.

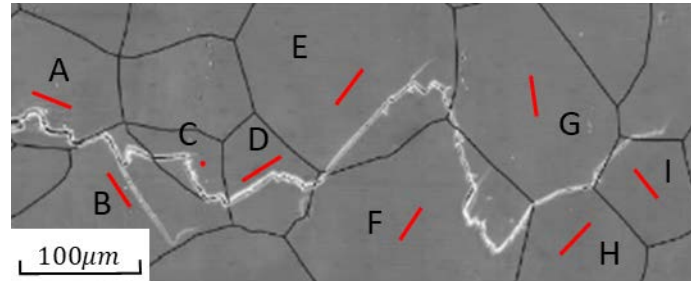


Figure 4.1: Experimental image of a fatigue crack in WE43 Mg alloy (HCP) adapted from [1]. Transgranular fracture followed the basal trace (red line) in grains A-F. Intergranular fracture occurred between grain pairs (A,B), (G,H) and (G,I).

On the other hand, graph-based approaches are posed as combinatorial optimization problems with finite choices where, in some instances, the global minimum can be approximated reasonably quickly. In regards to fracture mechanics, such approaches have been employed in predicting fracture at the microscale. Sundararaghavan and Srivastava [105] developed MicroFract, a graph theory-based software to predict the

microstructural crack path in various materials. They validated their model by simulating the intergranular stress corrosion cracking of Inconel. Srinivasan et al. [139] used machine learning along with graph theory to substitute demanding computations with coarse-scale graphs. Hunter et al. [140] also incorporated machine learning along with graph theory and developed reduced-order models, which have sufficient accuracy and low computational cost, to capture brittle fracture. Recent machine learning techniques for fracture prediction have been reviewed in [141]. Needleman and coworkers [142, 143] have developed a model based on discrete unit events along with graph theory to predict intergranular fracture in polycrystalline metals. They investigated the effect of grain size on the propagation of ductile intergranular fracture. There is no graph-theoretic approach that can simulate microscale cracks in polycrystals, including intergranular and transgranular cracks, to the best of authors' knowledge.

This work aims to introduce a general and modular framework for multiscale modeling of a polycrystalline material using a graph-based technique. The macroscopic energy release and the microscopic surface energy, depending on the loading condition and the crystallography of the material, respectively, are encoded into an Ising-type hamiltonian. The non-local component of the hamiltonian (Smoothing term) is obtained as a metric, and therefore, the energy minimization is carried out using the Alpha-expansion method, which guarantees that the min-cut is within a known factor of the global minimum (Kolmogorov and Zabih [144]; Boykov and Kolmogorov [145]). Consequently, the energy of the predicted crack is very close to the globally optimal crack path. The presented scheme can model the crack propagation in mixed-mode-I-mode-II failure in a reasonable time with sufficient accuracy. Furthermore, the algorithm reduces to a Quadratic Unconstrained Binary Optimization (QUBO), which can be solved using quantum annealing systems, like D-wave. A new scheme is developed to incorporate the dihedral angle of cracks in fracture, which considers the

effective surface energy of a uniformly oriented out-of-plane crack. The techniques presented here are specialized for brittle fracture in elastic materials. The key idea of this work to represent the fracture energy using the Ising model can be extended to ductile materials, but no such attempt is made in this chapter.

The chapter is organized as follows: Section 4.2 develops the formulation for the total energy of the crack in a multiscale setting and relates it to the Ising hamiltonian of the graph. A procedure to estimate the 2D surface energy of 3D crystals is also presented in this section for studying fracture in thin-film specimens. In section 4.3, the algorithmic details are described, and a discussion on the computational complexity of the method is presented. In section 4.4, the procedure is applied to capture crack paths in two cases : (i) homogeneous media with an anisotropic surface energy and (ii) polycrystalline material. The results for crack growth in homogeneous media are compared against the analytical result. In the case of the polycrystalline material, a qualitative verification of results is presented. The effect of the grain boundary on the crack path is studied, and it is shown that at high loading conditions, the predicted crack path resembles the dynamic fracture path. Section 4.5 provides a summary of the chapter.

4.2 Mathematical Framework

Among the early energy-based methodologies for studying fracture in linear-elastic isotropic materials, the most noted ones are the S-Criterion (Strain Energy Criterion) [146, 147] and the ME-Criterion (Maximum energy release rate criterion) [148]. It was first shown in [149] and later proved in [148] that the ME-criterion is the same as Maximum-Stress Theory [150] i.e., the crack propagates in the direction of maximum circumferential stress. For this reason, the ME-criterion is often favored over the S-Criterion for crack angle prediction in mixed-mode failure. The basic hypothesis for ME-Criterion is that:

1. The crack propagates from the crack tip in a straight line along which the maximum energy is released.
2. The crack propagates when the energy release rate is greater than a critical value.

Consider a small kink from the original main crack tip at an angle, α with the horizontal axis. The application of the ME-Criterion to estimate of the propagation angle can be formalized as:

$$\max_{\alpha} G(\alpha) \geq G_c \quad (4.1)$$

where $G(\alpha)$ denotes the energy release rate for a kink angle, α , and G_c denotes the critical energy release rate. In the case of an ideal brittle fracture, $G_c = 2\gamma$, where γ is the surface energy density of the fractured surface. Now, consider a circular region around the main crack tip determined by the radius, $\bar{r}$, of the branched crack and a microscopic crack path, Γ , parameterized as $\Gamma(s) \equiv (x(s), y(s))$. In this region, the ME-Criterion can be equivalently written in terms of the total energy released (instead of the energy release rate) as follows:

$$\min_{\Gamma} \int_{\Gamma} (2\gamma - G) ds \leq 0$$

The above equation was derived by using the fact that G_c , or equivalently, γ is independent of α , which is true for isotropic materials. However, at the microscale, γ depends on the location and the local orientation of the cracked surface. Consequently, the minimizer of this functional is not necessarily a straight line, as postulated in the first hypothesis of the original ME-criterion. Moreover, the second hypothesis is also violated as the equivalent G_c changes with the kink angle. In favor of a multiscale approach, the energy release rate is assumed to be governed by the macroscopic feature of the crack, i.e., kink angle, defined as the average angle of the microscopic crack.

The assumptions considered in the new hypothesis for multiscale crack propagation are stated as:

1. Crack, Γ propagates from the crack tip in a possibly zig-zag direction towards the circular region's circumference so that the path minimizes the total energy of the system, $\varepsilon(\Gamma)$:

$$\varepsilon(\Gamma) = \int_{\Gamma} 2\gamma ds - G(\alpha_{\Gamma})$$

2. Crack propagation is initiated when the total energy is negative.

The energy release rate for mixed mode crack in isotropic material is estimated in [148] as follows:

$$G(\alpha) = \frac{\kappa + 1}{8\mu} \cos^2 \frac{\alpha}{2} \left(\cos^2 \frac{\alpha}{2} K_{II}^2 + \left(\cos \frac{\alpha}{2} K_I - 2 \sin \frac{\alpha}{2} K_{II} \right)^2 \right) \quad (4.2)$$

where κ is a material constant, which can be obtained as $\kappa = 3 - 4\nu$ for the plane strain condition and $\kappa = (3 - \nu)/(1 + \nu)$ for the plane stress condition. The variables, μ and ν are the shear modulus and Poisson's ratio for the homogenized material, respectively. Note that the energy release rate for a general anisotropic material ([151]) can be substituted here without influencing the formulation.

4.2.1 Graph representation of Microscopic crack

A random finite element mesh is considered where each element is a polygon represented by P_i with $i \in \{1, \dots, n\}$. The goal is to label each element as $+1$ or -1 based on a cost minimization criterion. Therefore, a particular partition of a mesh is given by an array of labels, $L = [l_1, \dots, l_n]$ with $l_i \in \{+1, -1\}$. The crack is represented as the element edge with different labels on either side. To define the energy of this partition, an undirected graph is embedded in this mesh. Each element, P_i , is treated

as a vertex, v_i , of the graph, $\mathcal{G}$. A connection, e , exists between two vertices if their respective elements share an edge¹. This cost is determined as the Ising hamiltonian, H_I , on the graph, $\mathcal{G}$, defined as:

$$H_I(L) = \sum_{v_i \in V} g(c_i, l_i) |P_i| + \sum_{(v_i, v_j) \in E} 2\gamma(\vec{n}) \Delta s_{ij} (1 - \delta_{l_i - l_j}) \quad (4.3)$$

where $g(\cdot, +1)$ and $g(\cdot, -1)$ are continuous functions defined over the domain. c_i and $|P_i|$ are the centroid and the area of the element, P_i , respectively. Δs_{ij} and $\vec{n}$ are the length and a unit normal to the interface between the elements, P_i and P_j , respectively. δ_x is the Kronecker-delta function, i.e. it has value 1 at $x = 0$ and 0 everywhere else. The function γ is strictly positive and represents the surface energy density of the crystal. It can be immediately observed that the second term does not contribute towards the cost if the labels of neighboring pair are the same.

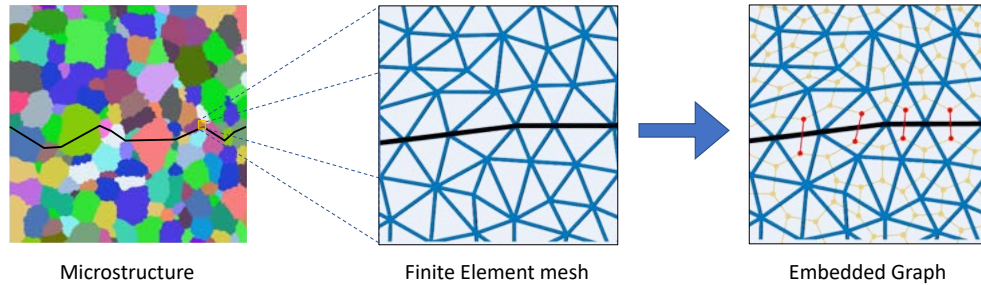


Figure 4.2: A portion of the FE mesh (in blue) with the dual graph, $\mathcal{G}$ (in yellow) embedded in the mesh. Only the connections across the crack (in red) contribute to the cost. The energy of the connection between elements A and B is determined by the length and the normal of the interface edge.

In the next section, the procedure to encode the total energy as an Ising hamiltonian is presented. The first term in Eq (4.3) is used to encode the macroscopic energy release, while the second represents the microscopic surface energy of the crack.

¹In the subject of Graph theory, this construction is referred to as the weak dual of a primal graph, where the vertices and connections of the primal graph are defined identified as the nodes and the edges of the mesh respectively.

4.2.2 Graph based modeling of Energy release

A semicircular region of radius, $\bar{r}$ can be defined as an area of interest where the macroscopic crack grows from the center to the circumference in a straight line. The net energy release increases linearly with the radius, $\bar{r}$. Moreover, it has functional dependence on the kink angle. To encode this behavior of the energy release, following functional form of g is considered as an ansatz:

$$g(c_i, l_i) = (2 - k)\bar{r}^{k-1}\frac{\bar{g}(\theta_i, l_i)}{r_i^k} \quad (4.4)$$

where, θ_i and r_i define the angular and radial coordinates of the point c_i and $k < 2$ for admissibility. Consider a radial crack at an angle α . To understand this choice of function, g , observe that in the continuum limit of the mesh ($\mathcal{P}_h$ with $h \rightarrow 0$), the first term of Eq(4.3) can be expanded as (see PropositionA.1):

$$\sum_{v_i \in V} g(c_i, l_i)|P_i| = \bar{r} \int_{\alpha}^{\pi/2} \bar{g}(\theta, +1)d\theta + \bar{r} \int_{-\pi/2}^{\alpha} \bar{g}(\theta, -1)d\theta \quad (4.5)$$

where $\{+1\}$ is the label of upper region and $\{-1\}$ is the label for lower region. Moreover, the above equation represents the negative of the energy release in the continuum limit, i.e.

$$\bar{r} \int_{\alpha}^{\pi/2} \bar{g}(\theta, +1)d\theta + \bar{r} \int_{-\pi/2}^{\alpha} \bar{g}(\theta, -1)d\theta = -\bar{r}G(\alpha) \quad (4.6)$$

Therefore, the initial ansatz for g satisfies the linear dependence on $\bar{r}$ and angular dependence on the crack angle, α . This integral equation form is satisfied by the following form of $g(., l_i)$ (see PropositionA.2):

$$\bar{g}(\alpha, +1) = G'(\alpha)H(G'(\alpha)) + c$$

$$\bar{g}(\alpha, -1) = -G'(\alpha)H(-G'(\alpha)) + c \quad (4.7)$$

where H is the heaviside function and c is given by:

$$c = -\frac{1}{\pi} \left(G\left(-\frac{\pi}{2}\right) + \int_{-\pi/2}^{\pi/2} G'(\theta)H(G'(\theta))d\theta \right)$$

The derivative G' can be estimated from Eq(4.2) as:

$$\begin{aligned} G'(\alpha) = & -\frac{\kappa+1}{8\mu} \cos \frac{\alpha}{2} \left(\left(\sin \frac{\alpha}{2} + \sin \frac{3\alpha}{2} \right) \frac{K_I^2}{2} \right. \\ & \left. + \left(5 \sin \frac{\alpha}{2} - 3 \sin \frac{3\alpha}{2} \right) \frac{K_{II}^2}{2} - 2 \cos \frac{3\alpha}{2} K_I K_{II} \right) \end{aligned} \quad (4.8)$$

Moreover, a phenomenological choice of $k = 1$ is made in this chapter. To understand the reason behind this, consider a rough crack represented using its angular position (α) as a function of the radial position. The following relation holds true in the continuum limit (see PropositionA.3):

$$\sum_{P_i \in \mathcal{P}_h} g(c_i, l_i) |P_i| = (k-2)\bar{r} \int_0^1 \frac{1}{s^{k-1}} G(\alpha(\bar{r}s)) ds \quad (4.9)$$

Therefore, the parameter k can be used to give different weightage to energy release rate at different radial distances. It can be easily observed that for the given choice of k , the excess energy release rate is weighed uniformly.

4.2.3 Modeling surface energy

The surface energy density, γ , in Eq (4.3) varies spatially and is dependent on the angle of the cracked surface. The treatment of edges between elements of the same grain (transgranular) is done differently from the ones with a distinct grain (intergranular). Here, the energies related to both the crack growth pattern of transgranular

and intergranular types are elaborated.

Transgranular surface energy: Transgranular cracks usually prefer propagation along crystallographic planes, typically those with low indices. This effect is due to the relatively smaller energy density for these planes. Due to the symmetry of crystals, the surface energy density also obeys these rules, i.e. for the function γ^{tg} to represent the surface energy density of a crystal system, it must satisfy: $\gamma^{tg}(Q \star \vec{n}) = \gamma^{tg}(\vec{n})$, where Q is an element of the group, $\mathcal{G}$, consisting of appropriate symmetry operations, and $Q \star \vec{n}$ denotes the action of Q on $\vec{n}$. These energies are often available in the literature for specific crystallographic planes. The symmetry conditions can then be encoded using appropriate interpolation. This practice is fairly common in smeared crack approaches; for instance, surface energy density interpolation for crystals with cubic symmetry is introduced in [152].

As an example, a model for transgranular surface energy density of a crystal with cubic symmetry is chosen. In this case, $\mathcal{G}$ is the group generated by the operations:

$$\{R_x(\pi/2), R_y(\pi/2), R_z(\pi/2), -I\}$$

where R_x , R_y and R_z are rotations about the chosen coordinate system and $-I$ is the inversion operator. The surface energy density of the crack for a cubic material is modelled as:

$$\gamma^{tg}(\theta, \phi) = \gamma_0 \left(1 + 3\delta \left(\frac{\cos^2 \theta \sin^2 2\phi}{4} + \sin^2 \theta \right) \cos^2 \theta \right) \quad (4.10)$$

where θ and ϕ are the elevation and the azimuth of the unit normal to the cleavage plane relative to the crystal frame of reference. The $\{001\}$ family of planes has the minimum energy given by the parameter, γ_0 , while the $\{011\}$ family has a saddle with energy $\gamma_0(1 + 3\delta/4)$ and the $\{111\}$ family has a maximum with energy $\gamma_0(1 + \delta)$. An

illustration of this model is shown in Fig4.3 where the cubic symmetries are evident.

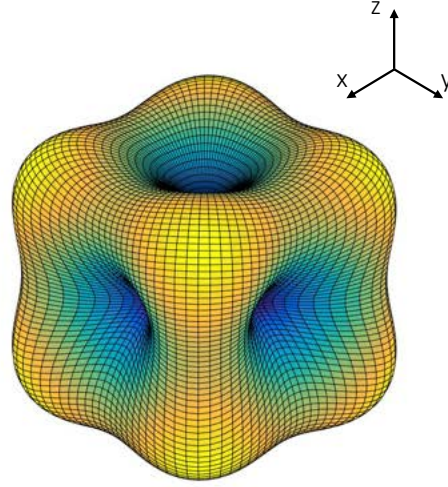


Figure 4.3: An illustration of the 3D surface energy density of a cubic material with $\gamma_0 = 1J/m^2$ and $\delta = 2$ in the direction normal to the cleavage surface.

Intergranular surface energy: An intergranular crack propagates along the grain-boundaries. And the grain-boundary energy, γ_{GB} , determines the surface energy. The effective intergranular surface energy density of the cracked surface, γ^{ig} , is determined as:

$$\gamma^{ig} = \gamma_{\text{surface}} - \frac{1}{2}\gamma_{GB} \quad (4.11)$$

where γ_{surface} is the average energy required to break atomic bonds per unit surface area of individual grain species. As suggested by the above relation, grain boundaries with lower γ_{GB} are more stable as the surface energy is higher along those grain boundaries. These energies are crucial in modeling crack propagation via special grain-boundaries like the coherent twin boundary (CTB), which has the lowest energy.

4.2.3.1 Fracture in thin foils

During crack propagation in these films of crystalline material, unlike isotropic material, the dihedral angle (out of plane angle) cannot be assumed to be zero. Consider a specimen with uniform thickness and a 2D crack that extends in the third dimension with a dihedral angle, ψ_D . Therefore, the surface area generated by a unit 2D crack is

$\sec \psi_D$. The dihedral angle is chosen such that the total surface energy is minimized for all possible dihedral angles. It is also evident that the dihedral angle depends on the direction of the edge. Without a loss of generality, the microstructure surface is chosen to be on the $x_1 - x_2$ plane. The dihedral angle is defined as the angle between the unit normals, $\vec{n}_{2D}$ and $\vec{n}_{3D}$. The general unit normal to the 3D crack can be parameterized with an angle ψ as:

$$\vec{n}_{3D} = [0, 0, 1]^T \sin \psi + \vec{n}_{2D} \cos \psi$$

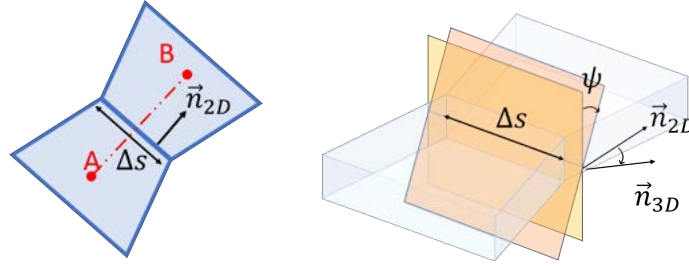


Figure 4.4: An illustration for the dihedral angle of a 2D crack.

And the dihedral angle is estimated as:

$$\psi_D = \operatorname{argmin}_{\psi \in (-90^\circ, 90^\circ)} \gamma(\vec{n}_{3D}(\psi)) \sec \psi \quad (4.12)$$

The effective surface energy for the 2D crack is then calculated as:

$$\tilde{\gamma} = \gamma(\vec{n}_{3D}(\psi_D)) \sec \psi_D \quad (4.13)$$

The surface energy density and effective dihedral angle for three different grain orientations are shown in Fig4.5 for a cubic material with the surface energy model defined as Eq(4.10). It can be observed that cracks with lower surface energy density are possible by considering the dihedral angle.

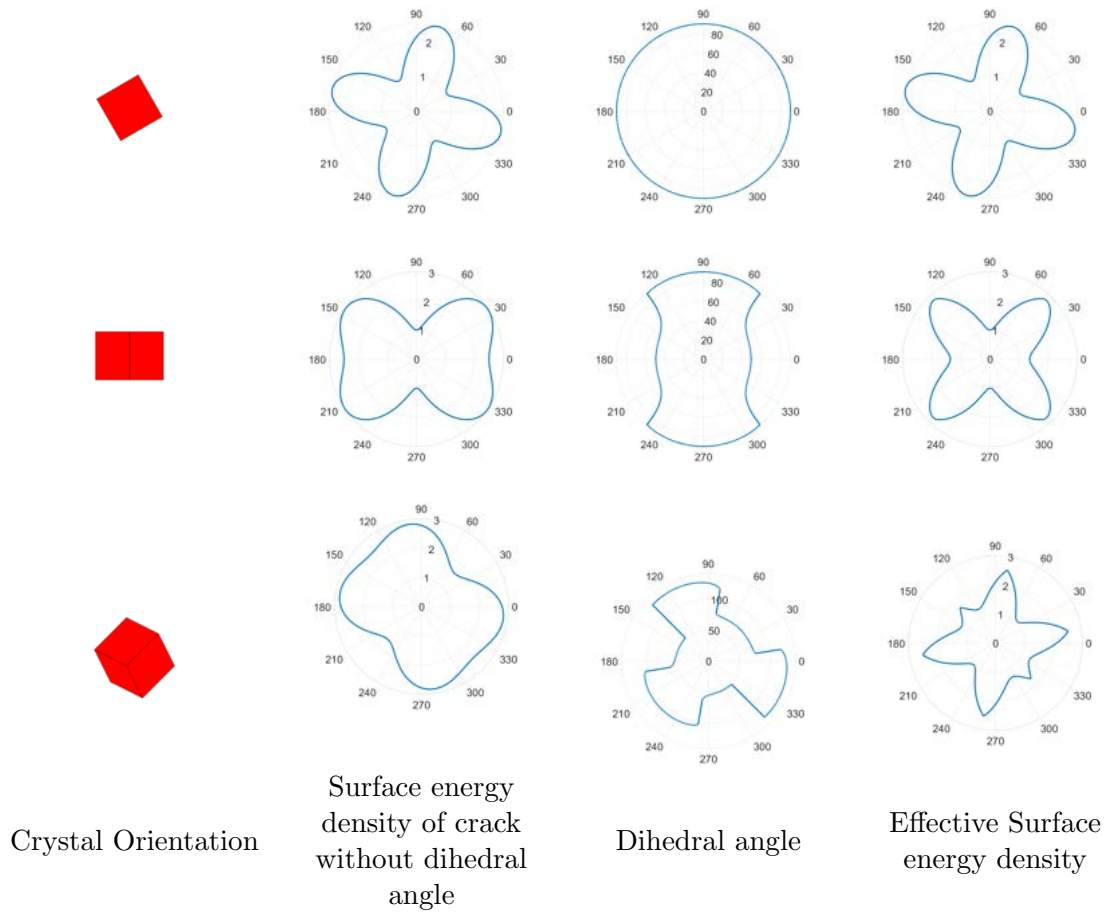


Figure 4.5: 2D surface energy density of a cubic material with $\gamma_0 = 1J/mm^2$ and $\delta = 2$. Each row represents a specific crystal orientation in the specimen frame of reference, and the polar plots are evaluated with respect to cleavage normals. 2D surface energy densities are evaluated for a specimen with unit thickness. The dihedral angle is plotted as $\psi_D + 90^\circ$ for positivity considerations in the polar plot.

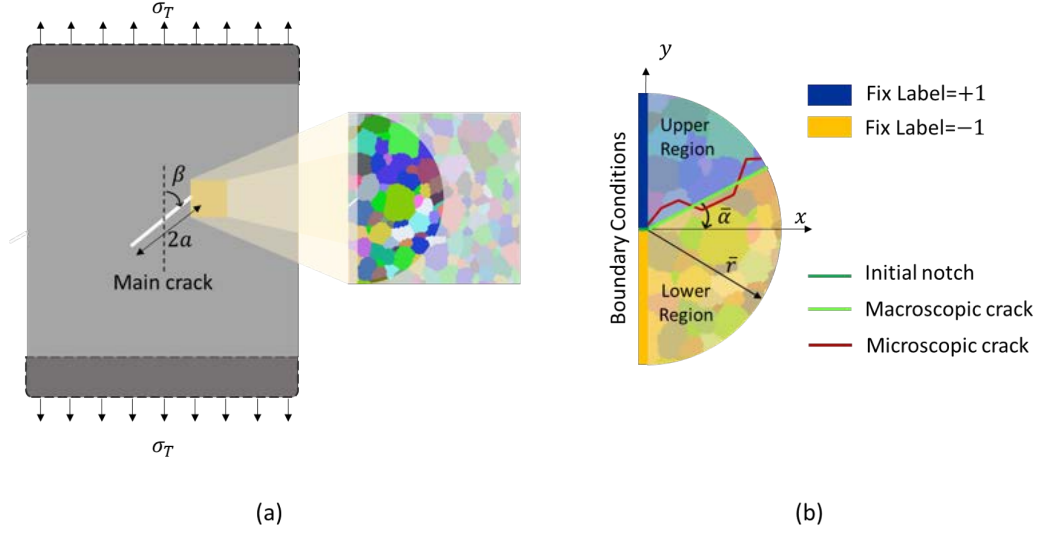


Figure 4.6: (a) A tensile test specimen for mixed mode failure (b) The definition of the macroscopic crack angle from the microscopic crack path.

4.3 Method

4.3.1 Problem Setup

A Tensile test is considered with an inclined main crack to replicate the conditions of Mixed mode failure (shown in Fig 4.6). The main crack has a half-length, a and an inclination angle, β .

The stress intensity factor can be estimated as:

$$\begin{aligned} K_I &= \sigma_T \sqrt{\pi a} \sin^2 \beta \\ K_{II} &= \sigma_T \sqrt{\pi a} \sin \beta \cos \beta \end{aligned} \quad (4.14)$$

where σ_T is the applied tensile stress.

4.3.1.1 Boundary Condition

The minimization of Eq(4.3) leads to two types of solutions: (1) Both labels are present - This means that there is a crack propagating at an effective angle, $\bar{\alpha} \in$

$(-\pi/2, \pi/2)$ (2) Only one label is present. The latter condition should not be confused with a vertical crack because there is no surface energy being generated in this case. Moreover, due to the lack of surface energy, this erroneous solution's total energy can be substantially low in the absence of a strong directional dependence of the macroscopic energy release rate. For this reason, boundary conditions are applied where the labels $+1$ and -1 are imposed on a thin strip on top and bottom of the main crack-tip (as shown in Fig4.6(b)). It is done by artificially increasing the value determined by the term $g(c_i, -1)|P_i|$ in the top strip and $g(c_i, +1)|P_i|$ in the bottom strip.

4.3.1.2 Checking total energy

After enforcing the boundary conditions, the minimization process predicts the crack path as per the Revised ME-Criterion's first hypothesis. However, this procedure does not check whether the total energy is negative; i.e., it violates the second hypothesis. This problem is resolved by checking the total energy after the labeling is done, and the solution is discarded if the total energy is positive.

4.3.2 Computational Procedure

Experimental imaging techniques usually provide a pixelated image with a grain index of each pixel and a table relating grain index to its orientation. This data is superposed on a mesh, which is generated using a Delaunay triangulation of randomly sampled points by determining the grain index of each element using the technique developed in [153]. This method ensures that the grain boundaries in the resulting mesh are smooth. The crack is identified as the boundary between the elements with different labels. This labeling is carried out by minimizing the cost function presented in Eq(4.3) using the Alpha-expansion method. The gco library developed by [116] is used to minimize the cost function. This library requires three costs functions namely,

label cost, LC, data cost, DC, and neighbor cost, NC. The label cost is given as:

$$LC(i, j) = 1 - \delta_{(l_i - l_j)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (4.15)$$

The neighbor cost defines the weight of connection between the i^{th} and j^{th} element, which can be defined as follows:

$$NC(i, j) = 2\gamma(\vec{n})\delta s_{ij} \quad (4.16)$$

The data cost describes the cost of labeling the j^{th} vertex. It is decomposed as, $DC = DC_0 + DC_{BC}$ where DC_0 denotes the data cost for emulating the macroscopic energy release and is given as

$$\begin{aligned} DC_0(1, j) &= g(\theta_j, +1)|P_j|/r_j \\ DC_0(2, j) &= g(\theta_j, -1)|P_j|/r_j \end{aligned} \quad (4.17)$$

The term DC_{BC} denotes the data cost used for imposing boundary condition. Taking k as a large constant, the term DC_{BC} is set as $DC_{BC}(1, j) = k$ when the j^{th} vertex is in the bottom strip and $DC_{BC}(2, j) = k$ when the j^{th} vertex is in the top strip. For all other indices, $DC_{BC} = 0$.

A flowchart for the computational procedure is presented in Fig 4.7. The list of all the variables and pseudo-codes for all the subroutines are provided in Appendix B.2. It can be observed that this procedure allows for a unidirectional workflow with (1) Mesh generation (2) Microstructure specific calculation, i.e., identification of grain ID and setting up Neighbor cost (3) Test-specific calculation, i.e., setting up data cost and estimation of the crack. Independence of Step(1) from Step(2) allows using the same random mesh multiple times by re-estimating the Grain index of the mesh

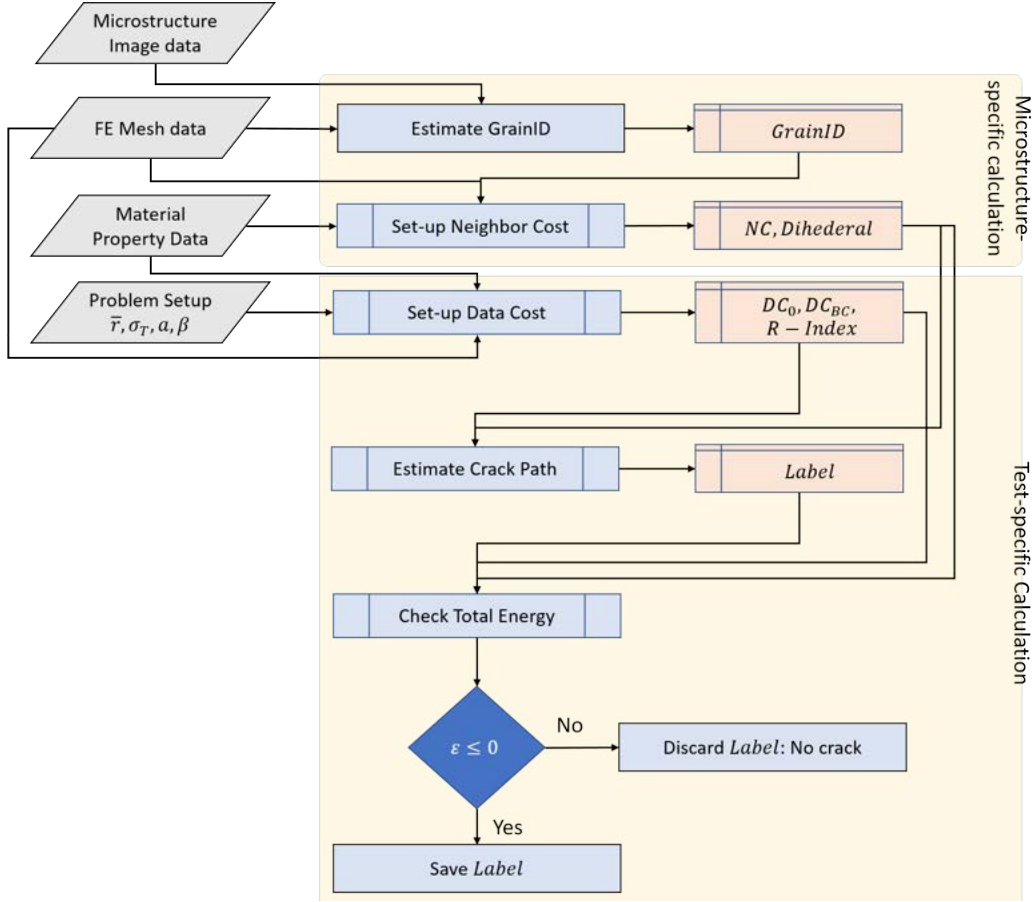


Figure 4.7: Flowchart of the procedure for crack estimation.

for different microstructures. Similarly, the independence of Step(2) from Step(3) relieves the burden of calculation of the Neighbor cost for different loading scenarios on the same microstructure. This effect is achieved by generating the list of indices, *R-Index*, of the elements in the radial region, and using it to extract the Neighbor Cost of the radial region.

4.3.3 Computational complexity

This procedure's total runtime can be estimated as the sum of the individual components of the workflow mentioned above. Before analyzing them separately, it is crucial to understand the resulting graph's topology. The Finite element mesh can also be thought of as a graph with its nodes as the vertices of a graph and the edges

as the connections of the graph. Since $\mathcal{G}$ is constructed as a subgraph of a dual of a planar graph (Mesh Graph), it is planar. Based on the construction, the number of vertices, $|V|$ of $\mathcal{G}$ is equal to the number of elements (N) in the mesh. Moreover, the connections, $|E|$ in $\mathcal{G}$ is bounded from above by $3N - 6$.

The *Estimate GrainID* subroutine developed in [153] uses a graph with the same topology as used in this work. The main bottleneck in this procedure, as well as the *Estimate Crack Path* procedure, is the GraphCut step. It has been shown in [154], that the runtime complexity for this step is $\mathcal{O}(N \log N)$. The complexity for the *Set-up Neighbor Cost* and *Set-up Data Cost* can be estimated as $\mathcal{O}(|V|)$ and $\mathcal{O}(|E|)$, respectively. Therefore, both the cost setup procedures are $\mathcal{O}(N)$. Finally, the *Check Total Energy* subroutine uses the sparsity of a graph to estimate the cost in $\mathcal{O}(|V| + |E|)$, i.e., it is bounded linearly by N . This analysis shows that the overall runtime complexity of this method is $\mathcal{O}(N \log N)$. The run-time of each process for various mesh-sizes is provided in Fig 4.8. It is observed that the *Set-up Data Cost* takes the most computation time for the given choices of the mesh sizes. However, the computation time for *Estimate Crack Path* shows the fastest growth with mesh size.

4.4 Application to examples

The essential characteristics of the brittle fracture using the proposed formulation are verified in this section. All simulations are conducted for quasistatic crack growth due to the abundance of analytical results in such cases. The quasistatic crack growth condition is replicated by choosing the value of σ_T such that the total energy, ε , is as close to 0 as possible while maintaining non-positivity. The reason for this choice is that, in practice, when the tensile load is increased, the crack propagation initiates as soon as the total energy of the crack is negative. The first set of simulations is conducted on a homogeneous material, and the results are verified against analytical

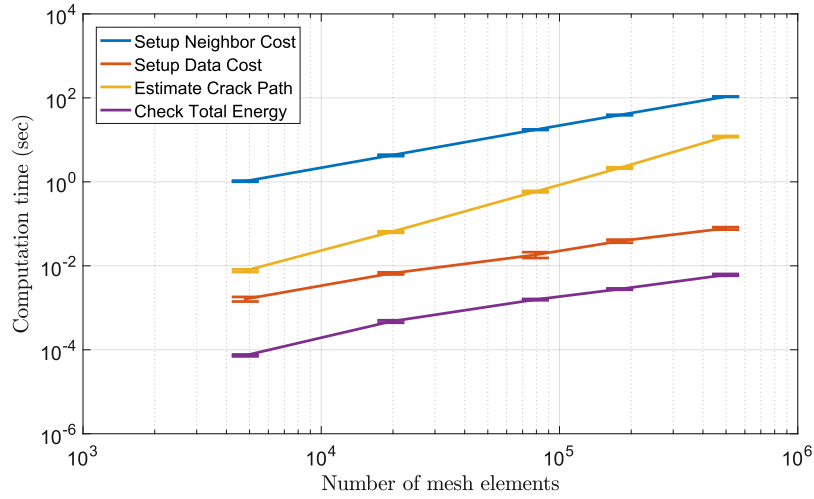


Figure 4.8: Run-time Analysis: Computation time is estimated for each process for different mesh sizes. A sample set of 20 meshes is used for each mesh size. Error bars represent the standard deviation in the computation time for each sample set.

results. In the next set of simulations, multiple grains are considered. In this case, analytical results are rare, and the results are analyzed in a qualitative sense. All numerical studies are done on a square domain, discretized into approximately $180k$ elements. Consequently, the complete graph, $\mathcal{G}$, has around $180k$ nodes and $270k$ connections. When restricted to the semicircular simulation domain, the sub-graph contains around $70k$ nodes and $105k$ connections. A mesh convergence analysis is presented in the case study for isotropic fracture (Section 4.4.1.1) that validates this choice of the mesh size. The radius ($\bar{r}$) of crack-growth region is chosen as half the image's dimension. This hyper-parameter defines the macroscopic length scale of the problem and influences the crack path in the polycrystalline medium. This effect is analyzed in Section 4.4.2.2.

4.4.1 Fracture in a homogeneous medium

Homogeneous materials have uniform properties throughout the domain. Three different cases are considered based on the form of surface energy density. In the first case, an isotropic material is studied. Next, a surface energy density with a single

preferential direction is considered. Finally, a surface energy density with multiple preferential directions is studied. For simplicity, all the surface energies are formulated for 2D, so the crack's dihedral angle is uniformly zero in all the cases studied here. It should also be noted that the anisotropy is only induced through the surface energy while the energy release rate is based on isotropic material (as defined in Eq(4.2)).

4.4.1.1 Isotropic crack

Firstly, the mesh convergence analysis is conducted using the test case of Mode-I fracture in this medium. The analytical crack grows horizontally while the numerically estimated crack grows in a zig-zag fashion about the horizontal line. The difference between the 'Analytical' and the 'Numerical' crack can be estimated using the Simple matching distance of the corresponding labeling problem. In this case, the distance can be evaluated using the following form:

$$d(C_{\text{Numerical}}, C_{\text{Analytical}}) = 2 \frac{\text{Area between}(C_{\text{Numerical}}, C_{\text{Analytical}})}{\pi \bar{r}^2} \quad (4.18)$$

where, $C_{\text{Numerical}}$ and $C_{\text{Analytical}}$ represent the Numerical and the analytical crack path. An illustration of the area along with the estimated between the two cracks is presented in Fig 4.9. It is observed that $d(C_{\text{Numerical}}, C_{\text{Analytical}})$ decreases with an increasing mesh size. The mesh with approximately $180k$ elements is chosen for its reasonable accuracy and computation time.

Next, it is verified that the revised ME-Criterion used in the formulation of the energy minimization problem is, in fact, equivalent to the original ME-criterion for an isotropic material. Erdogan and Sih [150] performed a mixed-mode fracture experiment using plexiglass sheets with oblique cracks under tension. The crack's average

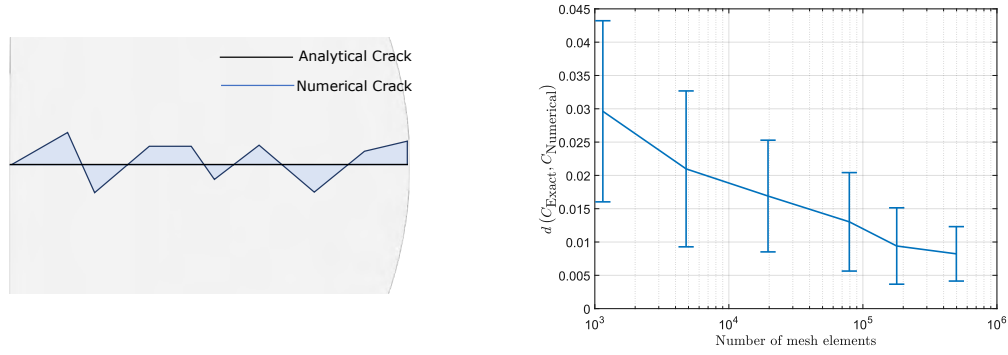


Figure 4.9: Mesh convergence analysis: $d(C_{\text{Numerical}}, C_{\text{Analytical}})$ is estimated for different mesh sizes. A sample set of 20 meshes is used for each mesh size. Error bars represent the standard deviation in $d(C_{\text{Numerical}}, C_{\text{Analytical}})$ for each sample set.

angle in their experiments is presented in Table 4.1. The crack angle, as predicted by the original ME-criterion, is also provided. The numerical simulation using the revised ME-criterion is provided in Fig4.10. The macroscopic crack propagation angle matches reasonably well with the experimental and analytical results.

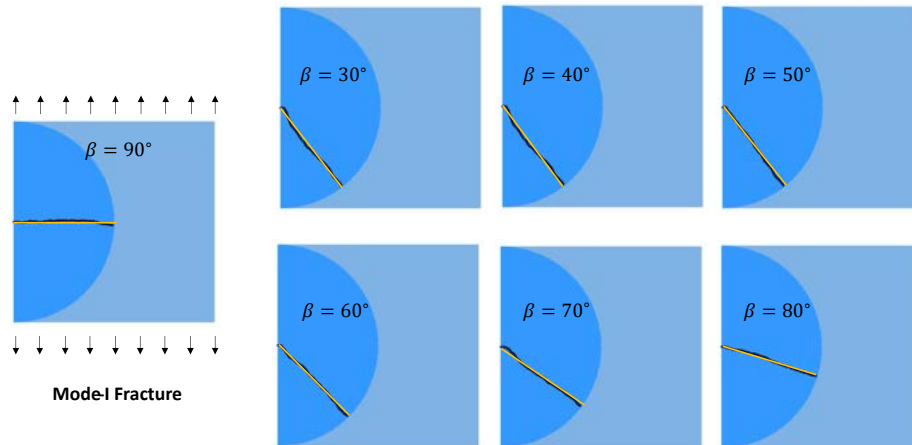


Figure 4.10: The crack path predicted in mixed-mode failure of an isotropic media. The black line represents the numerically simulated crack in the semi-circular region. The overlaid yellow line represents the corresponding macroscopic crack direction.

	$\beta = 30^\circ$	$\beta = 40^\circ$	$\beta = 50^\circ$	$\beta = 60^\circ$	$\beta = 70^\circ$	$\beta = 80^\circ$	$\beta = 90^\circ$
Experiment [150]	-62.4°	-55.6°	-51.1°	-43.1°	-30.7°	-17.3°	-
ME-criterion	-60.1°	-55.5°	-50.2°	-43.2°	-33.2°	-19°	0°
Numerical	-62°	-56°	-52°	-43°	-34°	-20°	0°

Table 4.1: Crack propagation angle in mixed-mode failure

4.4.1.2 Single cleavage plane

As the next example, a material with a preferred cleavage direction for the microscopic crack is studied. The surface energy of the crack is low in the prescribed direction of the cleavage plane. The following 2-parameter model for energy function with $\gamma_0, \delta > 0$ is used:

$$\gamma(\theta) = \gamma_0(1 + \delta \cos^2(\theta - \omega)) \quad (4.19)$$

where θ is the angle of the edge normal and $\omega \in [0, \pi]$ is the preferred direction of the microscopic crack, i.e., the minimum surface energy is generated when the crack propagates along the vector $[\cos \omega, \sin \omega]$. The parameter, γ_0 , denotes the preferred cleavage plane's surface energy, and $1 + \delta$ is the ratio of the maximum surface energy to the preferred cleavage plane's surface energy.

A horizontal macroscopic crack is preferred in Mode-I failure. However, the microscopic crack prefers an oblique crack with angle, $\alpha = |\omega - 90^\circ|$. The competition between these two effects results in a crack propagating in the direction with the absolute angle less than the one predicted by the surface energy. Takei et al. [155] introduced a Wulff's diagram based criteria to determine the effective angle of the crack. In the case of a differentiable form of surface energy and Mode-I failure, their method can be described as: the crack angle is determined by the point on the γ^{-1} polar plot (with respect to the crack propagation angle) which is tangentially touched

by a vertical line moving continuously from right to left. The polar plot of γ_0/γ with

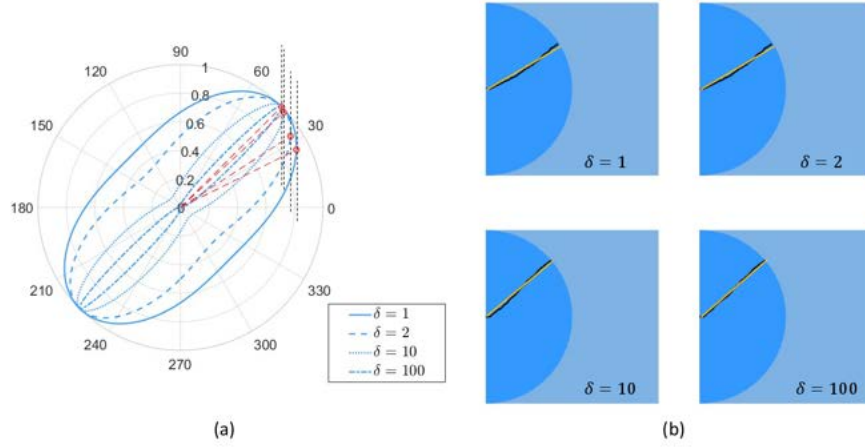


Figure 4.11: The crack path predicted in the mode-I failure of a homogeneous media with a preferred cleavage direction of 45° . (a) Polar plot of the surface energy density for different values of δ . (b) Numerically simulated crack where the black line represents the microscopic crack, and the overlaid yellow line represents the corresponding macroscopic crack direction.

respect to the crack propagation angle, $\theta' = \theta + 90^\circ$, is presented in Fig 4.11(a) for $\delta = 1, 2, 10, 100$. The analytical values of the crack propagation angle for each δ value, as estimated by the geometric method [155], is presented in the Table 4.2. Numerical simulations for crack propagation are conducted for each value of delta, and the results are presented in Fig 4.11(b), with the numerical value of the effective angle provided in table 4.2. The normalized values of σ_T^2 are also provided. The numerically predicted effective angles of propagation match well with the analytically predicted angles. It should be noted that the surface energy in the preferred direction is the same in all cases. However, it can be observed that the critical tensile stress is significantly higher for higher values of δ . There are two reasons behind this discrepancy: (1) The effective angle of the macroscopic crack is at an angle which is different from the preferred cleavage direction, and (2) the effective surface energy is overestimated due to the zig-zag nature of the crack. The former is the desired effect predicted by [155]; however, the latter is due to mesh discretization and is undesirable. To a certain degree, it can be suppressed by using adaptive meshing techniques

to iteratively generate a better-suited mesh for each test case based on the predicted crack. However, this step can lead to high computational costs.

δ	1	2	10	100
Analytical [155]	26.4°	35°	42.6°	44.8°
Numerical	28°	31°	40°	42°
$\frac{(\kappa+1)a\pi}{8\mu\gamma_0}\sigma_T^2$	2.7	3.1	5.4	30.8

Table 4.2: Crack propagation angle in mixed-mode failure

4.4.1.3 Bi-directional preference of cleavage plane

Crystalline solids usually have multiple cleavage directions, and their relative strength, along with their relative orientation to the loading direction, determines the crack path. In this section, the effect of crystal orientation on the crack path in a simplified setting is studied. Consider a material with two cleavage planes perpendicular to each other with equal cleavage energy in a mode-I test. The surface energy of the material can be chosen as follows:

$$\gamma(\theta) = \gamma_0 (1 + \delta \sin^2 (2(\theta - \omega)))$$

The physical interpretation of the parameters, γ_0 , and δ is the same as in the previous case. The preferred direction of the microscopic crack is along the vector $[\cos \omega, \sin \omega]$ and $[-\sin \omega, \cos \omega]$. The polar plot of γ_0/γ is presented in Fig 4.12(a,b) for two different values of ω . It was shown by [155] that the regions of local minimum in these polar plots have a neighborhood around them in which the crack never propagates. They referred to these regions as the forbidden regions, which they determined to be the regions of negative curvature in the polar plot. The geometric method based on Wulff's diagram is extended to such cases by determining all the points that have vertical tangents in the polar plot and then removing those in the forbidden region.

For instance, in the case of $\omega = 30^\circ$, there are two optimal crack propagation angles, while in the case of $\omega = 10^\circ$ there is only one (as shown in Fig 4.12(a,b)). The existence of such a region implies that in the presence of multiple optimal crack propagation angles with a forbidden region between them, the local orientation of the crack path rapidly switches between these optimal angles to accommodate the microscopically preferred path. The result is a saw-tooth pattern on the cracked surface. The fractured surfaces for $\omega = 30^\circ$ and $\omega = 10^\circ$ as estimated by the method in this chapter are shown in Fig 4.12(c). The saw-tooth pattern is present in the case of $\omega = 30^\circ$ but absent in $\omega = 10^\circ$ as expected.

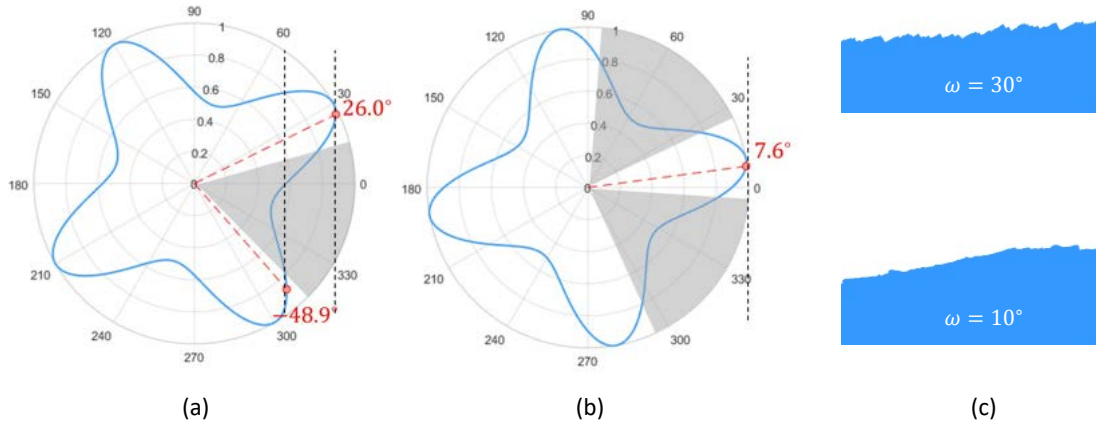


Figure 4.12: Polar plot for γ_0/γ with respect to the crack propagation angle, $\theta' = \theta - 90^\circ$ for a bi-directional surface energy with $\delta = 1$ and the crystal orientation (a) $\omega = 30^\circ$, (b) $\omega = 10^\circ$. The grey area represent the forbidden regions (c) The lower region of the cracked surface for $\omega = 30^\circ$ (top) and $\omega = 10^\circ$ (bottom)

4.4.2 Cracks in Polycrystalline material

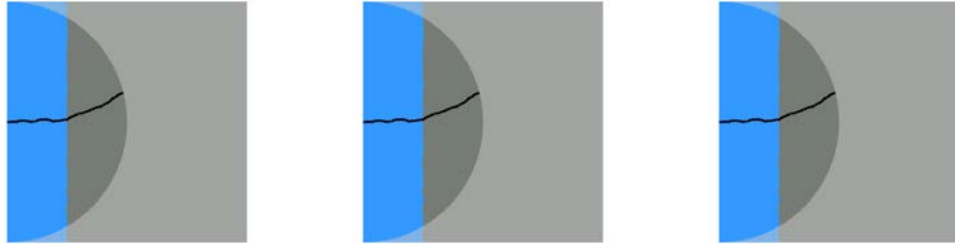
The presence of multiple grains has two effects: (1) The spatial variation of the preferred cleavage directions, and (2) intergranular fracture along grain boundaries. These effects are qualitatively analyzed in this section. Next, the effect of having non-zero surface energy is studied.

4.4.2.1 Effect of grain boundary

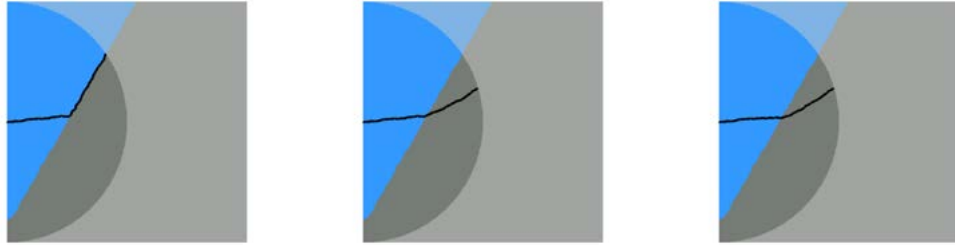
A bi-crystal in the mode-I fracture is considered in this section, with each grain having a single preferred cleavage plane. Both grains have the same parameters for energy described by Eq(4.19) with $\delta = 2$. The bicrystal's geometric construction is such that the grain boundary is a straight line passing through the point $0.25\bar{r}, 0.5\bar{r}$ at an angle, χ , to the horizontal axis. The crystal in the top left corner is labeled Grain 1, and the one in the bottom right corner is labeled as Grain 2. The orientation for Grain 1 is chosen to be $\omega = 0^\circ$, and for Grain 2 is chosen to be $\omega = 45^\circ$. Based on the analysis from section 4.4.1.2, the effective crack path angles in the single crystal of Grain 1 and 2 in mode-I fracture are $\alpha = 0^\circ$ and $\alpha \approx 35^\circ$, respectively. Numerical simulations are carried out by prescribing the surface energy of the grain-boundary, γ^{ig} , in the range: $\{0.5\gamma_0, \gamma_0, 1.5\gamma_0\}$ and the boundary angle, χ , in the range: $\{0^\circ, 30^\circ, 60^\circ, 90^\circ\}$. The results from these numerical simulations are presented in Fig 4.13.

The key observations from the numerical simulations are as follows:

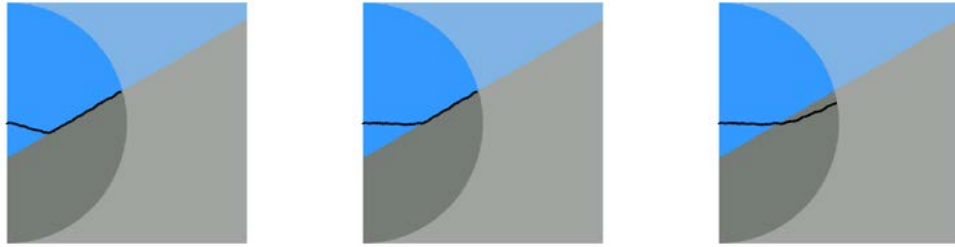
1. As γ^{ig} increases, the transgranular fracture is preferred over the intergranular fracture. This can be observed from Fig 4.13(b) where increasing γ^{ig} from $0.5\gamma_0$ to γ_0 shows the transition from transgranular to intergranular fracture.
2. The influence of the grain-boundary is more prominent when it is aligned closer to the macroscopically preferred crack path. The reason is that the crack path simultaneously minimizes the macroscopic energy release and surface energy. Therefore, the crack switches from transgranular to intergranular when the difference of surface energy compensates for the lower macroscopic energy release.
3. The grain-boundary can influence the crack path before it passes through it. This can be observed in Fig 4.13(c) with $\gamma^{ig} = 0.5\gamma_0$. The preferred crack propagation angle in Grain 1 is 0° . However, the lower value γ^{ig} influences the crack to dip downwards to lower the crack's overall surface energy. This result



(a) $\chi = 90^\circ$ with $\gamma^{ig} = 0.5\gamma_0$ (left), $\gamma^{ig} = \gamma_0$ (center), $\gamma^{ig} = 1.5\gamma_0$ (right)



(b) $\chi = 60^\circ$ with $\gamma^{ig} = 0.5\gamma_0$ (left), $\gamma^{ig} = \gamma_0$ (center), $\gamma^{ig} = 1.5\gamma_0$ (right)



(c) $\chi = 30^\circ$ with $\gamma^{ig} = 0.5\gamma_0$ (left), $\gamma^{ig} = \gamma_0$ (center), $\gamma^{ig} = 1.5\gamma_0$ (right)



(d) $\chi = 0^\circ$ with $\gamma^{ig} = 0.5\gamma_0$ (left), $\gamma^{ig} = \gamma_0$ (center), $\gamma^{ig} = 1.5\gamma_0$ (right)

Figure 4.13: Effect of grain boundary on crack propagation

is an artifact of the global energy minimization.

A similar analysis was conducted in [132], where a phase-field model was used to analyze intergranular and transgranular crack propagation in ZrB_2 bicrystal systems. It was observed that the crack path deviates from its initial trajectory before its incidence on the grain boundary. This behavior was more prominent for weaker grain boundaries. These observations are similar to the ones made in this work; however, the length-scale of these deviations is far smaller in the phase-field study.

4.4.2.2 Effect of Macroscopic length scale

The macroscopic length scale is defined based on the choice of the parameter $\bar{r}$. A smaller choice of $\bar{r}$ implies a smaller macroscopic crack. If this parameter is chosen to represent a sub-grain length scale, then the estimated crack is expected to be closer to the corresponding homogeneous medium case. In contrast, if a larger value of $\bar{r}$ is chosen, then the predicted crack path will depend on the microstructure's spatial distribution. A thin foil specimen of polycrystalline material with cubic grains under the Mode-I tensile test is analyzed as the test case. The surface energy is determined by Eq (4.10) with parameters: $\gamma_0 = 1J/mm^2$ and $\delta = 2$. The intergranular surface energy density is taken as $\gamma^{ig} = 1J/mm^2$, and the thickness of the specimen is taken as $1mm$. The crack path is estimated for 3 different values of $\bar{r}$ and the results are presented in Fig4.14. A prominently transgranular fracture in the macroscopically preferred direction is observed for the smallest value of $\bar{r}$. As the value of $\bar{r}$ is increased, the crack path converges towards a prominently intergranular fracture and moves away from the macroscopically preferred direction. It should be pointed out that this convergence is not guaranteed, for instance, in the specimens with large spatial variations in the microstructure's statistical features. For such cases, it is recommended to tune the hyperparameter, $\bar{r}$, using experimental studies.

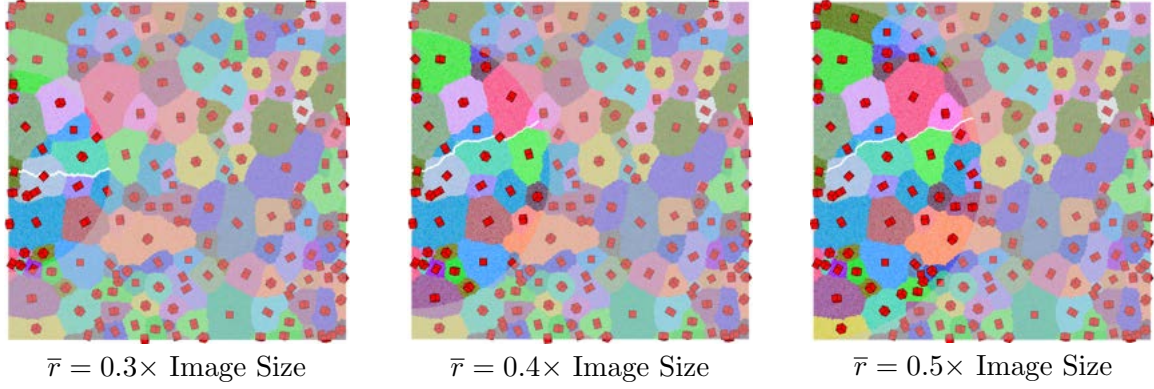


Figure 4.14: Effect of $\bar{r}$ on crack path

4.4.2.3 Dynamic fracture

In previous examples, the value of σ_T was chosen such that the total energy of the crack was almost zero (i.e., $\varepsilon(\Gamma) \rightarrow 0^-$). This was done to emulate quasistatic crack growth conditions. In this section, the effect of the non-zero total energy of the crack is studied.

The test case is a thin foil specimen of cubic material with the same energy parameters as used in Section 4.4.2.2. The crack paths are predicted for three different values of σ_T , including the critical value, representing the quasistatic case. The values of σ_T and the resulting energies are presented in Table 4.3 and the predicted crack paths are shown in Fig 4.15. It should be noted that the crack paths are dependent on the microstructure. For this particular choice of microstructure, it can be observed that: (1) The transgranular crack seems to follow the trace of $[100]$ cleavage plane as expected. (2) In the critical case, an intergranular fracture is dominant, and the crack has a non-zero macroscopic angle. This macroscopic angle of the crack decreases in magnitude with increasing σ_T . It is observed from Table 4.3 that the total energy of the crack decreases with increasing tensile stress on the specimen. However, the total surface energy of the fractured surface increases with tensile stress. This result is expected as the increase in the energy release rate shifts the competition of the microscopic surface energy and macroscopic energy release in favor of the latter

energy.

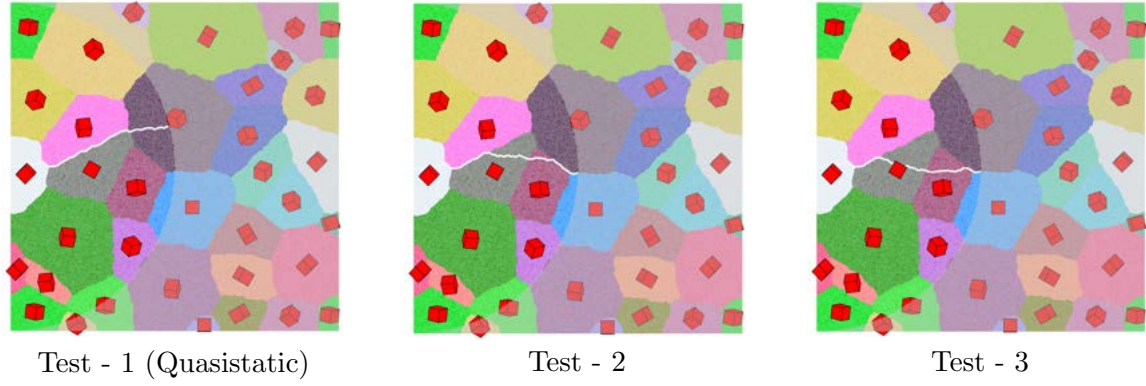


Figure 4.15: Dynamic crack propagation in a polycrystalline material

	Test - 1	Test - 2	Test - 3
$\frac{(\kappa+1)a\pi}{8\mu\gamma_0}\sigma_T^2$ (J/mm)	2.8	10.0	20.0
Microscopic surface energy (J)	1.36	1.41	1.43
Macroscopic energy release (J)	1.36	4.97	9.97
Total Energy (J)	0.00	-3.56	-8.54

Table 4.3: Crack propagation angle in mixed-mode failure

The second point in the revised ME Criterion hypothesis states that the crack propagation is possible in any loading condition, which satisfies $\varepsilon(\Gamma) < 0$. An elastodynamic extension to Griffith's fracture as presented in [156] showed that an equilibrium crack in-plane strain mode - I failure with finite velocity (v) has an augmented energy release rate ($\tilde{G}$) given as: $\tilde{G}(v, \alpha) = A(v)G(\alpha)$. In the limiting case of quasistatic fracture, the solution can be recovered from the finite velocity case as $\lim_{v \rightarrow 0^+} A(v) \rightarrow 1$. Similar results are available for Mode-II failure as well. This analysis qualitatively agrees with the hypothesis of this work. Essentially, the cracked path in Mode-I dynamic fracture is likely to follow the cracked path as determined by the macroscopic conditions. Moreover, the crack velocity can be estimated by the ratio of macroscopic energy release of the dynamic crack and the respective quasistatic case. Further

investigation is needed to verify this claim quantitatively. However, the rarity of experimental and analytical studies for microscopic dynamic growth in a polycrystalline material has prohibited the authors from doing so.

4.5 Conclusion

A graph-based approach is developed to predict the microscopic crack path during brittle fracture in a polycrystalline material. The method is based on an energy minimization principle with the crack's total energy as the cost. This approach naturally allows intergranular and transgranular fractures. The implementation presented in this work specializes in macroscopically isotropic materials; however, it can be easily extended to the anisotropic case by considering the respective energy release rate. One major drawback of this procedure is mesh-dependence as the crack is only allowed to move along the mesh's edges. The present work uses a mesh generated by Delaunay triangulation of randomly sampled points. The final placement of edges is dependent on the sampled random points. However, it is observed from numerical simulations that the crack path from two different mesh has a minimal difference if the mesh is fine enough.

The accuracy of the outlined method is verified against analytical results of quasistatic fracture in homogeneous materials with different types of surface energy forms. In the case of isotropic surface energy, it was observed that the crack path appears to be radial with crack angles similar to those described by the ME-Criterion for mixed-mode failure. In case a single cleavage direction is present, the crack path is again radial and tends to follow the direction prescribed by a Geometric construction method provided by [155]. In the case of multiple cleavage directions, a transition from a smooth radial crack to a saw-tooth pattern is observed, as suggested by [155]. The effect of grain boundary energy is also studied, and the expected transition of intergranular to transgranular fracture is observed when the surface energy of the

grain boundary is increased. While studying the effect of the grain boundary's incidence angle, it was observed that the crack path might be affected by the boundary ahead of the intergranular fracture. The reason for this effect is that when the intergranular surface energy is too low, then the global minimization of energy may lead to sub-optimal choice locally in the transgranular region. Meanwhile, the fracture's overall energy is decreased due to the generation of a longer intergranular surface. One contribution of this chapter is in developing this approach for fracture in thin foil specimens by allowing the fractured surface to have a dihedral angle. This approach allows computing the 2D surface energy from the 3D crystal in a natural way. The simulations show that the cracked surface follows the cleavage plane's trace, as seen in many experimental studies. An unintended but favorable consequence of this formulation is that it allows for dynamic fracture as well. Simulations show that cracks with higher total energy tend to have a microscopic path which is macroscopically favorable but microscopically sub-optimal. This effect qualitatively makes sense but requires a rigorous verification with experimental or analytical studies.

The objective of this work is to introduce a graph-based procedure for estimating the microscopic crack path. The modularity of this procedure allows many possible modifications to the formalism of the fracture criterion. For instance, the ME Criterion used in this method is not suited for studying fracture due to compression. A possible remedy to this problem is to use the S-criterion, which can be implemented by replacing the energy release rate (G) with the negative of the strain energy density factor as introduced by [146]. Another possible extension of this method is the crack path prediction in ductile failure. In this case, the plastic strain energy can be included by augmenting the surface energy as $\gamma + \gamma_p$, as defined in [157]. The performance of these extensions will be addressed in subsequent works.

CHAPTER V

Graph Coloring Method for Modeling Evolving Microstructures

Same materials with different microstructure may show vastly different corrosion resistance, strength, ductility, and toughness. This characterization at the microscale is usually done using the mean grain size and the grain size distribution. Consequently, studying the evolution of this microstructure through different processes is of scientific and engineering importance. Usually, the local energy at the grain boundaries is higher than the grains' corresponding bulk energies. This energy provides a thermodynamic driving force for the motion of the grain boundary. The motion of grain boundaries decreases its surface area (or length in 2D), appearing as growth in a few grains, and decay and consequent annihilation of other grains. As time increases, the total number of grains decreases, and the average grain size increases.

Computational methods are employed to study this problem at multiple length scales. At the atomistic level, methods like Density functional theory (DFT) and Molecular dynamics (MD) allow the characterization of different energies. At the microscale, computational approaches like the Monte Carlo Potts model, Surface Evolver, the front-tracking method, vertex dynamics, phase field methods and cellular automata use these energies to simulate the behavior of multiple grains. At this length scale, a typical engineering problem involves studying the motion of thousands of grains in a

representative volume element (RVE).

The normal velocity, v , of a grain boundary Γ between grain i and j , is often modelled as the multiphase flow equation [158]:

$$v = \alpha (\gamma_{ij} \kappa + (b_i - b_j)) \quad (5.1)$$

where variable γ_{ij} represents the surface energy density of Γ , κ is the curvature of Γ and b_i is the bulk energy density of the grain i . The positive scalar quantity, α , can be regarded as the mobility which relates the driving force to the velocity of grain boundary. At equilibrium (no motion of grains), this results in a supplementary condition on the intersection of three grains, known as the triple point condition:

$$\frac{\sin \theta_1}{f_{23}} = \frac{\sin \theta_2}{f_{31}} = \frac{\sin \theta_3}{f_{12}} \quad (5.2)$$

Osher and Sethian [159] showed that this motion of grains with these velocities can be modelled using an Energy minimization method. Many Level set approaches based on this minimization have been employed since, for instance, [158]. These methods are usually plagued by the problem CFL-dependent stability. In a recent work, Estellers and coworker [160] developed a Level-set based approach that preserves distance functions and removes this CFL-dependence. Energy-based labeling methods have been formalized in [161] in the context of multiphase flow. These methods can be understood in terms of Level-sets which are constant over domains rather than distance functions. These similarities allow borrowing ideas for level-set approaches in developing labeling-based methods.

In this work, first, an energy minimization principle is developed, which emulates the first-order dynamics of multiphase flow. Then, this function is modeled in a pairwise energy framework. A critical aspect of this part is the estimation of the curve's lengths within the pairwise energy framework. This problem is especially challenging

for pixel mesh where the technique of adding edges (as developed in Chapter III & IV) lead to high metrication error. For this purpose, the methodology developed by [162] is followed. They borrow results from integral geometry to estimate the expected length of curves based on the labels in the pairwise formulation on pixel meshes. This work extends their methodology to Quadtree meshes, allowing simulation of higher resolution images with a low computational cost. This framework is then generalized to incorporate anisotropic grain-growth. The ideas developed here can be directly extended to 3D microstructures, and the relevant changes to the framework are listed. The idea of smoothening developed in Chapter III is extended to voxelated meshes. This method is used to remove noise from 3D experimental images of microstructures.

5.1 Problem formulation

Before formulating the Energy minimizing problem, it is crucial to ensure that the normal velocity defined in Eq (5.7) is enough to describe the grains' motion. The following theorem shows that the tangential velocities can be ignored under certain conditions by reparameterizing the curve. This theorem first appeared in [163] and is adapted in recent texts like [164]. The reader is referred to these works for proof.

Theorem V.1. *Consider the family of curves $C(p, t)$ that solve the evolution rule, $dC/dt = u\vec{n}_{||} + v\vec{n}_{\perp}$ where $\vec{n}_{||}$ and $\vec{n}_{\perp}$ denote the tangential and normal components of the curve. The variable p parameterizes the spatial location (eg, arc-length) on the curve and t denotes the time parameter. If v does not depend on the parameterization of the curve (v is thus called an "intrinsic" or "geometric" quantity), the evolution can be converted into the solution of $dC/dt = v\vec{n}_{\perp}$, by a change of parameterization.*

Remark V.2. It is clear that the velocity described in Eq (5.7) is in terms of intrinsic parameters, and therefore, Theorem V.1 applies to this case.

With this property in place, the main idea behind the energy minimization approach

is presented next in the form of a theorem (proof in Appendix A). The theorem investigates a single grain's motion, represented by the region, Ω , inside a domain, D . This domain is treated as another grain with some bulk energy.

Theorem V.3. *Consider, $D \subseteq \mathbb{R}^2$ where D is a fixed region. And a open set $\Omega \subseteq D$ with smooth boundary, $\Gamma = \partial\Omega$. Then for any Ω , with $\bar{\Omega} \neq \phi$ or D , to be the extremizer of functional:*

$$F(\Omega) = b_1 \text{Area}(\Omega) + b_0 \text{Area}(D - \Omega) + \gamma \text{length}(\Gamma)$$

following condition is satisfied on the boundary, Γ , in the interior of domain D :

$$\gamma\kappa + b_1 - b_0 = 0 \tag{5.3}$$

Moreover, in the absence of any fixed/Dirichlet boundary conditions, Γ makes perpendicular intersections with ∂D .

Remark V.4. It can be verified from the proof that the LHS of the Eq (5.3) is the magnitude of the gradient (upto scaling) of $F(\Omega)$ and the direction of steepest descent is normal to the boundary Γ .

$$\frac{d\Gamma}{dt} \propto (\gamma\kappa + b_1 - b_0) \vec{n}_\perp$$

Remark V.5. If $\gamma < 0$, then the creation of more Γ will lower the energy. Thus the existence of a sensible solution requires that $\gamma > 0$.

Remark V.6. A multi-label generalization of this case is presented in [158].

The Global minimization of the functional $F(\Omega)$ results in the following solution:

$$\Omega = \phi \quad \text{if } b_1 > b_0$$

$$\Omega = D \quad \text{if } b_1 < b_0$$

In either case, there is no Γ and the content of the Theorem V.3 is identically true but moot. The phases should evolve, making small moves in each time step. This intuition conforms with annealing's physical process, where the resulting low-temperature solution is a polycrystalline solid with suboptimal energy (local minimum). These local moves can be modeled by adding a regularization term to the functional F , penalizing large grain boundary motions between time steps. This term is discussed in the next section.

5.1.1 Temporal evolution: First order dynamics

The region Ω , as used in Theorem V.3 can be equivalently identified using its boundary, Γ . The first order dynamics refers to the following evolution of the boundary:

$$\frac{d\Gamma}{dt} = v \vec{n} \tag{5.4}$$

where v denotes the normal velocity and $\vec{n}$ denotes the unit normal at any point on the curve. This relation is same (upto proportionality) as the motion of boundary along the direction of steepest descent as suggested in Remark V.4.

The standard L_2 measure for change in boundary is defined as follows where ' s ' is some arc-length parameter:

$$\left\| \frac{d\Gamma}{dt} \right\|^2 = \int_{\Gamma} |v_n|^2 ds \equiv \left\langle \frac{d\Gamma}{dt}, \frac{d\Gamma}{dt} \right\rangle$$

Remark V.7. The last equivalence shows that the norm is induced by the Euclidean inner product, $\langle \cdot, \cdot \rangle$

This norm is used to approximate a distance function between two contours. Consider a smooth curve, Γ_0 , where each point is displaced as $d\Gamma = v_n dt$. The point-wise

position of new curve is given as, $\Gamma(s) = \Gamma_0(s) + d\Gamma(s)$ where s is the arc-length parameterization of Γ_0 . The distance between Γ_0 and Γ can be estimated using following relation:

$$\text{dist}(\Gamma, \Gamma_0) \approx ||d\Gamma||^2 = \int_{\Gamma_0} (\Gamma(s) - \Gamma_0(s))^2 ds$$

Remark V.8. The function $\text{dist}(\Gamma, \Gamma_0)$ is well defined in the case when both represents smooth closed curves. It is easy to see that if Γ_0 is smooth and velocities are smoothly varying, then the closed curve evolves into closed curves (illustrated in Fig 5.1(a)). In case of open curves, this property may not always be true. For instance, in Fig 5.1(b), if each point on Γ_0 (black curve) move in the perpendicular direction to Γ_0 , then the curve can evolve into the green curve but not the red curve. Therefore, this function does not tell the distance between the red and the black curve. Moreover, $\text{dist}(\cdot, \cdot)$ is not a metric as this relation is not symmetric i.e. $\text{dist}(\Gamma, \Gamma_0) \neq \text{dist}(\Gamma_0, \Gamma)$. However, it has some important properties like $\text{dist}(\Gamma, \Gamma_0) \geq 0$ and $\text{dist}(\Gamma, \Gamma_0) = 0 \iff \Gamma = \Gamma_0$.

It is useful to manipulate $\text{dist}(\cdot, \cdot)$ into the following form:

$$\text{dist}(\Gamma, \Gamma_0) \approx 2 \int_{\Gamma_0} \left(\int_0^{|\Gamma(s) - \Gamma_0(s)|} p dp \right) ds \quad (5.5)$$

where variable p denotes the perpendicular distance from the curve Γ_0 . This local coordinate system is illustrated in Fig 5.1(c). Later, it will be made clear that this form is better suited for implementation in the Pairwise formulation.

With these definitions in place, following theorem (c.f. [161], proof in Appendix A) can now be used to devise a functional minimization form for simulating the first order dynamics of the mutiphase flow.

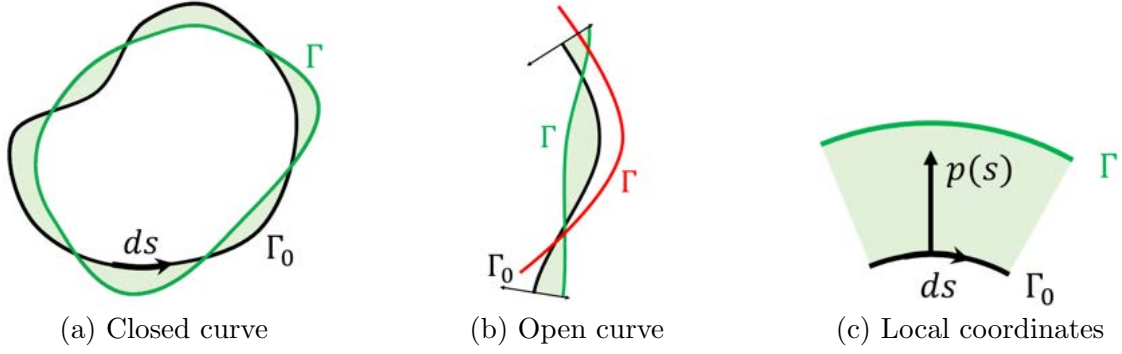


Figure 5.1: Illustration of $\text{dist}(\Gamma, \Gamma_0)$ function where the initial curve Γ_0 evolves to Γ

Theorem V.9. *Consider the following minimizer*

$$\Gamma_{t+\Delta t} = \arg \min_{\Gamma} \left(\alpha F(\Gamma) + \frac{\text{dist}(\Gamma, \Gamma_t)}{2\Delta t} \right) \quad (5.6)$$

where Γ and Γ_t are closed curves and $F(\Gamma) \equiv F(\Omega)$ with Ω denoting the interior region of the curve. It has following property

$$\Gamma_{t+\Delta t} = \Gamma_t + (v\Delta t)\vec{n}, \quad v = \alpha(\gamma\kappa + (b_1 - b_0)) \quad (5.7)$$

Remark V.10. It can be easily observed that in the limit of $\Delta t \rightarrow 0$, Eq (5.7) represents the first order dynamics, Eq (5.4).

Pairwise formulation of A pairwise formulation of the functional described in Eq (5.6), it is crucial to encode boundary lengths. In the context of random meshes (Chapters III and V), this was done simply by summing all the individual edges of partitions. Adopting this process as it is can introduce a lot more error in pixelated meshes. It is easy to see that this method estimates the L_1 -distance between any two points on the image, which severely overestimates the required L_2 -distance. Therefore, a different approach is taken to approximate the length of the curve accurately. This method is developed in the next section.

5.2 Estimation of Length

The length of a curve can be approximated using concepts from Integral geometry, specifically, Cauchy-Crofton relations. A formal discussion of these relations is out of this work's scope. The readers are referred to [165] for a complete and rigorous introduction to this topic. This work will adopt a much more applied outlook.

The main idea that is pursued here is that given a curve, and a set of lines, the total length of the curve is proportional to the number of intersections it makes with the lines (illustrated in Fig 5.2(a)) The following theorem (for proof refer to [166]) establishes this intuition as equality.

Theorem V.11 (Cauchy-Crofton Formula). *Let C be a regular plane curve with length, ℓ . The measure of the set of straight lines (counted with multiplicities) which meets C is equal to 2ℓ .*

This theorem will be the basis of the method developed later in the section. However, before going into this theorem's applications, it is necessary to define a measure on the set of lines. Consider a straight line L in the plane $\mathbb{R}^2$ determined by its normal parameters ρ and ϕ . In this parameterization, each line is described by a point in $\Xi = \{(\rho, \phi) : \rho \geq 0, \phi \in [0, 2\pi]\}$. This parameterization is illustrated in Fig 5.2. The Lebesgue measure for a set of lines $K \subseteq \Xi$ is defined as, $\mu(K) = \int_K d\rho d\phi$. This measure is, in fact, the only measure that is invariant under rigid motions of lines ([166]).

The Cauchy-Crofton formula establishes a connection between Euclidean length $|C|$ of any rectifiable curve $|C|$ in $\mathbb{R}^2$ and a measure of a set of lines intersecting it.

$$2|C|_\epsilon = \int n_c(L) d\mu(L) \quad (5.8)$$

Function $n_c(L)$ specifies a number of times any given line L intersects (see Fig 5.2).

The next step is to adopt this theory for meshed structures. The main ideas of this approach are developed in [162]. The authors developed a pairwise energy formulation that approximates the perimeter of clusters. In terms of the Multiway-Cut energy, they prescribe interaction strength w_i such that $|C| = E(\mathbf{S})$. This procedure's motivation is that it is easy to count intersections, $n_c(L)$, in the Multiway Cut framework. It can be observed that for a given choice of labels, the cardinality of the cut-set, $\mathcal{C}'$, i.e., set of edges with different node labels can be evaluated as:

$$|\mathcal{C}'| = \sum_{i=1}^{N_C} \left(1 - \delta_{s_{\pi(i,1)}, s_{\pi(i,2)}}\right)$$

Therefore the ingenuity of the method proposed in [162] is that they relate the weights, w_i , to μ . This method is described next; however, the final weights suggested here are slightly different from the original work. As will be made clear later in the text, this change is made to formulate parameters with symmetric edge weights and is a consequence of the finite difference scheme chosen here.

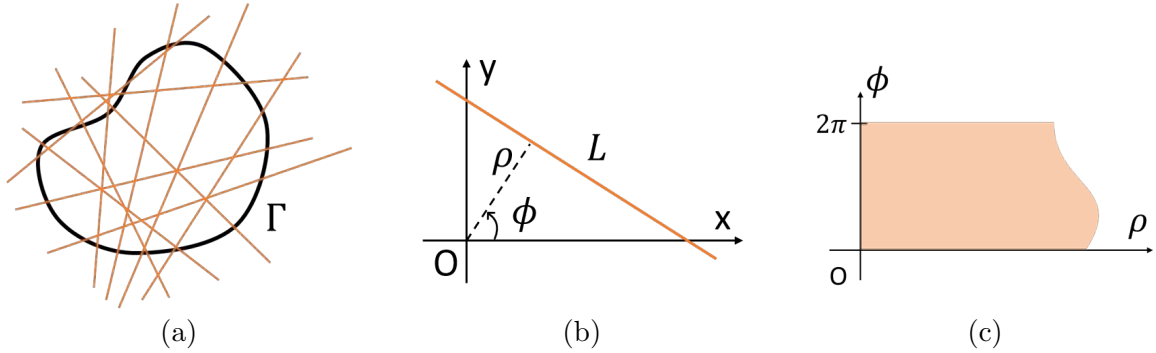


Figure 5.2: (a) Illustration for Cauchy crofton relations: $\text{length}(\Gamma) \propto$ number of intersections with lines, (b) Parameterization of a line, L , in terms of ρ and ϕ , and (c) Set, Ξ , of all lines in $\mathbb{R}^2$ represented in the (ρ, ϕ) -coordinates

5.2.1 Embedding graphs in Regular 2D grids

Every pairwise formulation starts by defining a Graph, $\mathcal{G}(\mathcal{V}, \mathcal{C})$. Each pixel in the 2D regular mesh is treated as a vertex of the graph. Consequently, each vertex of the

graph can be indexed as its respective (i, j) position on the grid. The connections between vertices are specified based on the approach described next.

It is desirable to have the following three properties for the connections :

1. Connections of each vertex are locally identical.
2. Connections describe a sparse graph structure.
3. The set of all connections can be used to describe set of all lines.

The first point is addressed by defining connections using a stencil. The idea is that if there is a connection between vertices, $v_{(0,0)}$ and $v_{(a,b)}$, then any other vertex, $v_{(i,j)}$ is connected with $v_{(i+a,j+b)}$. The readers should be aware of the slight sloppiness in this argument as it not necessary for $v_{(i+a,j+b)}$ to exist for all (i, j) . In non-periodic images, such connections are simply ignored. While in the case of periodic images, if $v_{(i+a,j+b)}$ goes out of the domain then a connection to the respective periodic node is made. For the sake of discussion, it is assumed that there is a vertex $v_{(i,j)}$ for each ordered pair (i, j) of integers. Under the above stated simplification, a stencil is chosen as a set of vectors, $\mathcal{N}_R = \{\vec{e}_i \equiv (a_i, b_i) : 1 \leq i \leq k(R)\}$. Here R defines a strategy and $k(R)$ is the number of connections in a given stencil. Consequently, this stencil defines connections between vertex, $v_{(i,j)}$ and following vertices:

$$\{v_{(i+a_\alpha, j+b_\alpha)} | (a_\alpha, b_\alpha) \in \mathcal{N}_R\}$$

The second point in the desirable properties can be addressed by choosing a stencil that contains vectors of small length. This allows connections only between vertices that are relatively closer to each other. It can be achieved by adopting a policy of only allowing stencil's vectors with length smaller than a prescribed value. For notational convenience, from here on, R as in $\mathcal{N}_R$, will denote the maximum length of the vectors contained in $\mathcal{N}_R$ i.e. if $(a, b) \in \mathcal{N}_R \implies a^2 + b^2 \leq R^2$.

Finally, the third requirement is addressed by adopting the following two policies for the stencil: (i) if $\vec{e}_i, \vec{e}_j \in \mathcal{N}_R$ then $\vec{e}_i$ and $\vec{e}_j$ are not parallel i.e. $a_i b_j \neq a_j b_i$ (ii) if $(a, b) \in \mathcal{N}_R$ then $|a|$ and $|b|$ are relative primes. Therefore, any line, L , passing through origin with a rational slope, p/q where p and q are relative primes, can be written as a union of following connections:

$$\{(v_{(iq,ip)}, v_{(iq+q,ip+p)}) | i \in \text{Integers}\}$$

The above mentioned policies avoid overlap of connections in the construction of a line. A consequence of these conditions is that if $\vec{e}_k \in \mathcal{N}_R$ then $-\vec{e}_k \notin \mathcal{N}_R$. Without loss of generality, the elements of $\mathcal{N}_R$ are chosen from 1st and 2nd quadrant. Moreover, the set $\mathcal{N}_R$ is ordered based on the angle, φ_k , of each vector, $\vec{e}_k$ from the horizontal axis. Examples of stencils that follow, these conditions are given below:

$$\mathcal{N}_1 = \{(0, 1), (1, 0)\}$$

$$\mathcal{N}_{\sqrt{2}} = \{(0, 1), (1, 1), (1, 0), (-1, 1)\}$$

$$\mathcal{N}_{\sqrt{5}} = \{(0, 1), (2, 1), (1, 1), (1, 2), (1, 0), (-1, 2), (-1, 1), (-2, 1)\}$$

$$\mathcal{N}_{\sqrt{10}} = \{(0, 1), (3, 1), (2, 1), (1, 1), (1, 2), (1, 3), (1, 0), (-1, 3), (-1, 2), (-1, 1), \\ (-2, 1), (-3, 1)\}$$

$$\mathcal{N}_{\sqrt{13}} = \{(0, 1), (3, 1), (3, 2), (2, 1), (1, 1), (1, 2), (2, 3), (1, 3), (1, 0), (-1, 3), (-2, 3), \\ (-1, 2), (-1, 1), (-2, 1), (-3, 2), (-3, 1)\}$$

It can be noted that a consequence of the conditions imposed on $\mathcal{N}_R$ is that if $R < R'$, then $\mathcal{N}_R \subseteq \mathcal{N}_{R'}$. The connections in the first three stencils are shown in Fig 5.3.

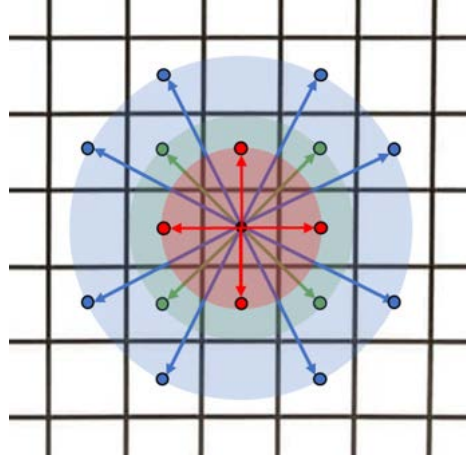


Figure 5.3: Connections in a stencil. The red region denotes the stencil, $\mathcal{N}_1$. The stencil, $\mathcal{N}_{\sqrt{2}}$ is constructed by adding connections in the green region to $\mathcal{N}_1$. The neighborhood, $\mathcal{N}_{\sqrt{5}}$ is constructed by adding the nodes in the blue region to the neighborhood, $\mathcal{N}_{\sqrt{2}}$.

5.2.2 Numerical approximation to Cauchy-Crofton relation

First, it is essential to establish a relationship between the space of all lines and the set of all edges of the graph. In the previous section, a line, L , passing through the origin and with rational slope was constructed using a set of graph connections. Next, the set of all lines parallel to this line that can be represented on the grid are considered. The next proposition reveals more information about this set of lines.

Proposition V.12. *Consider the set of all lines with rational slope p/q (p and q are relative prime) that pass through atleast one grid point (i.e a point $(a, b) \in \mathbb{I}^2$). These lines are regularly placed with following distance between two nearest lines:*

$$\Delta\varrho = \frac{1}{\sqrt{p^2 + q^2}} \quad (5.9)$$

Remark V.13. If the grid spacing is δ instead of the assumed value of 1, then the above relation can be modified as

$$\Delta\varrho = \frac{\delta}{\sqrt{p^2 + q^2}}$$

All lines that are representable by the edges of the graph can now be described as a element, $(\varrho, \varphi) \in \Xi_G$ where,

$$\Xi_G = \left\{ \left(\frac{k}{\sqrt{a^2 + b^2}}, \varphi \right) \middle| (a, b) \in \mathcal{N}_R, \tan(\phi) = a/b \in [0, \pi), k \in \mathbb{I} \right\} \quad (5.10)$$

Remark V.14. This representation does not adhere to the one used for describing the set of all lines Ξ in the Cauchy-Crofton relations. The distance parameter was restricted to the non-negative reals in the initial description. However, it is allowed to take negative values in the above notation.

This issue is resolved by adopting a variation of the latter notation, which is better suited for the numerical arguments that are being presented next. More specifically, any line L is now represented as an ordered pair $(\varrho, \varphi) \in \Xi'$ with

$$\Xi' = \{ (\varrho, \varphi) \mid \varrho \in \mathbb{R}, \varphi \in [\omega_0, \pi + \omega_0) \} \quad (5.11)$$

The reason for the arbitrary appearance of the variable, ω_0 , will be made clear later on. Nevertheless, this change is summarized as: “The slope, m , of the line is measured as $\arctan m \in [\omega_0$ and $\pi + \omega_0)$. The line’s distance from the origin is signed, and positive in the direction along the vector with angle $\pi/2 + \arctan m$ (chosen arbitrarily).”

Remark V.15. The Cauchy-Crofton relations in this notation can now be reformalized as:

$$|C| = \frac{1}{2} \int n_c d\mu = \frac{1}{2} \int_{\omega_0}^{\pi + \omega_0} \int_{-\infty}^{\infty} n_c(\varrho, \varphi) d\varrho d\varphi \quad (5.12)$$

The numerical estimation of this integral is carried out by discretizing the set, Ξ' , using a grid. This is achieved by first discretizing the space Ξ' with horizontal strips with centers at $0 = \varphi_1 < \varphi_2 < \dots < \varphi_{\mathcal{N}_R} < \pi$. The φ -axis is divided into regions, $\{[\omega_{i-1}, \omega_i)\}_{i \in \{1, \dots, |\mathcal{N}|_R\}}$ with $\omega_{|\mathcal{N}|_R} = \pi + \omega_0$. The values of ω_i are chosen as

the angle bisector of ϕ_i and ϕ_{i+1} . This relation is formalized as follows:

$$\omega_i = \begin{cases} \frac{\varphi_{|\mathcal{N}_R|} - \pi}{2} & \text{if } i = 0 \\ \frac{\varphi_{i+1} + \varphi_i}{2} & \text{if } 1 \leq i \leq |\mathcal{N}_R| - 1 \\ \frac{\varphi_{|\mathcal{N}_R|} + \pi}{2} & \text{if } i = |\mathcal{N}_R| \end{cases} \quad (5.13)$$

The size of each region containing line with angle ϕ_i , denoted as $\Delta\varphi_i$, is estimated as:

$$\Delta\varphi_i = \omega_i - \omega_{i-1} = \begin{cases} \frac{\pi + \varphi_1 + \varphi_2 - \varphi_{|\mathcal{N}_R|}}{2} & \text{if } i = 1 \\ \frac{\varphi_{i+1} - \varphi_{i-1}}{2} & \text{if } 2 \leq i \leq |\mathcal{N}_R| - 1 \\ \frac{\pi - \varphi_{|\mathcal{N}_R|} - 1}{2} & \text{if } i = |\mathcal{N}_R| \end{cases} \quad (5.14)$$

Next, the k^{th} strip is further discretized into uniformly spaced elements of length, $\Delta\varrho_k = \delta / \|e_k\| \equiv \delta / \sqrt{a_k^2 + b_k^2}$. Thus each element of $\Xi_G \subseteq \Xi'$ is treated as a node in the discretized space of Ξ' . The Cauchy-Crofton relation in this discrete space can be approximated as the following summation:

$$|C| \approx \frac{1}{2} \sum_{k=1}^{|\mathcal{N}_R|} \sum_{j \in \mathbb{I}} n_c(\varrho_j, \varphi_k) \Delta\varrho_k \Delta\varphi_k$$

where $\Delta\varrho_k \Delta\varphi_k$ is the area of the element of the grid corresponding to the lines in the direction, $\vec{e}_k$. And, $n_c(\varrho_j, \varphi_k)$ is the number of intersection that a curve makes with the line $L(\varrho_j, \varphi_k)$. Since a stencil is used to define these lines and parallel lines are given the same weight, w_k , the dependence on ϱ_j can be dropped. Consequently, the length can be written in terms of $n_c(\varphi_k) = \sum_{j \in \mathbb{I}} n_c(\varrho_j, \varphi_k)$ (i.e. intersections that the curve makes with a family of lines with slope φ_k) as:

$$|C| \approx \sum_{k=1}^{n_G} n_c(\varphi_k) \frac{\delta \Delta \varphi_i}{2||e_k||}$$

In the familiar Multiway cut framework, this is equivalent to choosing weight w_k as:

$$w_k = \frac{\delta \Delta \varphi_i}{2||e_k||} \quad (5.15)$$

The computation of flows with Pixel-meshes can be found in [161]. However, some microstructures have large grains, and the grain boundary motion is often restricted to the peripheral region of these grains. This feature can be utilized to simulate the problem in a smaller graph, where the deep interior of any grain can be treated as a single node that requires higher energy to change labels. For this reason, a Quadtree-based implementation of this theory is developed in the next section.

5.3 Computation with Quadtree structure

Quadtree structures are often used in image processing for reducing the size of an image while at the same time capturing its essential features. They operate on the idea that if many pixels in a neighborhood have similar features, they can be clustered together and saved as a single node. In the context of the microstructure image, only the pixels near the grain boundary have contrasting features. Therefore, the pixels that are further from the grain boundary can be clustered together and treated as a single node in the embedded graph. See Fig 5.4(b) for a comparison of a circular grain represented using a Pixelated and Quadtree mesh with a similar number of nodes. However, the motivation here is not the compression of data. The graph-labeling procedure is a time-consuming step, and the capability to embed large problems in smaller graphs gives a tremendous numerical advantage.

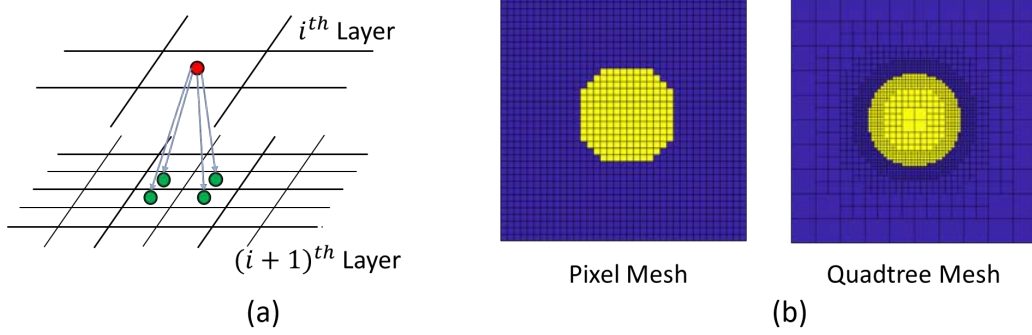


Figure 5.4: (a) Illustration shows the relationship of a pixel in parent layer and its children (b) Comparison of two circular grains represented using Pixel Mesh and Quadtree mesh with similar number of active nodes. The pixelated mesh has a discretization of 38×38 while the Quadtree mesh has 4 layers with the initial layer of size 10×10

The structure of Quadtree can be thought of as a layer of refinement on a pixelated mesh. Starting the coarsest layer on top (of discretization size $N_d \times N_d$ i.e. each direction has N_d pixels), each pixel is split into 4 pixels (2×2 grid) as illustrated in Fig 5.4. Thus the second layer has a total of $4N_d^2$ pixels. This procedure of adding children to each pixel is repeated $N_p - 1$ times where N_p is the user-defined number of pixelated layers. The pixels in the last layer, also known as the leaves, have the smallest size. The size of this smallest pixel dictates the resolution of the image. For instance, in Fig 5.4(b), even though Quadtree mesh has fewer nodes than the pixelated mesh of size 38×38 , it offers the resolution similar to a pixel mesh of size 64×64 . Next, the minimization problem in these two meshes is formalized in terms of pairwise energy.

5.3.1 Pairwise formulation for pixel mesh

The implementation of multiphase growth on pixelated-mesh can be equivalently posed as:

$$U(s_i = \ell) = \delta^2 f_l + \frac{2\delta^2}{\Delta t} \min_{\{j|s_j=\ell\}} d_{ij}$$

$$V(s_i, s_j) = \begin{cases} 0 & s_i = s_j \\ 1 & s_i \neq s_j \end{cases} \quad (5.16)$$

where ℓ represents some index of the grain with f_ℓ denoting its bulk energy density. The second term in U encodes the regularisation term. d_{ij} , refers to the L_2 -distance between nodes (pixels) i and j . The minimization step, $\min_{\{j|s_j=\ell\}} d_{ij}$ gives the distance between pixel i and the closest cluster with label ℓ . This minimum approximates the value of p described in Eq (5.5). One can also observe that if the node i is inside a cluster of label ℓ , then the regularization term evaluates to zero, this may seem to conflict with the definition of $dist(.,.)$ function, however, the total regularization evaluated as sum over all labels is still positive so there is no contradiction. The term $V(s_i, s_j)$ is the usual Multiway cut type of energy. The weights J_k are decided using the stencil. It is evident that only the field term, $U(\ell)$, need to be updated in each iteration as the regularization needs to be updated.

5.3.2 Pairwise formulation for Quadtree mesh

The quadtree structure is such that any point in the domain belongs to exactly one pixel. Depending on the layer it belongs to, this pixel may have a different size. Therefore, the pairwise cost parameters described in Eq (5.16) are valid for Quadtree mesh as well with appropriate selection of Pixel size, δ . The main challenge in implementing this procedure in Quadtree mesh is that it does not preserve the uniform structure of pixel mesh, and hence the stencils cannot be employed to write the pairwise energy, i.e., the parameter J_k cannot be determined in the manner it was done in the case of Pixel mesh. However, each layer's structure is uniform, and the interaction weights of pixels within the same layer can be easily computed using the Crofton-relations for pixel mesh using the appropriate pixel size. Therefore, only the connection weights between neighbors belonging to different layers need

to be modeled. Choosing a coarser pixel enforces that each of its children's label changes simultaneously to the same value. Thus an intuitive way to model the edge weights between a smaller and larger pixel is by summing all the connection weights between the smaller pixel and the larger pixel's descendants with the same size as the smaller pixel. In a slightly more formal tone, consider the nodes v_i^l and v_j^m where the superscript defines the layer number, and subscripts denote the index of the node. Without loss of generality, assuming $l < m$ i.e. Layer- l is coarser than Layer- m , the weight, w_{ij} , between v_i^l and v_j^m can be estimated as:

$$w_{ij} = \sum_{v_k^m \in \text{Descendants}(v_i^l)} w_{kj}$$

Unlike the usual Quadtree method where a pixel is split when it contains a boundary (Γ), in this case, the pixel is split when one of its neighbors contains a boundary. This neighborhood can be decided by looking at pixels that lie within a ball of a prescribed radius. Choosing a larger ball result in a less efficient quadtree structure. However, choosing a smaller ball can induce more error in length estimation as the neighborhood's resolution defines the discretization for the Crofton relation. The pseudocodes and recommended data structure for implementing this problem are presented in Appendix B.3.

5.4 Simulation Results

As a first test case, simulation of a pill-shaped grain is considered (see Fig 5.5). The problem's setup is similar to the one used in Theorem V.3 with no bulk energies. It is observed that the grain first reduces in length, and then once it achieves a spherical shape, it starts shrinking. This behavior is expected in this problem. It is also observed that the surface energy (also the total energy) continuously decreases, as is required.

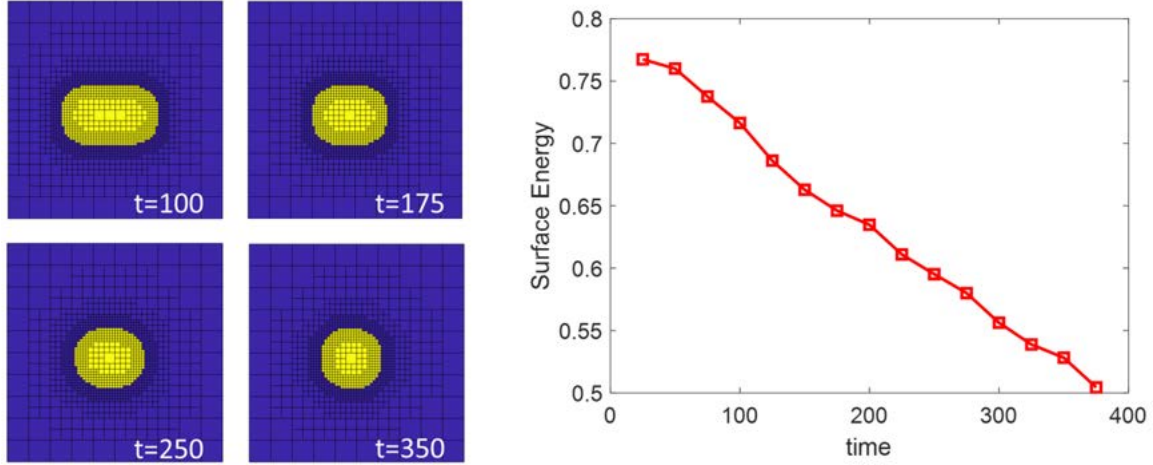


Figure 5.5: Simulation results for a pill shaped grain.

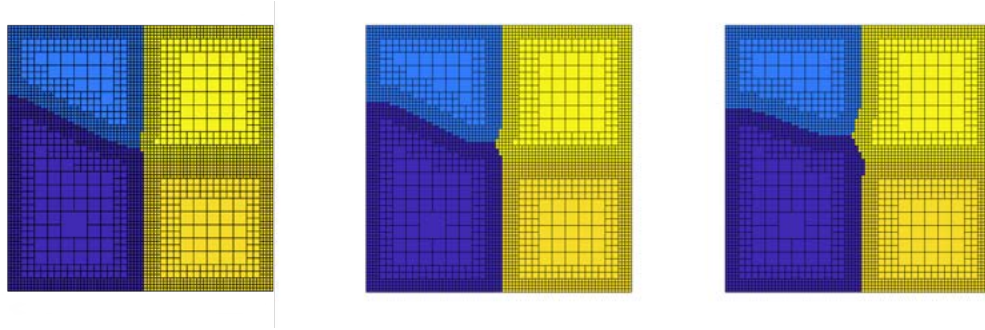


Figure 5.6: A 4-point junction splitting into 2 3-point junctions. All grains have no bulk energy density and surface energy between each pair of grain is also same.

In the next case, a 4-point junction of grains is considered (see Fig 5.6). It is known that these points are not stable, as is evident through the simulation. The initial junction splits into two triple-point junctions. Moreover, the junctions form approximately equal 120° angles between grain boundaries. This observation follows from the triple point junction's analytical solution given in Eq (5.2) with equal surface energies. Moreover, it can also be seen that the grain boundaries make a vertical intercept to the domain. This result is expected for a non-periodic domain, as suggested in Theorem V.3.

Finally, a simulation of periodic microstructure is considered in Fig 5.7. As expected, the smaller grains are annihilated as time progresses. Meanwhile, larger grains get bigger. The condition for perpendicular intercept to the boundary is not valid here

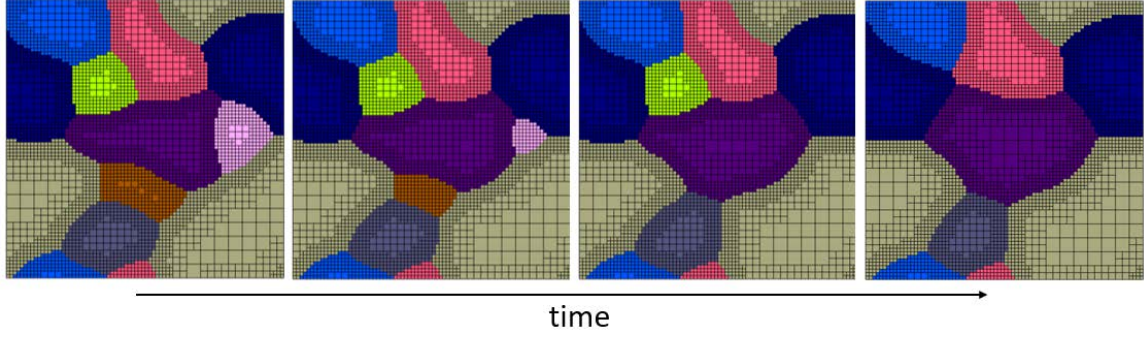


Figure 5.7: simulation of a polycrystalline microstructure

because of the periodicity.

5.5 Extensions

The method presented so far only allows growth laws represented as Eq (5.1). However, the framework developed here allows for much more general evolution laws. For instance, spatial variations in the surface energy γ can be directly modeled as suggested in [164, 161]. In this study, the first extension considers the case when surface energies are directionally-dependent or anisotropic. Such surface energies can be computed by manipulating the metric used to estimate the curve's length. Next, the ideas are extended to 3D voxelated structures. In this case, surface area minimization leads to mean curvature flow. This idea is used to develop a physics-inspired method for denoising experimental microstructure images.

5.5.1 Modelling anisotropic surface energies for non-equiaxed microstructure

In this section, the surface energy is treated as a function of the surface normal i.e. $\gamma \equiv \gamma(\vec{n})$. This better encapsulates the physics of crystal growth where energies are directionally-dependent. The minimization of the total energy (sum of bulk and surface energy) results in a new growth law. This growth law can be evaluated using the

following theorem (proof in Appendix A):

Theorem V.16. *In the setup of Theorem V.3, consider the following functional:*

$$F(\Omega) = f_1 \text{Area}(\Omega) + f_0 \text{Area}(D - \Omega) + \int_{\Gamma} \gamma(\vec{n}(s)) ds$$

where $\gamma(\vec{n})$ is the surface energy in the direction of unit normal, $\vec{n}$. The extremizer satisfies the following condition on the boundary, Γ :

$$\nabla \cdot (\gamma \vec{n} + (\nabla \gamma \cdot \vec{t}) \vec{t}) + f_1 - f_0 = 0 \quad (5.17)$$

where $\vec{n}$ and $\vec{t}$ are normal and tangents of Γ . Moreover, in the absence of any fixed/Dirichlet boundary conditions, Γ intersects ∂D at an non-zero angle such that following condition holds true:

$$\gamma \vec{n} \cdot \vec{n}_{\partial D} + (\nabla \gamma \cdot \vec{t}) \vec{t} \cdot \vec{n}_{\partial D} = 0$$

Remark V.17. It is an easy exercise to check that choosing γ as a constant simplifies the expressions to the ones in Theorem V.3. Moreover, the regularization for temporal evolution is carried out in the same way as before in Theorem V.9.

The evaluation of surface energy is still carried out using the Crofton-relations. This can be achieved by choosing a metric space, $\mathcal{M}$ such that the following relation holds for all possible boundaries Γ

$$\int_{\Gamma} \gamma(\vec{n}(s)) ds \equiv |\Gamma|_{\mathcal{M}}$$

where $|C|_{\mathcal{M}}$ denotes the length of the boundary in the given metric. [162] considered

a special case where this length is measured using a Riemannian metric as follows:

$$|C|_{\mathcal{M}} = \int_{\Gamma} \sqrt{\vec{t}^T \cdot D \cdot \vec{t}} ds \quad (5.18)$$

where a positive definite matrix D specifies the local Riemannian metric at a given point in the image and $\vec{t}$ is a unit tangent vector to the contour. In this case, the Crofton formula (Eq (5.8)) is estimated as [162]:

$$2|C|_{\mathcal{M}} = \int \frac{\det D}{2(u_L^T D u_L)^{3/2}} d\mu(L) \quad (5.19)$$

where u_L is the unit vector¹ in the direction of line L . Moreover, the weights of the stencils are estimated as:

$$w_k = \frac{e_k^T e_k \det D}{2(e_k^T D e_k)^{3/2}} \delta \Delta \phi_i \quad (5.20)$$

It should be remarked that arbitrary surface energy functions cannot be modeled in this manner. It was shown by [167], that the most general case that can be considered in this framework is when $|C|_{\mathcal{M}}$ is measured with respect to a subclass of Finsler metric. Interested readers are referred to their work for further discussion on this topic.

5.5.2 Modelling 3D grains for denoising experimental Microstructure image

Crofton formula can also be extended to estimate lengths of cures and surface area of hypersurfaces in higher dimensions. These relations can be found in the senior thesis of Sweeney [168] along with the proofs. A particularly useful corollary is for estimating the surface area of surfaces in 3D. In this case, the Crofton is modified as

¹The reference to unit tangent is made in the Euclidean metric and not in the Riemannian metric

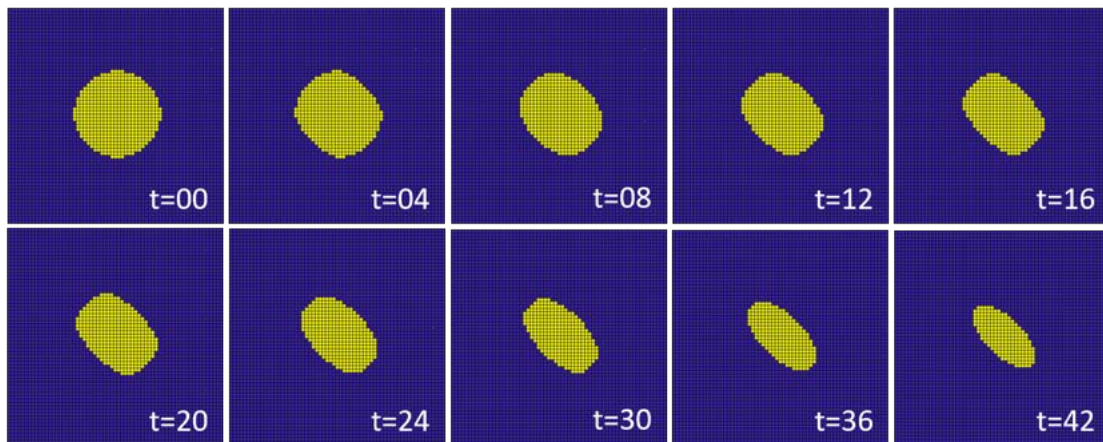


Figure 5.8: Simulation of an initially circular grain with surface energy estimated using Eq (5.18) with $D_{11} = D_{22} = 1, D_{12} = D_{21} = 0.75$.

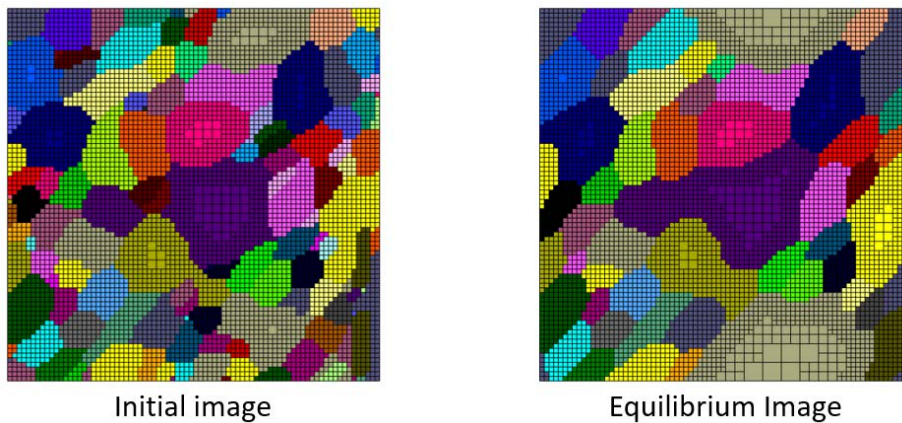


Figure 5.9: Initial and final image of a microstructure estimated using Eq (5.18) with $D_{11} = D_{22} = 2.5, D_{12} = D_{21} = -1.5$.

follows:

Theorem V.18 (3D Crofton). *Let M be a 2D surface with an area, A . The measure of the set of straight lines (counted with multiplicities) which meets M is equal to $\pi^2 A/2$.*

The numerical version of this relation can be formalized by following the same strategy to construct stencils, $\mathcal{N}_R$ in 3D. The weights of the connections are estimated as:

$$w_k = \frac{\delta^2}{\pi ||e_k||} \Delta\phi_k$$

Here, the $\Delta\phi_k$ represents the solid angle in contrast to the 2D case. It is evaluated using the strategy proposed in [169, 170]. Each vector $\vec{e}_k$ is projected onto a unit sphere. The intersecting points are then used as a germ/grain for the Voronoi tessellation of the sphere. The surface area of each spherical domain is then used as the weight for the corresponding direction.

It can be shown that the first variation of the surface area results in Euler-Lagrange equations corresponding $H\vec{n} = 0$ where H is the mean curvature given in terms of principal curvatures (maximum and minimum of the normal curvature a given point) as: $H = 0.5(\kappa_1 + \kappa_2)$. The proof of this result can be found in [164]. This property is useful in denoising 3D microstructure experimental data. These data are usually provided in voxelated images, with each voxel representing a grain identifier. Taking a similar approach as in Chapter III, the Eq (3.1) is used with the weights described by the Crofton relations based stencil. The results are presented in Fig 5.10, where the procedure shows to smoothen the data by relabeling the noisy voxels. Moreover, using a larger stencil provides better results in terms of the curvature of grains.

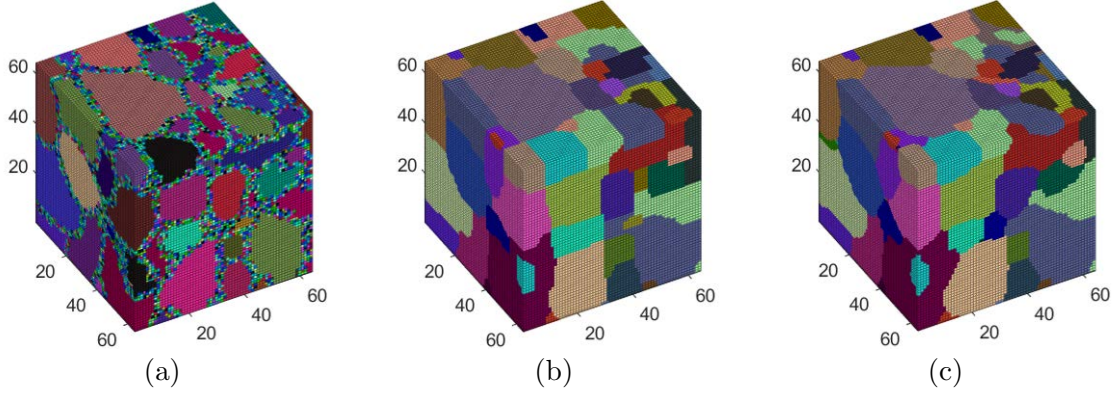


Figure 5.10: Denoising 3D microstructure data as a graph labeling problem. (a) Noisy raw data from experiments (b) Microstructure after smoothing procedure using $\mathcal{N}_1$ (c) Microstructure after smoothing procedure using $\mathcal{N}_{\sqrt{6}}$

5.6 Conclusion

An energy-based model is developed to simulate multiphase flow for studying microstructures evolution. This model is implemented in the pairwise energy form. The interaction energy formalized in this chapter is the same as the one presented in Chapter III. However, the embedded graph is non-planar. Thus this formulation is computationally more intensive than the formulation of the surface on random meshes, but at the same time, metrication errors are severely suppressed. In terms of flexibility to use different surface energy functions, the random mesh has an advantage over the pixelated mesh, limiting the surface energies that can be represented using lengths in a particular class of metric spaces.

A significant contribution of this work is to extend this framework to Quadtree mesh. The evolution of microstructures in both periodic and non-periodic domains are exhibited using examples. It is shown that the methodology captures the necessary physics of microstructure evolution in terms of annihilation and merger of grains. Moreover, the equilibrated solution matches the analytical conditions at the triple-point junctions. The evolution laws for anisotropic length metrics are identified and are used to simulate non-equiaxed microstructures. The ideas can be extended to-

wards segmentation of 3D microstructures where unlabelled grains in experimental images can be identified by minimizing data-cost and by using a regularization term for the surface energy. This procedure can also be used to remove the noisy voxels of images while preserving the overall data microstructure. Future work may include inclusion of strain energy, in addition to bulk and interfacial energy, in the formulation to model other material science problems such as growth and coarsening of precipitates within a grain [171].

This work has identified a few areas where further research is required for improving microstructure evolution models. In its present state, the surface energies that can be represented in the pairwise formulation are restricted by corresponding length metrics that can be modeled using Crofton relations. This restriction is a significant hurdle in developing a complete simulation suite for microstructure evolution where grain boundary energy is much more complicated. This work has also shown that metrication errors can be suppressed using concepts from integral geometry. Currently, there is no such framework for non-uniform meshes. Introducing such ideas, by possibly including higher-order connectivity in the graphs used in Chapter III and IV can remove the restriction of random mesh that was assumed in those cases.

CHAPTER VI

Constrained Voronoi Models For Interpreting Surface Microstructural Measurements

Measurement and analysis of microstructures is an essential aspect of materials design and structural performance. In the case of surface experimental measurements such as digital image correlation (DIC), it is beneficial to know the subsurface microstructure to interpret the surface observations accurately. However, subsurface microstructures are expensive to obtain through three-dimensional (3D) tomography. Hence, it is of interest to generate these structures computationally. In this work, a generalized inverse Voronoi problem is used to grow an approximate representation of the 3D microstructure from a surface electron backscatter diffraction (EBSD) image. The novelty of the approach is that the surface microstructure is retained during the simulation. This technique is employed for the reconstruction of a recrystallized magnesium alloy microstructure. Crystal plasticity finite element modeling (CPFEM) was employed for comparing the predicted surface strains in the reconstructed 3D microstructures against experimentally measured data. It is observed that the surface strains of different 3D reconstructions are qualitatively similar to the experiment. However, strong basal slip activation in some subsurface grains can influence the choice of activated slip systems on surface microstructures. The results show the implications of performing a full 3D crystal plasticity analysis of measured

surface data as compared to only analyzing a two-dimensional extruded microstructure.

6.1 Introduction

Microstructure reconstruction is growing to be an important topic of interest in the field of materials modeling, simulation, and design [81]. Voronoi diagram (also known as Dirichlet tessellation or Thiessen tessellation) is a popular tool for constructing polycrystalline microstructures of metallic alloys [108, 79]. Voronoi tessellation and its dual, Delaunay triangulation are often used to describe observed structures in crystallography. A historical perspective for the development of the Voronoi diagram, including its implementations in crystallography along with the recent applications in a wide variety of fields, is provided in [172]. Voronoi tessellation essentially consists of partitioning a plane into polygons whose boundaries are defined based on a set of nodes called generator points. Given these generators, the Voronoi polygons can be constructed using the algorithm described in [173]. For a two-dimensional (2D) micrograph, the Voronoi diagram is similar to a planar straight-line graph (PSLG) in which every polygon consists of the nodes that are closer to its corresponding generator point. The problem of obtaining these generators from the microstructure, referred to as the inverse Voronoi problem (IVP), has been well studied in [174, 173, 175]. While the IVP is limited to finding one generator point for every grain, its variant, generalized inverse Voronoi problem (GIVP), has been used to represent every polygon with multiple generators [176]. Such an approach is also followed in this chapter. The properties of a GIVP for a simple case of rectangular tessellation is studied in [177], where a method to find the minimal required number of generator points is described. Additionally, there are various measures used to describe deviation of grains from fully convex shapes. A measure for concavity or convexity of the

grains, referred to as percentile average relative indented surface (PARIS) factor, is introduced in [178]. The PARIS factor quantifies the difference between the actual perimeter of the grain and the perimeter of its respective convex hull, relative to the convex hull. Additionally in [179], the area difference ratio between the original shape and its convex envelope is captured by a descriptor, denoted as *deltA*, for analyzing faults in rocks. It is to be noted that contrary to the PARIS factor, *deltA* is sensitive to the tightness of the indenting angles. For very small angles such as a crack-tip or a propagating deformation twin within a grain, *deltA* approaches 0%, while the PARIS factor may remain significantly larger.

Kinematic measurements at the microstructural scale are performed using atomic force microscopy (AFM) or scanning electron microscopy (SEM). These imaging methods allow for the mapping of strain fields at the surface of polycrystals with high spatial resolutions [180]. However, in the case of surface experimental measurements such as digital image correlation (DIC), it is important to know the subsurface effects to interpret the surface observations more accurately [181]. Yet, subsurface information is expensive to obtain via non-destructive 3D tomography methods such as diffraction contrast tomography (DCT) and high-energy X-ray diffraction microscopy (HEDM) as well as destructive methods such as serial sectioning, which can only provide post-deformation data. Hence, it is of interest to numerically generate initial microstructures for computational modeling. Previously, for a better interpretation of surface experimental techniques such as DIC and orientation imaging microscopy (OIM), three-dimensional (3D) finite element modeling has been employed by using columnar grains [182]. Alternatively, in [79], the grains are approximated as a set of optimally packed ellipsoids, and the 3D microstructure is obtained by transferring the crystallized structure to a Voronoi tessellation. Morphological models are also developed for 3D concrete microstructures using multi-scale Poisson polyhedra in [183].

Recently, several methods based on geometric and feature matching algorithms [77, 184, 185, 35, 36] have been developed to reconstruct 3D microstructures. However, when these methods generate the 3D microstructures, the surface microstructural information is not retained. An alternative approach used in this chapter, is to construct the 3D microstructure using Voronoi sampling in the third dimension while maintaining the experimental surface image. The microstructure of interest in this chapter underwent dynamic recrystallization, resulting in fairly equiaxed microstructures with a convex topology [186, 187]. As a result, the original surface SEM microstructure is first convexified, where internal twinned facets are removed from the 2D micrograph. Several sub-sections of a large representative EBSD map are then randomly extracted and successively stacked beneath the convexified 2D measurement. This is followed by perturbation of the Voronoi generators in the stacking direction using a normal random distribution. The resulting 3D structure is convexified once again to remove any artifacts penetrating onto the measured 2D surface. This approach faces difficulties when modeling the transition region between the surface microstructure and synthetic 3D morphology, leading to surface artifacts (e.g., subsurface 3D grains protruding on the surface) or unrealistic subsurface grains (e.g., concave grains, very small grains needed to close gaps in the transition zone). In [188], this issue is addressed by expanding or eroding each surface grain along the interior using a systematic process to generate the first layer of the 3D grains, followed by Voronoi seeding for the next layers. Yet, it is worth noting that such a procedure leads to unacceptable grain morphologies just underneath the first layer. As a result, here in this chapter, an iterative convexifying process is carried out to improve grain quality. In [189], the transition zone is rectified by annealing the microstructure using a standard 3D Monte Carlo (MC) grain growth model, where the voxels in the surface layer are fixed and not allowed to evolve while the subsurface grain topology is energetically optimized. However, such an approach requires considerable computational effort. In

the presented work, a novel GIVP approach is employed for the 3D reconstruction of polycrystalline microstructures. The approach leads to a single convex hull for each crystal that may overlap with its neighboring grains. The resulting microstructure is subsequently convexified by using a process that eliminates the intersection regions systematically.

The chapter is organized as follows: first, within Section 6.2.1, an overview of Voronoi 2D diagrams is described. Next in Section 6.2.2, the process of eliminating the overlapping granular regions through a convexification algorithm is explained for a simple 2D micrograph. Afterward, the process of obtaining Voronoi generator points from the micrograph (i.e., GIVP) is discussed in Section 6.2.3. This is followed by Section 6.2.4, outlining the algorithm for successively stacking 2D experimental EBSD images and constructing 3D microstructures, while retaining the experimental surface measurements. Subsequently, the implementation of the algorithm for obtaining a convexified 3D microstructure is discussed in Section 6.2.5. Next in Section 6.3, an overview of the SEM-DIC experiment, as well as the crystal plasticity finite element method (CPFEM) [190] is provided. Additionally, comparisons are made with surface DIC measurements for random samples of 3D microstructures as compared against the columnar 2D microstructures.

6.2 Methodology

This section begins with formally defining the Voronoi tessellation and the generator points following the notation used in [191].

6.2.1 2D Voronoi Diagram

Let $P(\mathbf{x})$ denote a point in the N -dimensional Euclidean space $\mathcal{R}^N$, where $\mathbf{x}$ is an N -dimensional vector (i.e., $\mathbf{x} = (x^1, x^2, \dots, x^N)$). Given n distinct generator points (e.g., $P_1(\mathbf{x}_1), P_2(\mathbf{x}_2), \dots, P_n(\mathbf{x}_n)$) in $\mathcal{R}^N$, the 2D surface can be partitioned to polygons, as

follows:

$$Q_i = \underset{j:j \neq i}{\{ \mathbf{x} \in \mathcal{R}^N \mid \|\mathbf{x} - \mathbf{x}_i\|_2 < \|\mathbf{x} - \mathbf{x}_j\|_2 \}} \quad (6.1)$$

where Q_i denotes a set of points in $\mathcal{R}^N$ that are closer to $P_i(\mathbf{x}_i)$ than to any other $P_j(\mathbf{x}_j)$ for $i \neq j$. Here, $\|\cdot\|_2$ denotes the Euclidean (i.e., L_2 norm) distance. Hence, each polygon in the diagram contains points that are closer to its center as compared to the center of any other polygon. This partitioning, as shown in Figure 6.1(a), is called the Voronoi tessellation for given n generator points (i.e., $P_i(\mathbf{x}_i)$ where $i = \{1, 2, \dots, n\}$). The problem of detecting Voronoi polygons is equivalent to finding Voronoi centers, if they exist. If however, the Voronoi centers do not exist (i.e., the polygons are not Voronoi polygons), the points that can best approximate the Voronoi centers are useful to locate [173].

It is to be noted that two corollaries can follow from Equation (6.1): (i) Voronoi polygons are convex in topology, and (ii) the line segment joining centers of adjacent polygons is perpendicularly bisected by the boundary line, common to both polygons. In a 2D diagram, the generator points can be obtained from the Voronoi tessellation using a purely geometrical method that is based on the property of the Voronoi diagram. In Figure 6.1(a), the points P_1 , P_2 , and P_3 are generators, while V_1 denotes a Voronoi point which is the circumcenter of the triangle $\Delta P_1 P_2 P_3$. Here, V_2 , V_3 , and V_4 are the neighboring Voronoi points. Using above denotation, let $\angle V_2 V_1 V_3 = \alpha$. Consequently, the $\angle P_1 P_3 P_2$ can be calculated as follows:

$$\angle P_1 P_3 P_2 = \pi - \alpha \quad (6.2)$$

Accordingly, from the inscribed angle theorem, the angle formed at the center of the

circle by lines originating from two points on the circle's circumference is double the angle formed on the circumference of the circle by lines originating from the same points (i.e., $\angle P_1 V_1 P_2 = 2(\pi - \alpha)$). Hence, given that the line $\overline{V_1 V_4}$ is bisecting the segment $\overline{P_1 P_2}$, one can calculate the angles $\angle P_1 V_1 V_4$ and $\angle P_2 V_1 V_4$ as follows:

$$\angle P_1 V_1 V_4 = \angle P_2 V_1 V_4 = \pi - \alpha \quad (6.3)$$

Therefore, given an exact Voronoi diagram, a generator can be determined by the intersection of rays such as r_1 and r_2 shown in Figure 6.1(b). These rays are emanating from the endpoints of a Voronoi edge, V_1 and V_4 , respectively. Once the generator point P_1 of a Voronoi region G_1 is obtained, the generators of the regions sharing a common edge with G_1 can also be found. These generators can be interpreted as the mirror images of P_1 with respect to the Voronoi edges bounding G_1 . It is proved in [174] that a proper convex plane tessellation, all of whose vertices have degree three, is

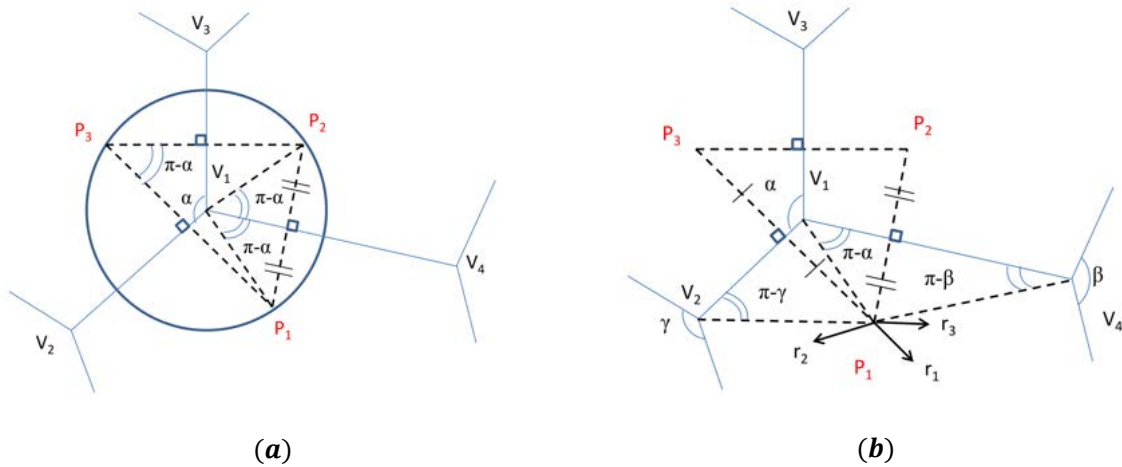


Figure 6.1: (a) Voronoi diagram with the generators and Voronoi points, (b) geometrical method to obtain the generator points from the Voronoi diagram.

6.2.2 Convex 2D Microstructure

During the recrystallization process, the cells grow in space, leading to near-convex microstructures. The microstructures that are used in this study underwent dynamic recrystallization due to a hot rolling process, resulting in fairly equiaxed microstructures. Recrystallized microstructures grow from a nucleus and form an near-convex topology. In this chapter, an age-hardened Mg-rare earth alloy WE43-T5, provided in the hot-rolled and aged condition with no intermediate solution treatment, is used. Since the microstructures are almost convex, it would be suitable to construct the equivalent convex microstructure before obtaining the Voronoi generators. It is to be noted that every Voronoi diagram produces convex polygons, but the converse does not always hold true.

Figure 6.2 outlines the process of obtaining the 2D convex microstructure from the original micrograph. Figure 6.2(a) shows the original microstructure, which is obtained in a rasterized format. The convex hulls of individual grains are constructed from the original microstructure using the *MATLAB* function `convhulln`, which uses the popular Qhull method. Figures 6.2(a)-(b) show an internal facet *A*, which corresponds to a twinned region. It can be observed from Figure 6.2(b) that there are many intersecting regions shared by two or three grains that need to be assigned to a unique grain in order to obtain a convex microstructure. The procedure to sort the overlapping region is as follows:

- 1) In 2D space, the convex hull is represented by a sequence of boundary points, as indicated in Figure 6.3(a). Connecting these boundary points results in the convex polygon. By removing a particular point, the remaining points can still be connected to form convex polygons.
- 2) The points of a polygon that are in other polygons are removed, and the intersecting points of the two polygons are included in both cells to form convex grains again. This procedure can be followed for any two intersecting polygons, resulting in the same

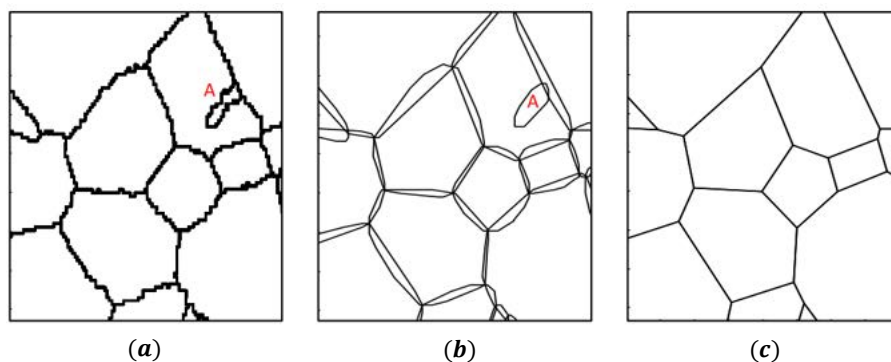


Figure 6.2: Method to obtain the 2D convex microstructure: (a) the original experimental microstructure (grain *A* represents a twinned region, which is an internal facet of the parent grain), (b) the convex hulls of individual grains, (c) the resulting convex microstructure after dividing the intersecting regions among the individual grains.

Figure 6.3 shows the procedure mentioned, applied to grains 1, 2, and 3. Figure 6.3(a) shows the grains with their original convex hulls along with their corresponding boundary points. Step 2 is applied to grain 1, resulting in Figure 6.3(b). Same procedure is again applied to grain 2, resulting in Figure 6.3(c). It is to be seen that after the procedure, every two neighboring grains share an edge with two common points, and every three neighboring grains share a triple point. It can be shown that Step 2 followed in any sequence among the three grains gives the same resulting microstructure. Figure 6.2(c) shows the procedure applied to the entire microstructure, resulting in a fully convexified topology. It is to be noted that since the region *A* is an internal facet, it will not be represented in the final microstructure. Additionally, the work in this chapter is targeted toward validation of the reconstruction method in a Mg–rare earth alloy at low strains, where experimental evidence indicates low propensity for twinning.

6.2.3 Obtaining Voronoi Generators for 2D Convex Microstructures

Once the convex 2D microstructure is generated, Equation (6.3) is used to check whether a unique generator point is achievable for each grain. Figure 6.4(a) shows

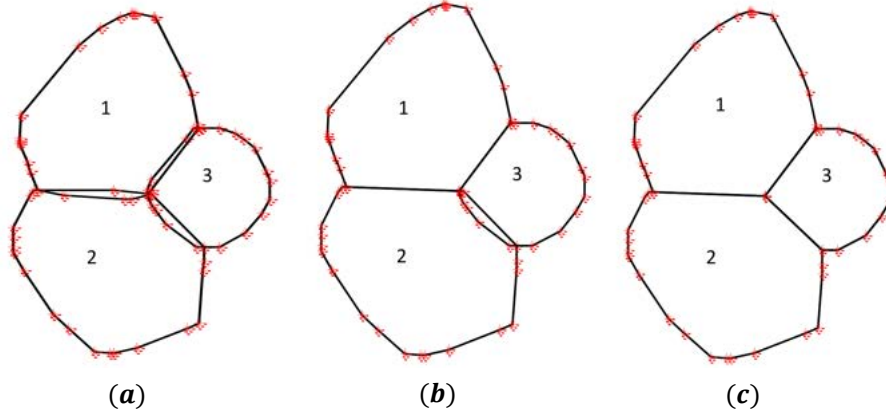


Figure 6.3: Method to eliminate the intersecting regions of the convex hulls: (a) grains 1, 2, and 3 with their overlapping convex hulls, (b) Step 2 is followed for grains 1 and 2, (c) Step 2 is followed for grain 3.

grain A with 6 edges and 6 vertices, respectively surrounded by the neighboring grains B, C, D, E, F , and G . The procedure mentioned earlier in Figure 6.1(b) is employed to construct the Voronoi generators A_1, A_2, A_3, A_4, A_5 , and A_6 for grain A as well as the corresponding generator points for the neighboring grains. Here, rays r_1, r_2, r_3, r_4, r_5 , and r_6 are constructed by connecting the generator points A_1, A_2, A_3, A_4, A_5 , and A_6 to their respective vertices. It can be seen from Figure 6.4(a) that the rays do not converge to a single point as in case for a Voronoi diagram.

Since the aim is to represent the convex microstructure with Voronoi generators, multiple generators are used to represent each grain as opposed to a single generator point. Here, each grain is represented by as many Voronoi generators as the number of vertices of the grain. Figure 6.4(b) illustrates the Voronoi polygons corresponding to each grain using dashed lines. For regular-shaped convex grains, the microstructure can be represented by placing Voronoi generators along the vertices or triple-points of the microstructure. In this chapter, the same radius of circumcircle r is used for obtaining generator points at all the vertices of the grain.

The method is also anomaly applicable for long slender grains, as shown in Figure 6.5. Figure 6.5(a) depicts the inverse Voronoi representation of grains A, B , and C . It can

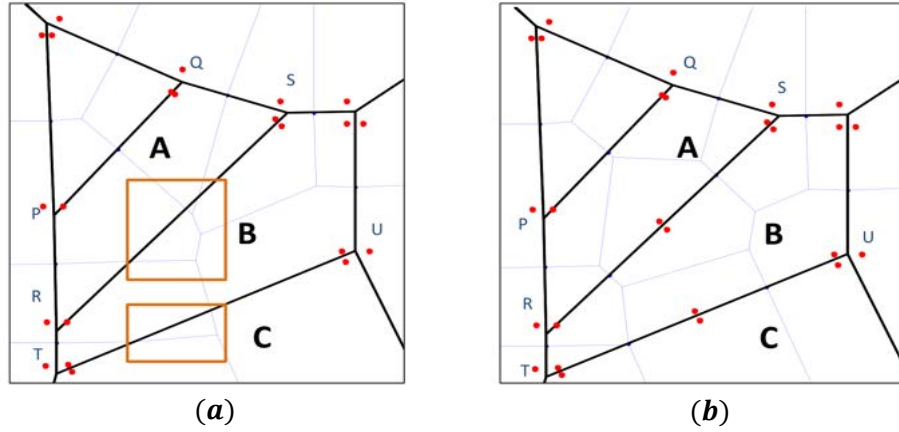


Figure 6.5: In the case of slender grains, Voronoi regions corresponding to some of the generator points can extrude into the neighboring grains across the long edges. (a) Voronoi regions of slender grains A and B extrude into grains B and C as indicated by the boxed regions. (b) Additional generator points are placed along the midpoint of edges $\overline{RS}$ and $\overline{TU}$ to bound the Voronoi regions within the grain.

6.2.4 Construction of 3D Microstructure Using Voronoi Generators

Once the Voronoi generators are identified for the 2D micrograph, a 3D microstructure can be readily created by adding Voronoi generator points in the z -direction. Here, a large surface observation such as an EBSD map is used to find the generators and their representative orientations. As a result, multiple sections of the EBSD map are randomly selected and stacked up beneath the measured surface image, as illustrated in Figure 6.6, in the z -direction. The following assumptions are made while generating the 3D microstructure:

- 1) The grains are equiaxed. With no information about the grain shape in the z -direction, it would be useful to assume equiaxed morphology for each grain. Therefore, the grains in layers with a gap-width of grain size d , as it is apparent from the 2D observation, are stacked above one another. Here, the microstructural images are stacked by a spacing of size d to mimic an equiaxial microstructure in the z -direction. Decreasing or increasing the value would increase anisotropy in the z -direction.
- 2) The surface EBSD map is representative of the microstructure in the $x - y$ plane across the depth of the sample. That is, the surface orientation distributions are

representative of the bulk orientation distributions, which is true in the case of rolled textures where ND direction represents the z -direction as well as random textures arising out of solidification processes.

3) The grains in the sections of the EBSD map are represented by their centroid as Voronoi generators. It is assumed that the overall texture of the 3D microstructure does not change considerably by this assumption. Alternatively, one could convexify the sections and represent them by Voronoi points as discussed in Sections 6.2.2 and 6.2.3.

To represent the distribution of Voronoi points in the z -direction, the generator points along each section are perturbed within the z -direction by a normal distribution with a standard deviation equivalent to the mean grain diameter (i.e., $\sigma = d/2$). The scanned microstructure of interest in this chapter comprises of grains with average grain size of $d = 13 \mu m$ with a standard deviation of about $6.5 \mu m$. Alternatively, for a more general case where the distribution of grain diameters is unknown, the Voronoi generators can be perturbed by $\pm d/2$ along the z -direction using a uniform random distribution. Additionally, to account for the tilt and twist of the grain boundaries, the Voronoi generators of the topmost layer are perturbed in the z -direction by a normal distribution with a standard deviation that is equal to the radius of the circumcircle (i.e., $\sigma = r$). Here, the standard deviation is chosen such that the generator points still lie close to the surface of the microstructure, while providing tilt and twist of the grain boundaries. The Voronoi diagram is recreated in 3D with the resultant Voronoi points. Figure 6.6 illustrates a schematic representation of the construction, resulting in a representative 3D microstructure.

6.2.5 Convexification of 3D Microstructure

It can be observed from Figure 6.6(rightmost) that the grains in the top layer are penetrated by the Voronoi generator points beneath the surface. Hence, a convexifi-

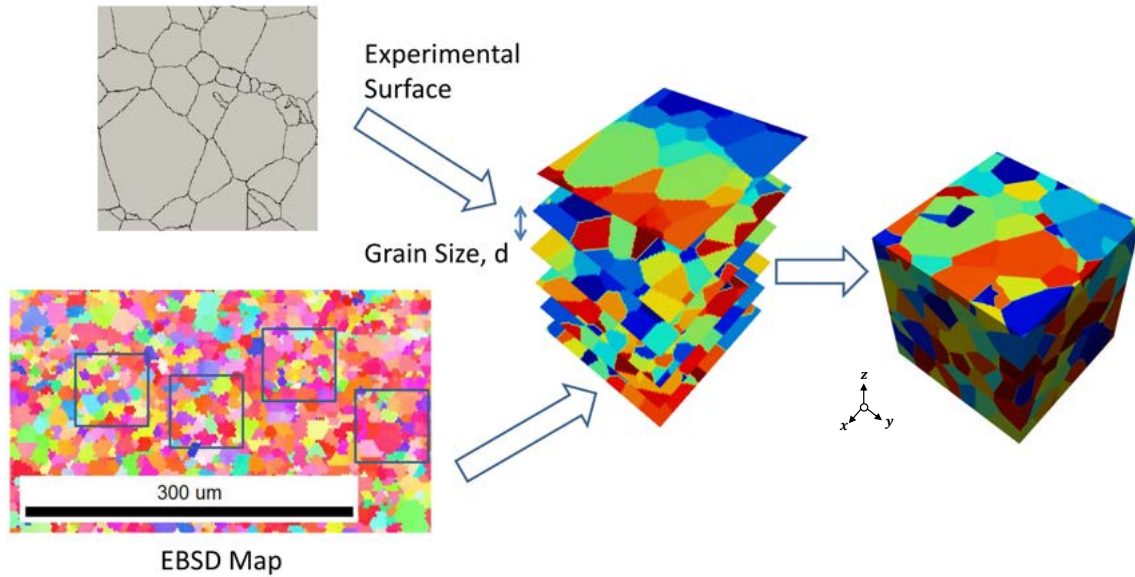


Figure 6.6: Schematic diagram showing the construction of 3D microstructure from the given surface observation and EBSD map. Sections of the EBSD map are randomly chosen and stacked below the microstructure using their centroid as Voronoi generators.

cation approach is implemented to recover the original surface microstructure. The property of convex hulls is employed, such that every section of a 3D convex polygon results in a 2D convex shape. Since 2D convex diagrams are obtained with the surface image, the surface microstructure can be fully recovered by convexifying the entire 3D microstructure. The problem of slender grains mentioned in Section 6.2.3 is also solved by convexifying the entire 3D microstructure. The procedure for convexification of the 3D microstructure is similar to the process of convexification, followed in Section 6.2.2. The modified 3D approach is described below for creation of the 3D solid microstructure:

- 1) The convex hulls of individual grains are recreated using the *MATLAB* function `convhulln`.
- 2) The points of a polygon that are penetrated in other polygons are eliminated, as shown in Figure 6.7(a). Instead of finding the intersecting planes of the convex hulls, the convex hulls of the grains are reconstructed using the remaining vertices,

as illustrated in Figure 6.7(b).

3) The gaps between the crystals are assigned to the nearest grain using a linear least-squares solver. The problem of finding the smallest distance of a given voxel to the grain can be described as:

$$\min_x \frac{1}{2} \|x - v\|_2^2 \quad Ax \leq b \quad (6.4)$$

where v corresponds to the voxel to be assigned, and $Ax = b$ represents the planes enclosing each grain. Here, the voxel v is assigned to the nearest grain in the Euclidean norm sense. Figure 6.7(b) shows the partition between the grains by assigning each of the overlapping voxels to the nearest grain as described. The regions near the edges of the grain are assigned in a non-planar way, as indicated in Figure 6.7(b). Such

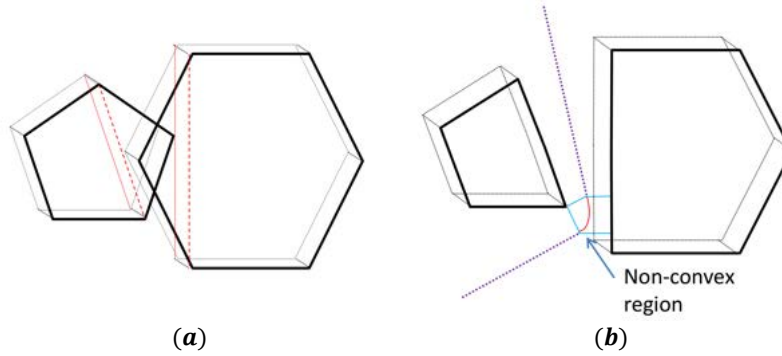


Figure 6.7: Method to convexify the 3D microstructure: (a) the intersecting points of the convex hulls of the grains are removed; (b) the voxels within the gap between the grains are reassigned to its nearest grain using a linear least-square solver, the non-convex region of the partition is indicated.

Figure 6.8 illustrates the initial 3D microstructure and the resulting convexified structure. Figure 6.9 shows the top three layers of the 3D microstructure before and after the convexification process. It can be observed that convexification affects only the top two layers, whereas the layers underneath remain convexified due to the nature

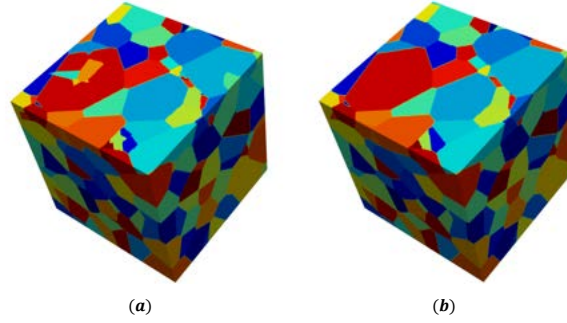


Figure 6.8: Convexification of 3D microstructure: (a) original microstructure generated by stacking up the Voronoi generators, (b) the convexified 3D microstructure.

of the V

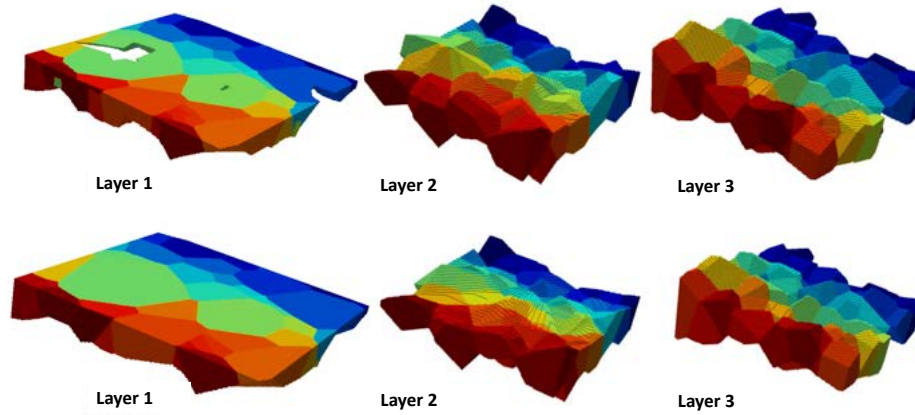


Figure 6.9: The First three layers of the 3D microstructure before convexification are shown on the top. The same three layers of the microstructure after convexification procedure are depicted below. Here, each color represents a different grain ID.

6.3 Results and Discussion

The above algorithms are implemented on an experimental microstructure obtained from SEM–DIC technique for WE43–T5 alloy. The surface strains are measured using SEM by tracking nanoparticles embedded in the alloy. The details of the experimental procedure are described in [187, 186]. The measured microstructure is of dimensions $75\mu m \times 75\mu m$ and is previously shown in Figure 6.6. The microstructure comprises of 49 grains initially. However, after the process of 2D convexification as described in

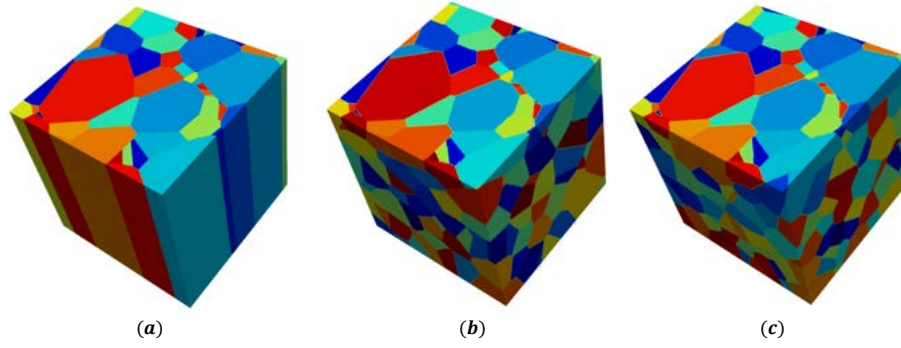


Figure 6.10: Three microstructures considered in this chapter are (a) convexified columnar microstructure with 46 grains, (b) variant 1 with 340 grains, and (c) variant 2 with 367 grains. Here, different stacks of EBSD images are used to build the variants 1 and 2. Additionally, for each microstructure, the perturbation of generator points along z -direction are distinct due to it being assigned randomly.

Section 6.2.2, the number of grains shrinks to 46, as the twinned regions such as one shown in Figure 6.2(a) are removed during the 2D convexification process.

The surface EBSD map, as obtained from experiments, is of dimensions $600\mu m \times 700\mu m$ and comprises of 6000 grains. The average grain size of the 2D EBSD map is found to be $d = 13 \mu m$ with a standard deviation of about $6.5 \mu m$. The procedure mentioned in Section 6.2.4 is followed by selecting five random windows from the surface EBSD image with dimensions $75\mu m \times 75\mu m$, with some windows identified in Figure 6.6. These collections of grains are subsequently stacked below the original experimental microstructure using a spacing of grain size d and perturbed in the z -direction by a normal distribution with a standard deviation of $\sigma = d/2 = 6.5\mu m$. The resulting microstructure is then convexified to obtain the 3D microstructural realizations as shown in Figures 6.10(b)-(c). These are referred to as the variants 1 and 2 of the 3D microstructure. The variant 1 comprises of 340 grains, whereas the variant 2 comprises of 367 grains. The constructions of these two variants are made by using different windows from the EBSD map followed by a unique random perturbation of the generator points along z -direction. Here, the reconstruction simulations are performed on a single 2.6 GHz Intel Core i5 processor machine. The

total computational time for generation of Variants 1 and 2 with a $100 \times 100 \times 100$ voxel resolution was in the order of 5000 – 7000 seconds (i.e. about 2 hours). A lower $50 \times 50 \times 50$ resolution can significantly decrease the total computational cost to the order of 1100 – 1400 seconds (i.e., about 20 minutes). Furthermore, a parallel processing implementation of such an algorithm can reduce the run-time and help to solve larger reconstruction problems, such an implementation is to be considered for future studies.

Additionally, two more examples were generated to demonstrate the methodology of obtaining convexified 3D microstructures. They are added as a supplementary material in Appendix A.

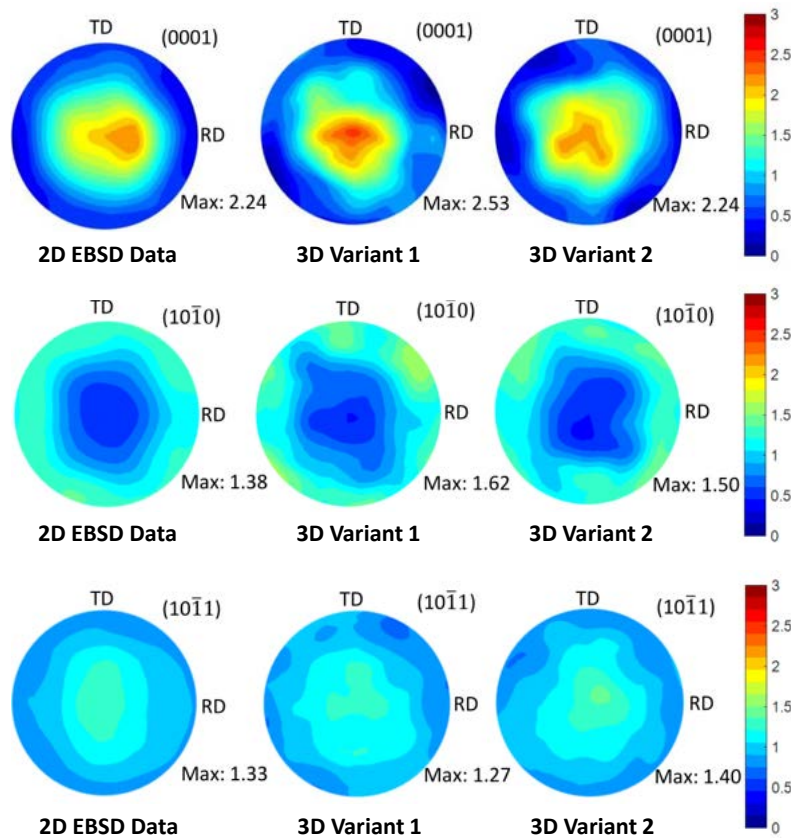


Figure 6.11: The recalculated pole figures for 2D EBSD data as well as 3D variants 1 and 2 in (0001), $(10\bar{1}0)$, and $(10\bar{1}1)$ directions are shown, respectively.

The pole figures corresponding to (0001), $(10\bar{1}0)$, and $(10\bar{1}1)$ for the resultant 3D

microstructures are plotted in Figure 6.11. It can be inferred from the plots that the texture of the variants 1 and 2 is on par with the texture from experiment. This is naturally the case with the reconstruction procedure since it samples sections from the surface EBSD scans to generate the 3D microstructure. The sampling determines the uncertainty range in pole figures [192] as seen in the 3D reconstructions.

Using these microstructures, a 3D CPFE PRISMS–Plasticity code [190] based on a fully implicit implementation of an elastoplastic single crystal model is used to capture strains under loading. Here, 8 noded linear brick elements are used to discretize the volume. The CPFE simulations use between 60 and 200 cores depending on available resources in the shared cluster. Computational details including solve and assembly time for the PRISMS–Plasticity code can be found in [190]. Within the simulations, the x and y displacements from experiments are applied along all the lateral surfaces (similar to the measured experimental data on the surface). Furthermore, the top and bottom surfaces are traction–free with z displacement constrained at the origin. Crystallographic slip, twinning, and reorientation of crystals are assumed to be the primary mechanisms of plastic deformation. For a material with $\alpha = 1, \dots, N$ slip/twin systems; the resistances $s^\alpha(t) > 0$ offered by the slip/twin systems at time t depends on the plastic shearing rates $\dot{\gamma}^\alpha$ on systems α through a hardening law:

$$\dot{s}^\alpha(t) = \sum_{\beta=1}^N h^{\alpha\beta} \|\dot{\gamma}^\beta\| \quad (6.5)$$

where,

$$\begin{aligned} s^\alpha(t=0) &= s_o^\alpha \\ h^{\alpha\beta} &= h_o^\beta (q + (1-q)\delta^{\alpha\beta}) \left(1 - \frac{s^\beta}{s_{sat}^\beta}\right) \end{aligned} \quad (6.6)$$

where the slip system hardening term, denoted as $h^{\alpha\beta}$, includes latent hardening through parameter q , taken to be 1 for coplanar slip systems and 1.4 for non-coplanar slip systems [193]. The parameter s_o^α is the initial critical resolved shear stress for system α , and h_o^β and s_{sat}^β are the hardening rate and saturation resistance of slip system β , respectively. These three parameters are calibrated for each slip system such that the CPFEE model predicts the experimental stress-strain response under different deformation modes. Other aspects of the model, including elastoplastic decomposition, flow rule, and constitutive update algorithm, are detailed in [190]. The volume fraction associated with each twinning system, denoted as g^{n,t_i} , increases with loading, and the fraction at quadrature point n at any given loading step is given by:

$$g^{n,t_i} = \sum_{steps} \Delta g^{n,t_i} \quad (6.7)$$

where,

$$\Delta g^{n,t_i} = \frac{\Delta \gamma^{n,t_i}}{S} \quad (6.8)$$

where t_i denotes the twinning system i , S is the characteristic shear strain of the twin, and $\Delta \gamma^{n,t_i}$ is the shear increment of the twin system at the loading step. For pure magnesium and Mg-alloys, $S = 0.129$. The total fraction of twins is calculated locally at each quadrature point. If it exceeds a critical value (i.e., $\sum_i g^{n,t_i} > F_T$), the quadrature point is reoriented. Here, F_T is taken to be 0.25 based on the predominant twin reorientation (PTR) scheme in [194]. The crystal elastic parameters for Mg alloy are taken to be: $C_{11} = 59.3$, $C_{12} = 25.7$, $C_{13} = 21.4$, $C_{33} = 61.5$, and $C_{55} = 16.4$ (in GPa) [195]. The calibrated hardening parameters employed for the basal, prismatic,

Table 6.1: Hardening parameters for simulating stress–strain curves.

Mode	s_o^α (MPa)	h_o (MPa)	s_{sat} (MPa)
Basal $_{\langle a \rangle}$	76.0	225.6	248.7
Prism $_{\langle a \rangle}$	163.2	124.9	356.3
Pyram $_{\langle a \rangle}$	160.3	120.2	347.8
Pyram $_{\langle c+a \rangle}$	187.4	237.9	350.4
Twin $_{\langle c+a \rangle}$	116.4	105.6	238.3

pyramidal, and twin systems are tabulated in Table 6.1.

The stress–strain curves of the variants 1 and 2 of 3D microstructures are generated for both pure tension and compression boundary conditions with a mesh of $100 \times 100 \times 100$ elements using PRISMS–Plasticity code [190]. The results of the stress–

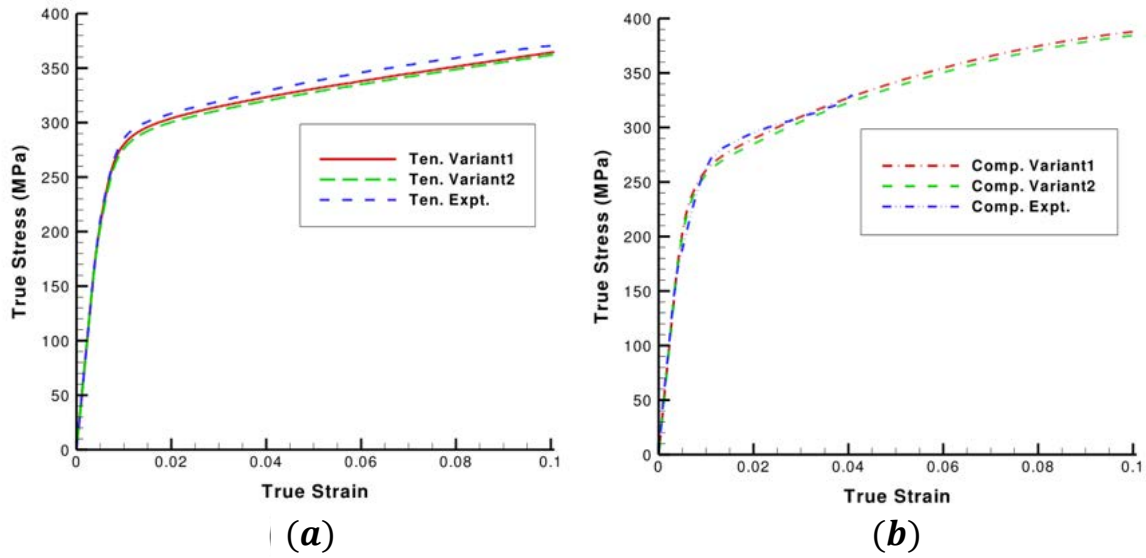


Figure 6.12: Experimental and simulated stress-strain curves for (a) tension (b) compression

The objective of the next exercise is to compare the CPFE surface strain results against DIC data for the reconstructed microstructures (e.g., variants 1 and 2) as well as a simple columnar representation, as seen in Figure 6.10(a), of the surface

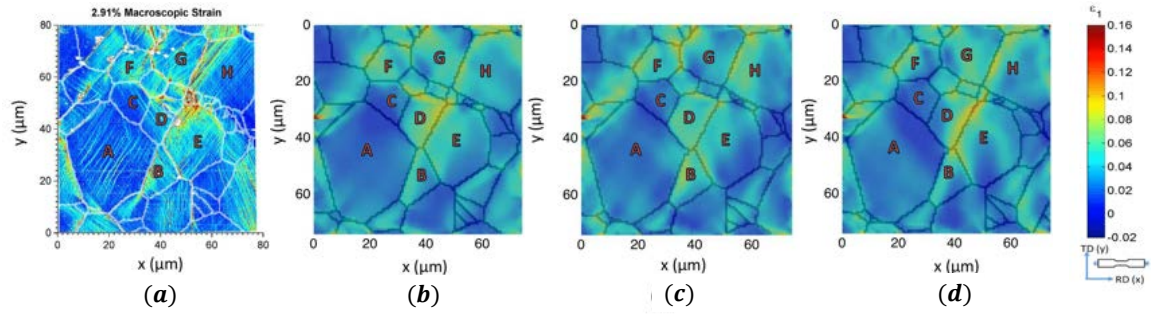


Figure 6.13: Comparison of principal strain ϵ_1 from (a) SEM–DIC experiment, (b) convexified columnar microstructure, (c) 3D variant 1, and (d) 3D variant 2 are illustrated.

grains. The approach is used to ascertain the sensitivity of the surface measurements to the underlying microstructure. Since the 3D boundary conditions are unknown from surface measurements, the measured surface displacements are primarily extended on the outer surface along the z -axis. As shown in the inset underneath the legend in Figure 6.13, the test corresponds to tension along with the x -axis of the microstructure.

The map of the principal strain from experiments, denoted as ϵ_1 , is given in Figure 6.13(a). Individual grains A–H are marked in order to investigate the variation in strains between columnar and the variants of the 3D microstructures. The corresponding map from our simulations on columnar and 3D variants 1 and 2 are indicated in Figures 6.13(b)–(d), respectively. It can be seen that the surface strain maps are similar across all observed morphologies and compare well with the experiments.

Note that the experimental data also contains slip traces that indicate the activity of different slip systems within the grains. These slip bands cannot be directly captured in CPFE and need to be inferred from the computed shearing rates, $\dot{\gamma}^\alpha$, for different slip and twinning systems. To better compare the surface observations, the relative activity of the individual slip systems (basal, prismatic, pyramidal $_{\langle a \rangle}$, pyramidal $_{\langle c+a \rangle}$, and twinning) are calculated and shown in Figure 6.14. The measured slip traces from experiments are shown on the left. The grains in the experimental results are colored

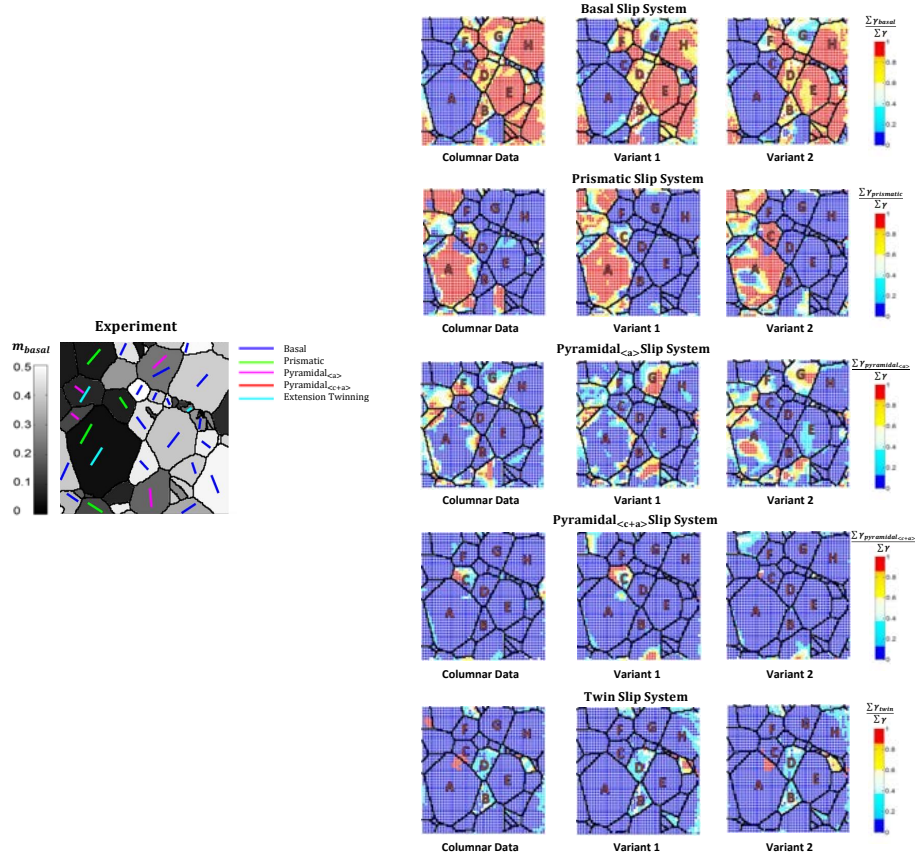


Figure 6.14: Comparison of relative activity of the various slip systems using quadrature point data from CPFE model for columnar microstructure, 3D variants 1 and 2. The m_{basal} denotes the Schmid factor for the basal slip system.

according to the Schmid factor for the basal slip system. In grains with high basal Schmid factor, the basal slip is typically active, and this trend is seen from all three cases considered. For grains with low basal Schmid factor, the activated non-basal slip modes show variations across the three cases.

From Figure 6.14, it can be seen that grain A depicts predominantly prismatic activity in columnar microstructure and variant 1, whereas it shows both prismatic and pyramidal $_{\langle a \rangle}$ activity in variant 2 of the microstructure. Additionally, grain B exhibits twinning activity in all three variants. Grain C observes pyramidal $_{\langle c+a \rangle}$ activity for both columnar microstructure and variant 1, whereas it shows predominantly prismatic slip activity in variant 2 as is also seen in the experiment. Grain D also exhibits both basal and twin activities within all three microstructures. Grain E predominately shows basal slip activity in all three microstructures, as also in the experiment. However, pyramidal $_{\langle a \rangle}$ activity of variant 2 can also be distinctly observed in grain E. Grains F and G exhibit basal and pyramidal $_{\langle a \rangle}$ slip activity for all three microstructures, in line with the experiment. Grain H shows basal slip activity for the columnar variant, whereas it shows both basal and twin activity for 3D variants 1 and 2. It can be seen that the slip activity in grains A, C, E, and H of microstructure variant 2 varies from the columnar microstructure. Grains A, C, and E also show differences in slip activity between 3D variant 1 and 3D variant 2. These grains were chosen to study the effect of the underlying microstructure. The slip activity at different depths of the material for the 3D variant 2 is particularly studied.

The relative slip activity of the variant 2 microstructure at various depths are plotted in Figure 6.15. Variations in non-basal activity occur due to 3D grain structure when strong basal slip activation in underlying grains triggers other non-basal systems in surface grains with low basal Schmid factor. It can be observed in case of grain A where both prismatic and pyramidal $_{\langle a \rangle}$ activities are observed, the re-

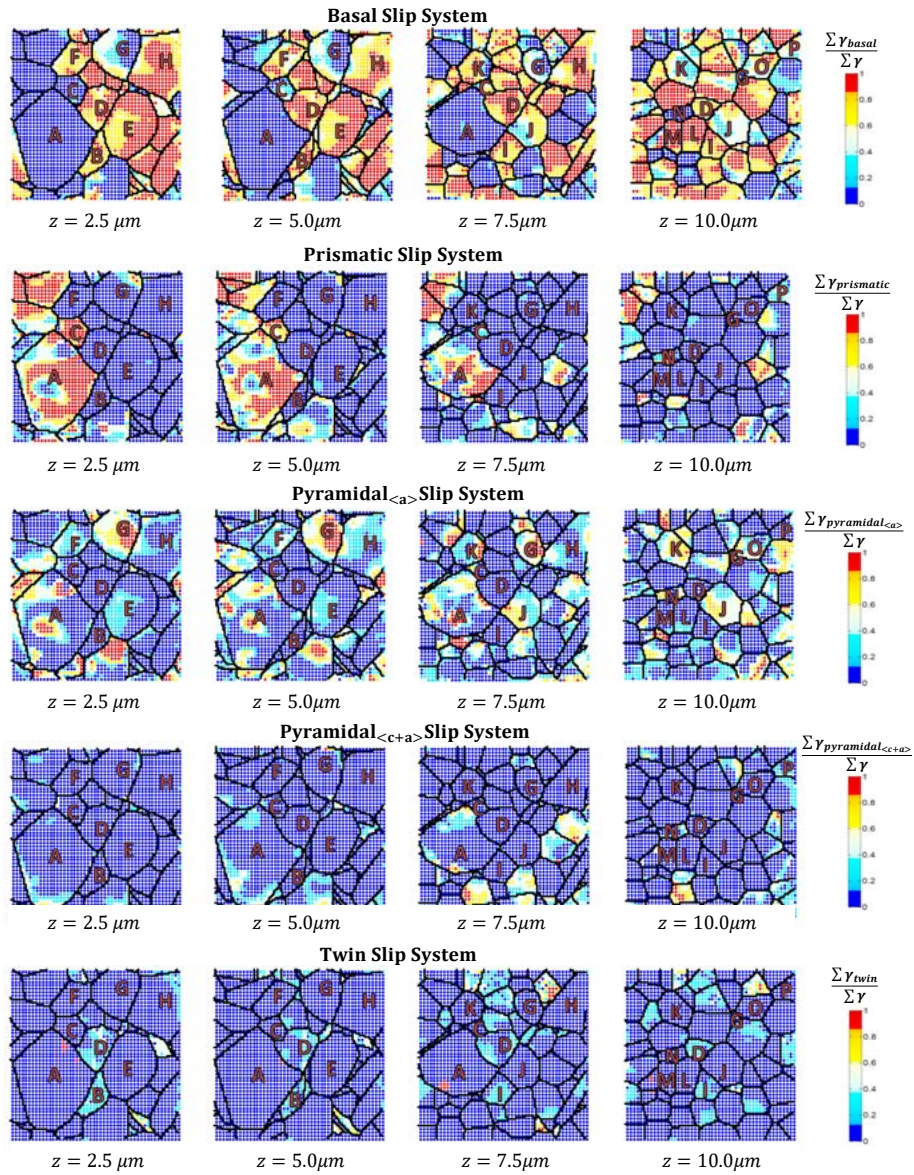


Figure 6.15: Comparison of the relative activity of the various slip systems using quadrature point data from CPFE model for 3D variant 2 along the depth of the sample.

gion of pyramidal $_{\langle a \rangle}$ activity is due to the grains underneath the surface, such as M and L. These two grains, as shown in Figure 6.15, exhibit strong basal slip activity. Other factors such, as slip systems that allow favorable transmission of loads across misoriented subsurface grains, are also at play. The region of grain E, which observes pyramidal $_{\langle a \rangle}$ activity arises possibly from the grain J, which also exhibits pyramidal $_{\langle a \rangle}$ activity. It is also observed that grains D and G extend far deeper into the surface of 3D microstructure, and hence show little differences in observed slip modes as compared to columnar microstructures. As a summary, these results indicate the influence of subsurface grains in determining slip modes on surface grains and the need for statistical analysis based on 3D realizations for comparing slip activity observed from DIC and CPFEM calculations.

In the case of non-convex microstructures, the reconstruction algorithm in its current format would convexify every non-convex grain in the microstructure through the process described in Section 6.2.2 and eliminate all the internal granular facets. Nevertheless, realistic microstructures are generally non-convex in topology, especially for the case of 3D microstructures developed by non-equilibrium processes such as 3D printing. Here, several experimental EBSD images are required to capture the average grain size in the z -direction. This can help in determining the length between stacked slices for 3D reconstruction, as discussed in Section 6.2.4. Alternatively, the additional imaging could be used to build the microstructure by stacking EBSD slices in x or y direction as opposed to the z -direction. Hence, a future extension of the current algorithm based on multiple generator points can be used to model non-convex microstructures. This can be done by first partitioning the non-convex granular domain into a sum of convex subgrains using an already available algorithm such as triangulation, Hertel-Mehlhorn algorithm, and Chazelle's complex cubic algorithm. Using such methods, the twinned region can be better represented as separate internal facets within the grains, and the remaining grain can be split into multiple convex

domains. Such an approach is out of the scope of the current chapter and will be directed in future.

6.4 Conclusions

In this chapter, an approach for the construction of 3D microstructures from 2D surface images through Voronoi tessellation is developed. The key novelty from classical reconstruction methods is that the surface microstructure is retained. The issue of realism of the transition region between the surface and subsurface microstructure is addressed using an efficient generalized inverse Voronoi approach with a convexification step. Crystal plasticity numerical models are used for comparing the predicted surface strains in the reconstructed 3D microstructures against experimental measurements. It is observed that the surface strains of different 3D reconstructed microstructures are qualitatively similar to the experiment. However, strong basal slip activation in some subsurface grains may influence the choice of activated slip systems on surface microstructure. Thus, the results presented in this work show the implications of performing a full 3D analysis of measured surface data as compared to only analyzing a 2D extruded model.

CHAPTER VII

Conclusions

7.1 Summary

This report's primary goal is to develop a comprehensive image-based framework was introduced for 3D microstructure synthesis over very large length-scales. The complete workflow from the 2D experimental EBSD images to a CAD model visualization is an important and novel contribution in this work. Here, a set of three Electron Backscatter Diffraction (EBSD) sections imaged on orthogonal planes was first obtained via microscopy imaging. The microstructures are subsequently converted to a 3D representative unit cell using the Markov Random Field (MRF) modeling technique. The MRF algorithm reconstructs 3D images through matching 3D slices at different voxels to the experimental sectional micrographs while ensuring neighborhoods of voxels taken from the 2D micrographs have meshed together seamlessly in the 3D image. Thereafter, statistical analyses of textural features, such as Orientation Distribution Function (ODF) and grain boundary misorientation angles for the reconstructed 3D microstructures were carried out against the original 2D EBSD images. The results demonstrate that MRF can effectively provide 3D prediction of microstructure. The LEGOMAT procedure embeds 3D microstructures in an engineering-scale Computer-Aided Design (CAD) geometry accordant with a material flow path, accounting for variations in grain size and grain shapes (equiaxed

vs. elongated). Here, the grain sizes and shapes are defined in terms of a tensor field based on known microstructure patterns seen from the experimental characterization technique. Because the large-scale microstructure models are primarily based on geometry, the reconstructions are extremely robust and require minimal computational power when compared to existing methods.

This report’s secondary goal is to identify, formalize, and solve important microstructure-related problems using a graph-based approach. We made the following contributions:

- We developed a methodology to label the elements of a finite element such that it can replicate experimental images of materials with smooth boundaries like polycrystalline and composite materials [153].
- We took a similar approach as above to model cracked surface in a finite element mesh [196]. In this case, we developed a modular framework for multiscale modeling of crack paths in polycrystalline materials. The modularity allows Plug’n’Play capability for the testing of different energy-based criteria for fracture. The problem pursued here employed Maximum energy release rate criteria to estimate brittle failure in elastic polycrystalline material with anisotropic surface energies.
- We also developed an approach for studying multiphase flow in uniform meshes. Surface energies drive this phenomenon; however, accurately estimating them in these meshes is challenging. We extended the known methodology for pixel mesh developed by [161] to Quadtree meshes. This modification can significantly improve the computational cost for high fidelity simulation. These methodologies are extended to anisotropic grains and 3D microstructure. Application to denoising microstructure is also discussed.

7.2 Applications of proposed methodologies

LEGOMAT: Although the study presented in this work shows significant promise of the LEGOMAT approach, future work will focus on increasing the resolution of the reconstructions and development of methods to reduce seams that occur along the intersecting surfaces of tetrahedral groups in the CAD model. In the presented cases, these seams are not pronounced due to the highly stochastic nature of the additive microstructure. The approach is specifically useful as an efficient visualization tool to understand microstructure variability as a function of process parameters and identify possible weak links (fracture paths) at larger scales than what can be measured experimentally. The current work can benefit from machine learning methods to map location-specific crystallographic texture to the reconstructed 3D grain structure and to identify outliers in the reconstruction where additional experimental data can be used to improve upon the visualization.

Fracture: The approach proposed for fracture has significant advantages over continuum methods with diffuse damage (smeared crack, gradient damage, or phase field methods). The fracture surface geometry is clearly identified (sharp interface), allowing one to calculate the surface energy along the path accurately. Thus, the surface energy of the crystal and grain boundaries modeled using smaller-scale simulations such as molecular dynamics [197] or density functional theory can be directly used in this approach. In a previous work (MicroFract code [105]), we had shown the applicability of the model to composite (matrix–fiber) crack paths (Fig 7.1). Here, matrix–inclusion or matrix–fiber interface properties can come from molecular dynamics [198, 199, 200]. It also showed the applicability of the method to modeling intergranular stress–corrosion cracks, where the effect of alloying or impurities on the grain boundary energy can come from density functional theory [201].

Further work on validating microstructurally short crack paths in 2D thin foil specimens against experimental results [202, 203] from Allison group at the University

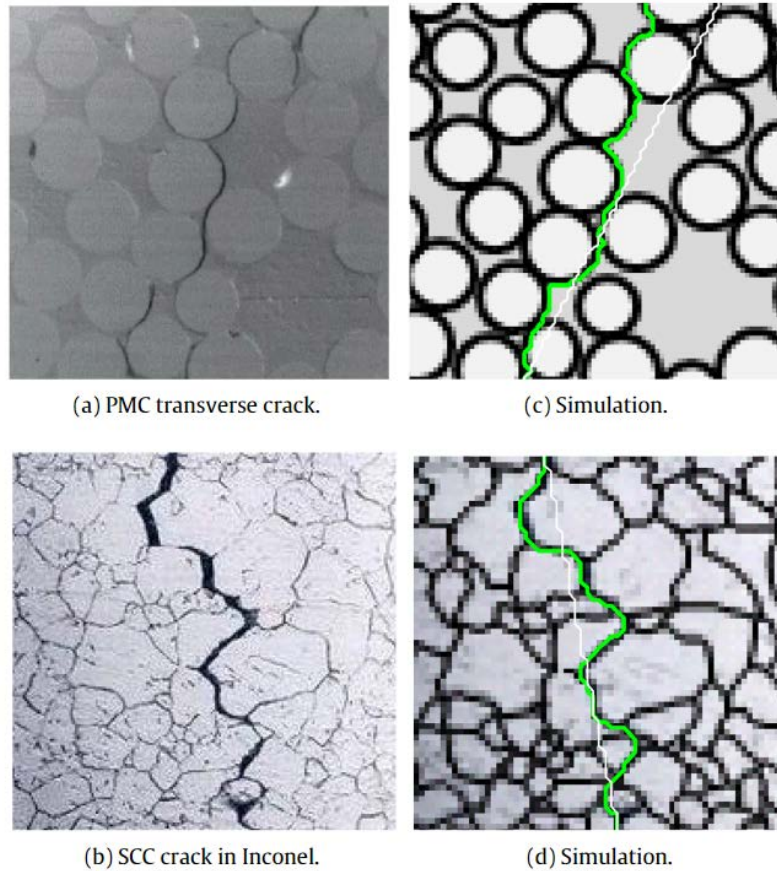


Figure 7.1: (a,b) Transverse cracks under tension normal to fibers in a polymer matrix composite and intergranular stress corrosion cracking of Inconel (c,d) Graph cuts solution: green line indicates the computed crack trajectory.

of Michigan is underway and has shown very promising results. Code extension to 3D fracture modeling has been already enabled (Fig 7.2), however formal theoretical models of anisotropic 3D crack tip stress intensity factor need to be developed for the model before comparison against experimental 3D fracture paths obtained from high energy synchrotron measurements (Allison group, [204]). It will be of interest to extend this method for modeling failure, including plasticity. In this case, one could split the surface energy into two terms, cleavage energy based on crack normal as described in this thesis and a plastic work that additionally depends on the shear direction obtainable from crystal plasticity models [205, 100, 190]. Beyond a post-fracture method, it is possible to use the code to identify possible fracture paths

during in-situ testing when using plastic strain localization images from methods such as digital image correlation [187, 186] or elastic stress localization images from high resolution electron backscatter diffraction as inputs [206, 207].

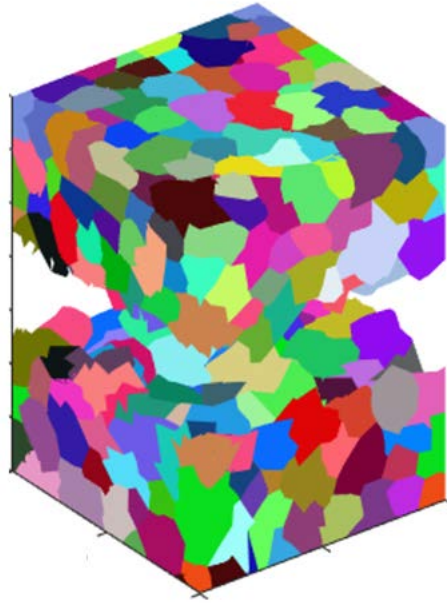


Figure 7.2: 3D fracture modelled using the graph coloring method proposed in this thesis.

Segmentation and denoising polycrystals Recently, new and powerful techniques based on feature-matching algorithms [67] or MRF techniques have been developed for synthetic reconstruction of multiphase alloys microstructures [35, 85, 89, 36, 208]. The techniques work by matching observed features in an example 2D image to a synthesized 3D microstructure. However, these techniques do not enforce any particular physics-based constraints on grain boundary structure. In materials science of GBs, certain facets are preferred, and GB junctions often follow certain geometric (angular) constraints. Graph segmentation methods provide a pathway towards including such physics-based constraints to improve the overall reconstruction. As proposed in the chapter on grain evolution, an energy-based approach can be used as a post-processing step, allowing for minimization of GB energy in segmented MRF images (Fig 7.3). The benefits are three-fold: the grains are segmented from the color space

outputs of these models; it gives the users an option to force desired grain shapes (e.g., equiaxed or columnar grains) during segmentation, and finally, the GB angles can be corrected based on global energetics. The Riemannian metric dictates optimal grain shapes. GB energies of all pairs of grains in the microstructure (as a function of crystallography only) can be included in the form of Ising interaction term. As a result, minimization of the cost function can lead to an energy-efficient description of GB interfaces in synthetic microstructure models.

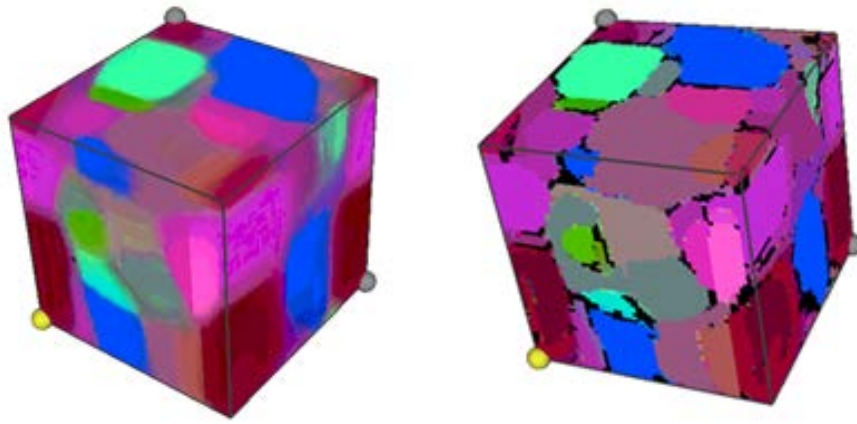


Figure 7.3: (left) Synthetic microstructure from MRF algorithm. (right) Use of graph cut segmentation to identify the grain interfaces (shown in black)

APPENDICES

APPENDIX A

Theorems and Propositions

Proposition A.1. *Given a straight crack at an angle α , defined as a mesh partition with label $+1$ in the upper sector and -1 in the lower sector, Eq (4.6) holds true in the continuum limit of the mesh when g has the following form where $k < 2$:*

$$g(c_i, l_i) \equiv (2 - k)\bar{r}^{k-1}\frac{\bar{g}(\theta_i, l_i)}{r_i^k}$$

Proof. Taking the continuum limit:

$$\begin{aligned} \lim_{h \rightarrow 0} \sum_{P_i \in \mathcal{P}_h} (2 - k)\bar{r}^{k-1}\frac{\bar{g}(\theta_i, l_i)}{r_i^k}|P_i| &= \int_A (2 - k)\bar{r}^{k-1}\frac{\bar{g}(\theta, l)}{r^k}dA \\ &= \int_{-\pi/2}^{\pi/2} \int_0^{\bar{r}} (2 - k)\bar{r}^{k-1}\frac{\bar{g}(\theta, l)}{r^k}rdrd\theta \\ &= \bar{r} \int_{-\pi/2}^{\pi/2} \bar{g}(\theta, l)d\theta \end{aligned}$$

Choosing the labels as $l = 1$ when $\theta \in [\alpha, \pi/2]$ and $l = -1$ when $\theta \in [-\pi/2, \alpha)$ gets the desired result. $\square$

Proposition A.2. *Eq (4.7) satisfies Eq (4.6)*

Proof. Taking derivative of Eq (4.6) with respect to α on both sides:

$$\bar{g}(\alpha, +1) - \bar{g}(\alpha, -1) = G'(\alpha)$$

Clearly the form provided in Eq (4.7) satisfies the above equation. The variable c can be estimated by substituting this form back in Eq (4.6) and evaluating it for $\alpha = \pi/2$ to get:

$$c = \frac{1}{\pi} \left(-G(\pi/2) + \int_{-\pi/2}^{\pi/2} G'(\theta) H(-G'(\theta)) d\theta \right)$$

Similarly, evaluating it for $\alpha = -\pi/2$ gives:

$$c = \frac{1}{\pi} \left(-G(-\pi/2) - \int_{-\pi/2}^{\pi/2} G'(\theta) H(G'(\theta)) d\theta \right)$$

Note that both values are equal as:

$$\begin{aligned} G(\pi/2) - G(-\pi/2) &= \int_{-\pi/2}^{\pi/2} G'(\theta) d\theta = \int_{-\pi/2}^{\pi/2} G'(\theta) (H(G'(\theta)) + H(-G'(\theta))) d\theta \\ \implies -G(-\pi/2) - \int_{-\pi/2}^{\pi/2} G'(\theta) H(G'(\theta)) d\theta &= -G(\pi/2) + \int_{-\pi/2}^{\pi/2} G'(\theta) H(-G'(\theta)) d\theta \end{aligned}$$

□

Proposition A.3. *Given a rough crack represented using its angular position (α) as a function of the radial position, Eq (4.9) holds in the continuum limit.*

Proof. As earlier, the crack is defined as a mesh partition with label +1 in the upper

sector and -1 in the lower sector. Taking the continuum limit:

$$\begin{aligned}
\lim_{h \rightarrow 0} \sum_{P_i \in \mathcal{P}_h} g(c_i, l_i) |P_i| &= \int_A (2-k) \bar{r}^{k-1} \frac{\bar{g}(\theta, l)}{r^k} dA \\
&= \int_0^{\bar{r}} \int_{-\pi/2}^{\pi/2} (2-k) \bar{r}^{k-1} \frac{\bar{g}(\theta, l)}{r^k} r d\theta dr \\
&= (2-k) \bar{r}^{k-1} \int_0^{\bar{r}} \frac{1}{r^{k-1}} \left(\int_{\alpha(r)}^{\pi/2} \bar{g}(\theta, +1) d\theta + \int_{-\pi/2}^{\alpha(r)} \bar{g}(\theta, -1) d\theta \right) dr
\end{aligned}$$

Using Eq (4.6),

$$\lim_{h \rightarrow 0} \sum_{P_i \in \mathcal{P}_h} g(c_i, l_i) |P_i| = (k-2) \bar{r}^k \int_0^{\bar{r}} \frac{1}{r^{k-1}} G(\alpha(r)) dr$$

Normalizing the radial variable from r to $s = r/\bar{r}$,

$$\lim_{h \rightarrow 0} \sum_{P_i \in \mathcal{P}_h} g(c_i, l_i) |P_i| = (k-2) \bar{r} \int_0^1 \frac{1}{s^{k-1}} G(\alpha(\bar{r}s)) ds$$

□

Proposition A.4. *For a planar curve defined as a level set function $C = \psi^{-1}(a)$ with $a \in \mathbb{R}$. Following geometric relations are true at any $(x, y) \in C$:*

1. *The normal, $\vec{n}$:*

$$\vec{n} = \frac{\nabla \psi}{|\nabla \psi|} \Big|_{(x,y)}$$

2. *The curvature, κ ,*

$$\kappa = -\nabla \cdot \frac{\nabla \psi}{|\nabla \psi|} \Big|_{(x,y)}$$

Proof. Let s denote the arc-length parameterization of the level set. As level set is constant over C , one has, $\frac{\partial^n \psi}{\partial s^n} = 0$ for all n .

1. Taking the case, $n = 1$:

$$\psi_x x_s + \psi_y y_s = 0$$

$$C_s \cdot \nabla \psi = 0$$

By definition of $\partial C / \partial s$ is tangential to the curve. Therefore, $\nabla \psi$ is orthogonal to the curve. The sign is arbitrarily chosen to be positive in the direction of $\nabla \psi$.

2. By definition of curvature of curves, $\kappa \vec{n} = \frac{\partial^2 C}{\partial s^2}$. The curvature is estimated by considering, $\frac{\partial^2 \psi}{\partial s^2} = 0$.

$$\frac{\partial}{\partial s} (\psi_x x_s + \psi_y y_s) = 0$$

$$\psi_{xx} x_s^2 + 2\psi_{xy} x_s y_s + \psi_{yy} y_s^2 + \psi_x x_{ss} + \psi_y y_{ss} = 0$$

$$\psi_{xx} x_s^2 + 2\psi_{xy} x_s y_s + \psi_{yy} y_s^2 + \nabla \psi \cdot C_{ss} = 0$$

$$\psi_{xx} x_s^2 + 2\psi_{xy} x_s y_s + \psi_{yy} y_s^2 + \kappa \nabla \psi = 0$$

Using the normals,

$$x_s = \frac{\psi_y}{|\nabla \psi|}, \quad y_s = -\frac{\psi_x}{|\nabla \psi|}$$

Substituting it back,

$$\frac{\psi_{xx} \psi_y^2 - 2\psi_{xy} \psi_x \psi_y + \psi_{yy} \psi_x^2}{|\nabla \psi|^2} + \kappa \nabla \psi = 0$$

$$\nabla \cdot \frac{\nabla \psi}{|\nabla \psi|} + \kappa = 0$$

□

Theorem V.3 Consider, $D \subseteq \mathbb{R}^2$ where D is a fixed region. And a open set Ω with smooth boundary, $\Gamma = \partial\Omega$. Then for any Ω , with $\bar{\Omega} \neq \phi$ or D , to be the extremizer of functional:

$$F(\Omega) = f_1 \text{Area}(\Omega) + f_0 \text{Area}(D - \Omega) + \gamma \text{length}(\Gamma)$$

following condition is satisfied on the boundary, Γ , in the interior of domain D :

$$\gamma\kappa + f_1 - f_0 = 0 \tag{A.1}$$

Moreover, in the absence of any fixed/Dirichlet boundary conditions, Γ makes perpendicular intersections with ∂D .

Proof. The level set approach of [158] is adapted here. Consider $\psi(x, y)$, a Lipschitz continuous Level-set function satisfying:

$$\begin{aligned} \psi(x, y) &> 0 && \text{for } (x, y) \in \Omega \\ \psi(x, y) &= 0 && \text{for } (x, y) \in \partial\Omega \\ \psi(x, y) &< 0 && \text{for } (x, y) \in D - \bar{\Omega} \end{aligned} \tag{A.2}$$

The boundary is estimated as the zero level set, $\Gamma = \psi^{-1}(0)$. The normal, $\vec{n}$ and curvature, κ are estimated in Proposition A.4. The geometric properties of Ω is estimated as follows:

$$\begin{aligned} \text{length}(\partial\Omega) &= \int \int \delta(\psi(x, y)) |\nabla \psi(x, y)| dx dy \\ \text{Area}(\partial\Omega) &= \int \int H(\psi(x, y)) dx dy \end{aligned}$$

where $H(x)$ is the Heaviside function and $\delta(x)$ is the Dirac delta function. An important property that will be used later in the proof is that the distributional derivative of $H(x)$ is $\delta(x)$. Writing $F(\Omega)$ using the level set:

$$F(\Omega) = \int \int \gamma \delta(\psi(x, y)) |\nabla \psi(x, y)| + (f_1 H(\psi) + f_0 H(-\psi)) dx dy$$

Taking the Gateaux derivative with respect to ψ in the direction of $\tilde{\psi}$:

$$\left(\frac{\partial F}{\partial \psi}, \tilde{\psi} \right) = \int_D \gamma \left(\delta'(\psi) |\nabla \psi| \tilde{\psi} + \frac{\delta(\psi) \nabla \psi \cdot \nabla \tilde{\psi}}{|\nabla \psi|} \right) + (f_1 \delta(\psi) - f_0 \delta(-\psi)) \tilde{\psi} dx dy$$

Applying Divergence theorem with $\vec{n}_{\partial D}$ as the normal to ∂D :

$$\begin{aligned} &= \int_D \left(\gamma \left(\delta'(\psi) |\nabla \psi| - \nabla \cdot \frac{\delta(\psi) \nabla \psi}{|\nabla \psi|} \right) + (f_1 - f_0) \delta(\psi) \right) \tilde{\psi} dx dy \\ &\quad + \int_{\partial D} \delta(\psi) \gamma \frac{\nabla \psi}{|\nabla \psi|} \cdot \vec{n}_{\partial D} ds \\ &= \int_D \delta(\psi) \left(-\gamma \nabla \cdot \frac{\nabla \psi}{|\nabla \psi|} + (f_1 - f_0) \right) \tilde{\psi} dx dy + \int_{\partial D} \delta(\psi) \gamma \frac{\nabla \psi}{|\nabla \psi|} \cdot \vec{n}_{\partial D} ds \end{aligned}$$

where the following fact (in the sense of distributions) is used:

$$\nabla (\delta(\psi) \nabla \psi / |\nabla \psi|) = \delta'(\psi) |\nabla \psi| + \delta(\psi) \nabla \cdot (\nabla \psi / |\nabla \psi|)$$

Setting the Gateaux derivative to zero gives following results:

$$\begin{aligned} \gamma \kappa + f_1 - f_0 &= 0 & \text{on} & \quad \Gamma \cap \text{Int}(D) \\ \gamma \vec{n} \cdot \vec{n}_{\partial D} &= 0 & \text{on} & \quad \Gamma \cap (\partial D)_{\text{Neumann}} \end{aligned}$$

It should be noted that on the boundary, if $\Gamma \cap (\partial D)_{\text{Neumann}}$ are disconnected points then $\vec{n}$ and $\vec{n}_{\partial D}$ orthogonal to each other. However, they are not allowed to be

parallel to each other. This creates an unimaginable situation where Γ intersects ∂D but does not merge with it. This contradiction arise from the way the surface energy is introduced through levels sets. Notice that there is no surface energy on ∂D when it does not intersect Γ while that is not the case when Ω merges with the boundary. It stands to reason, that the surface energy of Γ should not be included when it lies on the boundary. There is no attempt here to resolve this problem with complete mathematical rigour. However, following intuitive picture can be used to convince oneself that this contradiction does not interfere with the physical process described here. Consider an open set $D_0 \supset D$ and extend ψ to D_0 , now the merger of grain to ∂D can be understood as a portion of Γ being present in $D_0 - D$.

□

Theorem V.9 Consider the following minimizer

$$\Gamma_{t+\Delta t} = \arg \min_{\Gamma} \left(\alpha F(\Gamma) + \frac{\text{dist}(\Gamma, \Gamma_t)}{2\Delta t} \right)$$

where Γ and Γ_t are closed curves and $F(\Gamma) \equiv F(\Omega)$ with Ω denoting the interior region of the curve. It has following property

$$\Gamma_{t+\Delta t} = \Gamma_t + (v\Delta t)\vec{n}, \quad v = \alpha(\gamma\kappa + (b_1 - b_0))$$

Proof. The same level set approach as in the proof of Theorem V.3 is followed here. The level set for Γ_t is denoted as ψ_t and the level set for Γ is denoted as ψ . All the properties mentioned in Eq (A.2) are applied to the respective level sets. In addition, the level set functions ψ and ψ_t are endowed with a special property that they represent the negative signed distance to the curves $\psi^{-1}(0)$ and $\psi_t^{-1}(0)$, respectively. In concise notations: $p_{\Gamma_t}(x, y) = |\psi_t(x, y)|$ and $p_{\Gamma}(x, y) = |\psi(x, y)|$ where p is the perpendicular distance from respective curves as defined in Eq (5.5). It is clear that this extra condition does not interfere with the properties defined in Eq (A.2). Moreover,

the proof for theorem V.3 remains unaffected as it only depends on the local behavior of cure at the boundary.

The regularizer in the energy can be approximately represented as:

$$\begin{aligned}\frac{\text{dist}(\Gamma, \Gamma_t)}{2\Delta t} &\approx \frac{1}{2\Delta t} \int_{\Gamma_t} \int_0^{|\Gamma(s) - \Gamma_t(s)|} pdpds \\ &= \int \int |\psi_t| (H(\psi)H(-\psi_t) + H(-\psi)H(\psi_t)) dxdy\end{aligned}$$

The term $H(\psi)H(-\psi_t) + H(-\psi)H(\psi_t)$ has a value of 1 in the region between the zero level sets of ψ_t and ψ . Noting that ψ_t is constant and taking the Gateaux derivative with respect to ψ in the direction of $\tilde{\psi}$:

$$\begin{aligned}\left(\frac{\text{dist}(\Gamma, \Gamma_t)}{\Delta t}, \tilde{\psi}\right) &= -\frac{1}{\Delta t} \int \int \delta(\psi) |\psi_t| (H(\psi_t) - H(-\psi_t)) \tilde{\psi} dxdy \\ &= -\frac{1}{\Delta t} \int \int \delta(\psi) \psi_t \tilde{\psi} dxdy\end{aligned}$$

The Gateaux derivative of the functional described in Eq 5.6 is now estimated as:

$$\begin{aligned}\left(\alpha F(\Gamma) + \frac{\text{dist}(\Gamma, \Gamma_t)}{2\Delta t}, \tilde{\psi}\right) &= \int_D \delta(\psi) \left(-\gamma \nabla \cdot \frac{\nabla \psi}{|\nabla \psi|} + (f_1 - f_0) - \frac{\psi_t}{\Delta t} \right) \tilde{\psi} dxdy \\ &\quad + \int_{\partial D} \delta(\psi) \gamma \frac{\nabla \psi}{|\nabla \psi|} \cdot \vec{n}_{\partial D} ds\end{aligned}$$

Noting that for distance level sets, $\psi_t \delta(\psi) = (\Gamma - \Gamma_t) \cdot \vec{n}$ (in terms of distributions), setting the Gateaux derivative to zero yields following equation in the interior:

$$\alpha(\gamma \kappa + f_1 - f_0) - \frac{\psi_t}{\Delta t} = 0$$

The above equation can then be rearranged in terms of the extremizer, $\Gamma_{t+\Delta t} \equiv \Gamma$,

$$\Gamma_{t+\Delta t} = \Gamma_t + \alpha(\gamma\kappa + f_1 - f_0)\Delta t \vec{n}$$

□

Theorem V.16 In the setup of Theorem V.3, consider the following functional:

$$F(\Omega) = f_1 \text{Area}(\Omega) + f_0 \text{Area}(D - \Omega) + \int_{\Gamma} \gamma(\vec{n}(s)) ds$$

where $\gamma(\vec{n})$ is the surface energy in the direction of unit normal, $\vec{n}$. The extremizer satisfies the following condition on the boundary, Γ :

$$\nabla \cdot (\gamma \vec{n} + (\nabla \gamma \cdot \vec{t}) \vec{t}) + f_1 - f_0 = 0 \quad (\text{A.3})$$

where $\vec{n}$ and $\vec{t}$ are normal and tangents of Γ . Moreover, in the absence of any fixed/Dirichlet boundary conditions, Γ intersects ∂D at an non-zero angle such that following condition holds true:

$$\gamma \vec{n} \cdot \vec{n}_{\partial D} + (\nabla \gamma \cdot \vec{t}) \vec{t} \cdot \vec{n}_{\partial D} = 0$$

Proof. For most part, the proof follows from the proof of Theorem V.3, however, Gateaux derivative of the surface term needs to be calculated:

$$\begin{aligned} \left(\int_D \gamma \left(\frac{\nabla \psi}{|\nabla \psi|} \right) \delta(\psi) |\nabla \psi|, \tilde{\psi} \right) &= \int_D \gamma \left(\frac{\nabla \psi}{|\nabla \psi|} \right) \left(\delta'(\psi) |\nabla \psi| \tilde{\psi} + \frac{\delta(\psi) \nabla \psi \cdot \nabla \tilde{\psi}}{|\nabla \psi|} \right) + \\ &\quad \delta(\psi) \left(\nabla \gamma \cdot \nabla \tilde{\psi} - \frac{(\nabla \gamma \cdot \nabla \psi)(\nabla \psi \cdot \nabla \tilde{\psi})}{|\nabla \psi|^2} \right) dx dy \end{aligned}$$

Using Divergence theorem::

$$\begin{aligned}
&= \int_D \gamma \delta'(\psi) |\nabla \psi| \tilde{\psi} - \nabla \cdot \left(\gamma \delta(\psi) \frac{\nabla \psi}{|\nabla \psi|} + \delta(\psi) \nabla \gamma - \delta(\psi) \frac{(\nabla \gamma \cdot \nabla \psi) \nabla \psi}{|\nabla \psi|^2} \right) \tilde{\psi} dx dy \\
&\quad + \int_{\partial D} \delta(\psi) \left(\frac{\gamma}{|\nabla \psi|} \nabla \psi + \nabla \gamma - \frac{\nabla \gamma \cdot \nabla \psi}{|\nabla \psi|^2} \nabla \psi \right) \cdot \vec{n}_{\partial D} \tilde{\psi} ds
\end{aligned}$$

Simplifying:

$$\begin{aligned}
&= \int_D -\delta(\psi) \nabla \cdot \left(\gamma \frac{\nabla \psi}{|\nabla \psi|} + \nabla \gamma - \frac{(\nabla \gamma \cdot \nabla \psi) \nabla \psi}{|\nabla \psi|^2} \right) \tilde{\psi} dx dy \\
&\quad + \int_{\partial D} \delta(\psi) \left(\frac{\gamma}{|\nabla \psi|} \nabla \psi + \nabla \gamma - \frac{\nabla \gamma \cdot \nabla \psi}{|\nabla \psi|^2} \nabla \psi \right) \cdot \vec{n}_{\partial D} \tilde{\psi} ds
\end{aligned}$$

Observe that $\nabla \gamma - (\nabla \gamma \cdot \vec{n}) \vec{n}$ is the component of $\nabla \gamma$ that is tangential to the Γ .

The rest follows as the proof of Theorem V.3.

□

Proposition V.12: Consider the set of all lines with rational slope p/q (p and q are relative prime) that pass through atleast one grid point (i.e a point $(a, b) \in \mathbb{I}^2$).

These lines are regularly placed with following distance between two nearest lines:

$$\Delta \varrho = \frac{1}{\sqrt{p^2 + q^2}} \quad (\text{A.4})$$

Proof. It is easy to see that if the line passes through one grid point, then it passes through infinitely many grid points as if $(a, b) \in \mathbb{I}^2$ lie on the line then so does $(a + q, b + p) \in \mathbb{I}^2$. To see these lines are regularly placed, observe that these lines can be constructed by choosing a single line and then constructing the next parallel line as the one passing through a grid point nearest to the initial line. Without loss

of generality, consider that the initial line is passing through the origin,

$$L := qy - px = 0$$

The distance of any general point (x_0, y_0) from line L is given as:

$$d_L(x_0, y_0) = \frac{|qy_0 - px_0|}{\sqrt{p^2 + q^2}}$$

Clearly, if p, q, x_0, y_0 are integers, then the minimum possible non-zero value of $|qy_0 - px_0| = 1$. In fact, the minimum value is exactly 1. This is true because $qy_0 - px_0 = 1$ is a diophantine equation and $\gcd(p, q) = 1$. This is a sufficient condition for an integral solution of the above equation to exist. This proves that the minimum non zero distance between a vertex position and the line L is $\frac{1}{\sqrt{p^2 + q^2}}$. $\square$

APPENDIX B

Pseudo-Codes

B.1 Conformal mesh generation

The data structure and the pseudo-code for the mesh labeling algorithm are presented here. There are 2 kinds of data that are required in this algorithm: (1) FE Mesh data and (2) Image data. It is recommended to calculate and save all the variables as defined below at the time of mesh generation and be passed on to mesh labeling algorithm. The FE mesh data structure is pre-processed to include the following variables:

- *Dimension*: Value is 2 for 2D and 3 for 3D
- *MeshSize*: The x-dimension of mesh is in the range $[0, MeshSize(1)]$ and the y-dimension is in the range $[0, MeshSize(2)]$. For 3D mesh, the z-direction is in the range $[0, MeshSize(3)]$.
- *Coordinate*: Location of each node of the mesh
- *Nnode*: Number of nodes

- *Connectivity*: Contains tuples with nodes of each element
- *Nelement*: Number of elements
- *Neighbor*: In 3D meshes this data contains each pair of elements which share a face. In case of 2D, it contains each pair of elements which share an edge.
- *Npair*: Number of Neighbors
- *ElementVolume*: Defined only for 3D meshes and contains the volume of each element.
- *Face*: Defined only for 3D meshes and contains tuples with nodes of the shared faces (indexed as the *Neighbor* data)
- *FaceArea*: Area of face defined in *Face* data
- *FaceNormal*: Unit Normal of face defined in *Face* data
- *ElementArea*: Defined only for 2D meshes and contains the area of each element.
- *Edge*: Defined only for 2D meshes and contains tuples with nodes of shared edges (indexed as the *Neighbor* data)
- *EdgeLength*: Length of each edge defined in *Edge* data
- *EdgeNormal*: Unit Normal of each edge defined in *Edge* data

The Image data structure includes the following variables:

- *ImageSize*: The tuple (N_x, N_y, N_z) represents the Voxel size of the image in each direction. In case of 2D, this variable contains a pair (N_x, N_y)

- *XGridData*: Contains the x-coordinate value of the Voxelated/Pixelated image with size determined by *ImageSize*.
- *YGridData*: Contains the y-coordinate value of the Voxelated/Pixelated image with size determined by *ImageSize*.
- *ZGridData*: Defined only for 3D image data. Contains the z-coordinate value of the Voxelated/Pixelated image with size determined by *ImageSize*.
- *LabelGridData*: Label values at each grid point.
- *NLabel*: Number of Labels
- *LabelValues*: Contains the label values of enumerated from 1 to Nlabels

As discussed in the main text, this method uses an energy minimization approach. Pseudo-code for estimation of Data cost and Smooth cost is presented in Algorithm 2 and Algorithm 3, respectively. The energy minimization is carried out using the GCO library developed in [116]. The pseudocode for using this library is presented in Algorithm 4

Algorithm 2 Set-up Data cost (2D/3D)

- 1: $M \leftarrow$ Mesh data ▷ Import mesh data
- 2: $I \leftarrow$ Image data ▷ Import Experimental image data
- 3: $Unary \leftarrow \text{zeros}(M.Nlabel, M.Nelement)$ ▷ Initialize Data cost variable as an matrix of zeros
- 4: **if** $M.Dimension == 3$ **then** ▷ 3D Case
- 5: $temp \leftarrow \text{mode}(M.ElementVolume)$ ▷ Evaluate mode of the distribution of element volume
- 6: $Factor \leftarrow \alpha(1 - \exp(-M.ElementVolume/temp))$ ▷ Evaluate Eq (3.2)

```

7:   for  $index1 \leftarrow 1$  to  $M.Nelement$  do                                ▷ Loop over each element
8:        $xpos \leftarrow \text{mean}(M.Coordinate(M.Connectivity(index1,:),1))$   ▷ Coordinates
      of center of the element
9:        $ypos \leftarrow \text{mean}(M.Coordinate(M.Connectivity(index1,:),2))$ 
10:       $zpos \leftarrow \text{mean}(M.Coordinate(M.Connectivity(index1,:),3))$ 
11:       $nx = \text{floor}(xpos \times I.ImageSize(1)/M.MeshSize(1))+1$   ▷ Location of center
      with respect to image grid data
12:       $ny = \text{floor}(ypos \times I.ImageSize(2)/M.MeshSize(2))+1$ 
13:       $nz = \text{floor}(zpos \times I.ImageSize(2)/M.MeshSize(3))+1$ 
14:      for  $index2 \leftarrow 1$  to  $I.Nlabel$  do                            ▷ Loop over each Label value
15:          if  $LabelValues(index2) == I.LabelGridData(nx,ny,nz)$  then
16:               $Unary(index2,index1) \leftarrow 0$                     ▷ No cost if the Label matches the
              image data
17:          else
18:               $Unary(index2,index1) \leftarrow Factor(index1)$       ▷ Add cost determined
              by Eq(3.2)
19:          end if
20:      end for
21:  end for
22: else
23:     $temp \leftarrow \text{mode}(M.ElementArea)$                         ▷ Evaluate mode of the distribution of
      element area
24:     $Factor \leftarrow \alpha(1 - \exp(-M.ElementArea/temp))$         ▷ Evaluate Eq (3.2)
25:    for  $index1 \leftarrow 1$  to  $M.Nelement$  do                            ▷ Loop over each element
26:         $xpos \leftarrow \text{mean}(M.Coordinate(M.Connectivity(index1,:),1))$   ▷ Coordinates
        of center of the element
27:         $ypos \leftarrow \text{mean}(M.Coordinate(M.Connectivity(index1,:),2))$ 

```

```

28:          $nx = \text{floor}(xpos \times I.ImageSize(1) / M.MeshSize(1)) + 1$   $\triangleright$  Location of center
        with respect to image grid data
29:          $ny = \text{floor}(ypos \times I.ImageSize(2) / M.MeshSize(2)) + 1$ 
30:         for  $index2 \leftarrow 1$  to  $I.Nlabel$  do  $\triangleright$  Loop over each Label value
31:             if  $LabelValues(index2) == I.LabelGridData(nx, ny, nz)$  then
32:                  $Unary(index2, index1) \leftarrow 0$   $\triangleright$  No cost if the Label matches the
                image data
33:             else
34:                  $Unary(index2, index1) \leftarrow Factor(index1)$   $\triangleright$  Add cost determined
                by Eq(3.2)
35:             end if
36:         end for
37:     end for
38: end if
39: return  $Unary = 0$ 

```

Algorithm 3 Set-up Smooth cost (2D/3D)

```

1:  $M \leftarrow$  Mesh data  $\triangleright$  Import mesh data
2:  $I \leftarrow$  Image data  $\triangleright$  Import Experimental image data
3:  $Pairwise \leftarrow \text{zeros}(Nelement)$   $\triangleright$  Initialize Smooth cost variable as a square
    matrix of zeros
4: if  $M.Dimension == 3$  then  $\triangleright$  3D Case
5:     for  $index \leftarrow 1$  to  $M.Npair$  do
6:          $paircost \leftarrow g(M.FaceNormal(index)) \times M.FaceArea(index)$   $\triangleright$  Evaluate
            Eq((3.3))
7:          $Pairwise(M.Neighbor(index, 1), M.Neighbor(index, 2)) \leftarrow paircost$   $\triangleright$ 
            Update  $Pairwise$  variable
8:     end for

```

```

9: else ▷ 2D Case
10:   for  $index \leftarrow 1$  to  $M.Npair$  do
11:      $paircost \leftarrow g(M.EdgeNormal(index)) \times M.EdgeArea(index)$  ▷ Evaluate
        $Eq((3.3))$ 
12:      $Pairwise(M.Neighbor(index,1), M.Neighbor(index,2)) \leftarrow paircost$  ▷
       Update  $Pairwise$  variable
13:   end for
14: end if
15: return  $Pairwise$ 

```

Algorithm 4 Using GCO Library for estimating the labeling

```

1: Input:  $Unary, Pairwise, LabelValues, NLabel, Nelement$ 
2:  $h \leftarrow GCO\_Create(NElement, NLabel)$ 
3:  $GCO\_SetDataCost(h, Unary)$ 
4:  $Labelcost \leftarrow ones(Nlabel) - Id(Nlabel)$ 
5:  $GCO\_SetSmoothCost(h, Labelcost)$ 
6:  $GCO\_SetNeighbors(h, Pairwise)$ 
7:  $GCO\_Expansion(h)$ 
8:  $Label \leftarrow LabelValues(GCO\_GetLabeling(h))$ 
9: return  $Label$ 
10: end

```

B.2 Modeling microstructural fracture

The data structure used in the algorithm is presented. There are two kinds of data that are required in this algorithm: (1) FE Mesh data and (2) Image data. It is recommended to calculate and save all the variables as defined below at the time of mesh generation and be passed on to the mesh labeling algorithm.

FE Mesh Data: The basic structure for any FE Mesh includes folling bare-bone

data:

- *Nnode*: Number of nodes of the mesh
- *Coordinate*: Location of each node of the mesh
- *Nelement*: Number of elements
- *Connectivity*: Contains tuples with nodes of each element

It is post-processed to include the following derived quantities:

- *ElementCenter*: Coordinates of a point in the interior of the element (possibly centroid)
- *Neighbor*: It contains each pair of elements which share an edge.
- *Npair*: Number of Neighbors
- *ElementArea*: Area of each element
- *Edge*: It contains tuples with nodes of shared edges (indexed as the *Neighbor* data)
- *EdgeLength*: Length of each edge defined in *Edge* data
- *EdgeNormal*: Unit Normal of each edge defined in *Edge* data

Finally, the smooth labeling procedure [153] provides the *GrainID* of each element in the mesh, based on the experimental image.

Material Property Data: The Polycrystalline material properties are stored the form of the following variables:

- *GrainOrientation*: Axis-Angle representation for each value of GrainID

- *SurfaceEnergyParameter*: Surface energy parameters for each value of GrainID
- *ShearModulus*: Effective Shear Modulus of the specimen
- *PoissonRatio*: Effective Poisson's Ratio of the specimen
- *IntergranularEnergy*: γ^{ig} for each pair of grains

Next the basic Pseudocode modules are presented as used in 4.3.2:

Algorithm 5 Set-up Neighbor Cost

```

1:  $M \leftarrow$  Mesh data ▷ Import mesh data
2:  $P \leftarrow$  Material Property Data ▷ Import Material Property Data
3:  $NC \leftarrow \text{zeros}(Nelement)$  ▷ Initialize Smooth cost variable as a square matrix of zeros
4:  $dihedral \leftarrow \text{zeros}(Npair, 1)$  ▷ Initialize Smooth cost variable as a square matrix of zeros
5: for  $i \leftarrow 1$  to  $M.Npair$  do ▷ Loop over all shared edges
6:    $Grain1 \leftarrow M.GrainID(M.Neighbor(i, 1))$ 
7:    $Grain2 \leftarrow M.GrainID(M.Neighbor(i, 2))$ 
8:   if  $Grain1 == Grain2$  then ▷ Transgranular edge
9:      $dihedral(i) \leftarrow \psi_D$  (evaluated using Eq(4.12))
10:     $SurfaceDensity \leftarrow \tilde{\gamma}$  (evaluated using Eq(4.13))
11:   else ▷ Intergranular edge
12:      $SurfaceDensity \leftarrow P.IntergranularEnergy(Grain1, Grain2)$ 
13:   end if
14:    $NC(M.Neighbor(i, 1), M.Neighbor(i, 2)) \leftarrow 2SurfaceDensity \times M.EdgeLength(i)$ 
▷ Update Neighbor Cost
15: end for
16: return  $NC, dihedral$ 

```

Algorithm 6 Set-up Data Cost

```
1: Input:  $\bar{r}$ ,  $\sigma_T$ ,  $\beta$ ,  $a$   $\triangleright$  Input domain radius, Tensile stress, crack angle and length
2:  $M \leftarrow$  Mesh data  $\triangleright$  Import mesh data
3:  $P \leftarrow$  Material Property Data  $\triangleright$  Import Material Property Data
4:  $K_I, K_{II} \leftarrow$  Evaluate Eq(4.14) using argument  $\beta$ 
5:  $R\text{-Index} \leftarrow (\text{distance}(M.ElementCenter, Cracktip)_i \bar{r}) \triangleright$  Index of elements within
   domain defined by  $\bar{r}$ 
6:  $(r_i, \theta_i)_{i \in R\text{-Index}} = \text{cartesian2polar}(M.ElementCenter - Cracktip) \triangleright$  Find Polar
   coordinate of each element
7: for  $i \in R\text{-Index}$  do
8:    $G'(\theta_i) \leftarrow$  Evaluate Eq(4.8) using arguments  $K_I, K_{II}, P.ShearModulus$  and
      $P.PoissonRatio$ .
9:    $g(i, \cdot) \leftarrow$  Evaluate Eq(4.7)
10:   $DC_0(\cdot, i) \leftarrow$  Evaluate Eq(4.17) using arguments  $g(i, \cdot), r_i, M.ElementArea(i)$ 
11: end for
12:  $UpperBoundaryIndex \leftarrow (M.ElementCenter \in \text{UpperBoundary})$ 
13:  $LowerBoundaryIndex \leftarrow (M.ElementCenter \in \text{LowerBoundary})$ 
14:  $DC_{BC}(2, UpperBoundaryIndex) \leftarrow \text{LargeNumber}$ 
15:  $DC_{BC}(1, LowerBoundaryIndex) \leftarrow \text{LargeNumber}$ 
16: return  $DC_0, DC_{BC}, R\text{-Index}$ 
```

Algorithm 7 Estimate Crack Path

```
1: Input:  $DC_0, DC_{BC}, NC, R\text{-Index}$   $\triangleright$  Input variables
2:  $h \leftarrow \text{GCO\_Create}(N\text{Nodes} = \text{length}(R\text{-Index}), N\text{Label}=2)$ 
3:  $\text{GCO\_SetDataCost}(h, DC_0 + DC_{BC})$ 
4:  $LC \leftarrow \text{ones}(2) - \text{Id}(2)$ 
5:  $\text{GCO\_SetSmoothCost}(h, LC)$ 
6:  $\text{GCO\_SetNeighbors}(h, NC(R\text{-Index}, R\text{-Index})) \triangleright$  Use Neighbor cost only for
```

vertices in radial domain

7: GCO.Expansion(h)

8: $Label \leftarrow \text{GCO_GetLabeling}(h)$ $\triangleright$ This outputs label in $\{1,2\}$ format. Change it to $\{+1,-1\}$ format, if desired.

9: **return** $Label$

Algorithm 8 Evaluate total energy

1: **Input:** $DC_0, NC, Label, R\text{-Index}$

2: $EnergyRelease \leftarrow -\left(\sum_j DC(Label, j)\right)$

3: $x \leftarrow (Label == 1)$ $\triangleright$ TypeCast bool to float type for next calculation

4: $y \leftarrow (Label == 2)$ $\triangleright$ Or equivalently, $Label == -1$. TypeCast bool to float type

5: $NC_R \leftarrow NC(R\text{-Index}, R\text{-Index})$ $\triangleright$ Use Neighbor cost only for vertices in radial domain

6: $SurfaceEnergy \leftarrow \sum_{ij} x_i y_j (NC_R)_{ij}$

7: **return** $\varepsilon(\Gamma) \leftarrow SurfaceEnergy - EnergyRelease$

B.3 Multiphase flow

Only the pseudo-codes for Quadtree method is presented. The pixelated case can be treated as a specific instance of the Quadtree method with only one layer. Computational cost in this procedure can be reduced by pre-processing and saving the complete quadtree structure before solving the problem. This information can be saved in form of following variables:

- *LayerSize*: The size of i^{th} layer from top is given as $4^{i-1} \times N_d^2$.
- N_T : Total number of nodes, estimated as $\sum_{i=1}^{N_p} (LayerSize)$.
- *LayerIndex*: Nodes are enumerated sequentially starting from top layer to bottom layer. The nodes in the i^{th} layer ranges from $LayerIndex(i) + 1$ to $LayerIndex(i) + LayerSize(i)$.

- *Child*: List of children of each node. Nodes in bottom layer have no child.
- *Parent*: List of parent of each node. Nodes in top layer have no parent.
- *Descendent*: List of all children, children of children and so on for each node.
- *NumberDescendent*: Number of descendant of each node
- *LeafChild*: List of descendants that are in the bottom layer.
- *NumberLeafChild*: Number of descendants of each node.
- $\overline{\mathcal{N}}$: A list of neighbors of each node (within the same layer) including the node itself. This neighborhood is used to locate regions with similar labels. The set of neighbors for any node can be determined by a user prescribed strategy and it need not be same as the neighbors in the Crofton's relation.
- *Coord*: Coordinates of each node.
- *DistMap*: Square matrix with distance between each pixel.
- *Adjacency*: A block diagonal matrix with each block being the adjacency of the connections within the same layer. These connections are made using the Crofton's stencil, $\mathcal{N}_R$.

The multiphase flow is simulated by successively iterating over each time-step. The pseudocode for each time-step is presented in Algorithm 10. As before, the optimization is carried out using the GCO library developed in [116]. The pseudocode for using this library is presented in Chapter III.

Algorithm uses following important temporary variables. *NodeType*: is an array taking values in $\{-1, 0, 1\}$ for each node, v of the Quadtree. $NodeType(v)=1$ if the descendants of v and its neighbors have the same label but v 's parent does not have

this property, $NodeType(v)=0$ if one of its ancestor is of $NodeType=1$, $NodeType(v)=-1$ if one of its descendants is of $NodeType=1$ *NodeLabel*: is an array which assign a label to each node of graph. The label of last layer's nodes are prescribed/estimated by the algorithm. The label of nodes with $NodeType=1$ and 0 is assigned based on its respective *LeafChild*. The label of nodes with $NodeType=2$ is assigned arbitrarily. *ActiveParentNode*: is an array which is initialized as the index of each node. If the $NodeType$ of a node is 0, then the *ActiveParentNode* of that node is set as its ancestor's index which is of $NodeType=1$.

Algorithm 9 EstimateCost

```

1: Input: CurrLabel,  $\Delta t$ 
2:  $NodeType \leftarrow \mathbf{1}_{N_t \times 1}$  ▷ Initialize variables
3:  $ActiveNodeLabel \leftarrow \mathbf{0}_{N_T \times 1}$ 
4:  $ActiveParentNode \leftarrow [1, \dots, N_T]^T$ 
5: for  $i = 1 : N_p - 1$  do
6:   for  $j = 1 : LayerSize(i)$  do
7:      $CurrNode \leftarrow LayerIndex(i) + j$ 
8:     if  $NodeType(CurrNode) == 1$  then
9:       if Leaves of  $CurrNode$ 's neighbors have same label then
10:         Set  $NodeType$  of  $CurrNode$ 's descendant as 0
11:         Set  $ActiveParentNode$  of  $CurrNode$ 's descendants as  $CurrNode$ 
12:         Set  $ActiveNodeLabel$  of  $CurrNode$  and descendants as the leaf's label
13:       else
14:         Set  $NodeType$  of  $CurrNode$  as  $-1$ 
15:       end if
16:     end if
17:   end for

```

```

18: end for
19:  $ActiveNodeIndex \leftarrow$  Index of nodes with  $NodeType=1$ 
20:  $InactiveChildIndex \leftarrow$  Index of nodes with  $NodeType=0$ 
21: Set  $NewIndex$  to map  $ActiveNodeIndex$  to  $\{1, \dots, |ActiveNodeIndex|\}$ 
22:  $Pairwise \leftarrow Adjacency(ActiveNodeIndex, ActiveNodeIndex)$   $\triangleright$  Initiate  $Pairwise$ 
    cost
23: for  $v \in InactiveChildIndex$  do
24:    $temp \leftarrow Adjacency(ActiveNodeIndex, v)$ 
25:    $vParentIndex \leftarrow NewIndex(ActiveParentNode(v))$ 
26:    $Pairwise(:, vParentIndex) += temp$ 
27:    $Pairwise(vParentIndex, :) += temp^T$ 
28: end for
29:  $Unary \leftarrow \mathbf{0}_{N_L \times N_T}$ 
30: for  $i \in [1, \dots, N_L]$  do
31:    $CurrNodes \leftarrow (CurrLabel == l_i)$ 
32:    $NotCurrNodesIndex \leftarrow$  Index of nodes s.t. with  $CurrNodes==0$ 
33:   if there is atleast one node with label  $\ell$  then
34:     for  $v \in NotCurrNodesIndex$  do
35:        $Unary(\ell, v) \leftarrow Area(v) \times (f(\ell) + \min DistMap(CurrNodes, v))$   $\triangleright$ 
        Define  $p$  (Eq(5.5)) as the minimum distance to the boundary  $\Gamma$ 
36:     end for
37:   end if
38: end for
39: return  $Unary, Pairwise, ActiveNodeIndex$ 

```

Algorithm 10 Iterative step for Multiphase flow

- 1: **Input:** $CurrLabel, \Delta t$
- 2: $Unary, Pairwise, ActiveNodeIndex \leftarrow EstimateCost(CurrLabel, \Delta t)$

- 3: $NewActiveNodeLabel \leftarrow$ minimize using GCO: *Unary, Pairwise* $\triangleright$ Labeling step
on Quadtree graph with reduced nodes
- 4: Estimate $NewLabel$ from $NewActiveNodeLabel$ $\triangleright$ Transpose the Labels of Active
node to the respective leaves
- 5: **if** $NewLabel == CurrLabel$ **then**
- 6: Increase Δt and goto step 2
- 7: **end if**
- 8: **return** $NewLabel, \Delta t$

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Jacob Adams. *Investigating microstructural effects on short crack growth and fatigue life behavior of WE43 Magnesium*. PhD thesis, University of Michigan, Ann Arbor, 2018.
- [2] Michael Sangid, John F Matlik, Akin Keskin, Ben H Thacker, Barron J Bichon, Dale L Ball, Stephen P Engelstad, Charles Ward, Vasisht Venkatesh, H A Kim, et al. Integrating icme practices into design systems and structural analysis. In *55th AIAA Aerospace Sciences Meeting*, page 0874, 2017.
- [3] Wei Xu, Edward W Lui, Aaron Pateras, Ma Qian, and MJAM Brandt. In situ tailoring microstructure in additively manufactured ti-6al-4v for superior mechanical performance. *Acta Materialia*, 125:390–400, 2017.
- [4] Patrick Köhnen, Maike Letang, Maximilian Voshage, Johannes H Schleifenbaum, and Christian Haase. Understanding the process-microstructure correlations for tailoring the mechanical properties of l-pbf produced austenitic advanced high strength steel. *Additive Manufacturing*, 30:100914, 2019.
- [5] Carolin Körner, Harald Helmer, Andreas Bauereiß, and Robert F Singer. Tailoring the grain structure of in718 during selective electron beam melting. In *MATEC Web of Conferences*, volume 14, page 08001. EDP Sciences, 2014.
- [6] Tomasz Kurzynowski, Konrad Gruber, Wojciech Stopyra, Bogumiła Kuźnicka, and Edward Chlebus. Correlation between process parameters, microstructure and properties of 316 l stainless steel processed by selective laser melting. *Materials Science and Engineering: A*, 718:64–73, 2018.
- [7] Narendran Raghavan, Ryan Dehoff, Sreekanth Pannala, Srdjan Simunovic, Michael Kirka, John Turner, Neil Carlson, and Sudarsanam S Babu. Numerical modeling of heat-transfer and the influence of process parameters on tailoring the grain morphology of in718 in electron beam additive manufacturing. *Acta Materialia*, 112:303–314, 2016.
- [8] Wentao Yan, Stephen Lin, Orion L Kafka, Yanping Lian, Cheng Yu, Zeliang Liu, Jinhui Yan, Sarah Wolff, Hao Wu, Ebot Ndip-Agbor, et al. Data-driven multi-scale multi-physics models to derive process–structure–property relationships for additive manufacturing. *Computational Mechanics*, 61(5):521–541, 2018.

- [9] Bob Baoping He. Introduction to two-dimensional x-ray diffraction. *Powder diffraction*, 18(2):71–85, 2003.
- [10] Pavel Krakhmalev, Gunnel Fredriksson, Krister Svensson, Igor Yadroitsev, Ina Yadroitsava, Mattias Thuvander, and Ru Peng. Microstructure, solidification texture, and thermal stability of 316 l stainless steel manufactured by laser powder bed fusion. *Metals*, 8(8):643, 2018.
- [11] Qiang Zhang, Jing Chen, Pengfei Guo, Hua Tan, Xin Lin, and Weidong Huang. Texture and microstructure characterization in laser additive manufactured ti–6al–2zr–2sn–3mo–1.5 cr–2nb titanium alloy. *Materials & Design*, 88:550–557, 2015.
- [12] Hong Qin, Vahid Fallah, Qingshan Dong, Mathieu Brochu, Mark R Daymond, and Mark Gallerneault. Solidification pattern, microstructure and texture development in laser powder bed fusion (lpbf) of al10sing alloy. *Materials Characterization*, 145:29–38, 2018.
- [13] Michael A Groeber, BK Haley, Michael D Uchic, Dennis M Dimiduk, and Somnath Ghosh. 3d reconstruction and characterization of polycrystalline microstructures using a fib–sem system. *Materials Characterization*, 57(4-5):259–273, 2006.
- [14] RT DeHoff. Quantitative serial sectioning analysis: preview. *Journal of Microscopy*, 131(3):259–263, 1983.
- [15] Dunyong Deng, Ru Lin Peng, Håkan Brodin, and Johan Moverare. Microstructure and mechanical properties of inconel 718 produced by selective laser melting: Sample orientation dependence and effects of post heat treatments. *Materials Science and Engineering: A*, 713:294–306, 2018.
- [16] Gene E Ice and Bennett C Larson. Three-dimensional x-ray structural microscopy using polychromatic microbeams. *MRS bulletin*, 29(3):170–176, 2004.
- [17] Michael Groeber, Somnath Ghosh, Michael D Uchic, and Dennis M Dimiduk. A framework for automated analysis and simulation of 3d polycrystalline microstructures.: Part 1: Statistical characterization. *Acta Materialia*, 56(6):1257–1273, 2008.
- [18] DJ Rowenhorst, A Gupta, CR Feng, and G Spanos. 3d crystallographic and morphological analysis of coarse martensite: Combining ebsd and serial sectioning. *Scripta Materialia*, 55(1):11–16, 2006.
- [19] Jonathan E Spowart. Automated serial sectioning for 3-d analysis of microstructures. *Scripta Materialia*, 55(1):5–10, 2006.

- [20] David J Rowenhorst, Lily Nguyen, Aerial D Murphy-Leonard, and Richard W Fonda. Characterization of microstructure in additively manufactured 316l using automated serial sectioning. *Current Opinion in Solid State and Materials Science*, 24(3):100819, 2020.
- [21] Henning F Poulsen, Dorte Juul Jensen, and Gavin BM Vaughan. Three-dimensional x-ray diffraction microscopy using high-energy x-rays. *MRS bulletin*, 29(3):166–169, 2004.
- [22] Jean-Yves Buffiere, Emilie Ferrie, Henry Proudhon, and Wolfgang Ludwig. Three-dimensional visualisation of fatigue cracks in metals using high resolution synchrotron x-ray micro-tomography. *Materials science and technology*, 22(9):1019–1024, 2006.
- [23] Eric Maire and Philip John Withers. Quantitative x-ray tomography. *International materials reviews*, 59(1):1–43, 2014.
- [24] SA McDonald, P Reischig, C Holzner, EM Lauridsen, PJ Withers, AP Merkle, and M Feser. Non-destructive mapping of grain orientations in 3d by laboratory x-ray microscopy. *Scientific reports*, 5:14665, 2015.
- [25] Trevor Keller, Greta Lindwall, Supriyo Ghosh, Li Ma, Brandon M Lane, Fan Zhang, Ursula R Kattner, Eric A Lass, Jarred C Heigel, Yaakov Idell, et al. Application of finite element, phase-field, and calphad-based methods to additive manufacturing of ni-based superalloys. *Acta materialia*, 139:244–253, 2017.
- [26] Supriyo Ghosh, Li Ma, Nana Ofori-Opoku, and Jonathan E Guyer. On the primary spacing and microsegregation of cellular dendrites in laser deposited ni–nb alloys. *Modelling and simulation in materials science and engineering*, 25(6):065002, 2017.
- [27] Theron M Rodgers, Jonathan D Madison, and Veena Tikare. Simulation of metal additive manufacturing microstructures using kinetic monte carlo. *Computational Materials Science*, 135:78–89, 2017.
- [28] A Zinoviev, O Zinovieva, V Ploshikhin, V Romanova, and R Balokhonov. Evolution of grain structure during laser additive manufacturing. simulation by a cellular automata method. *Materials & Design*, 106:321–329, 2016.
- [29] Amir Reza Ansari Dezfoli, Weng-Sing Hwang, Wei-Chin Huang, and Tsung-Wen Tsai. Determination and controlling of grain structure of metals after laser incidence: Theoretical approach. *Scientific reports*, 7:41527, 2017.
- [30] Eisuke Miyoshi, Tomohiro Takaki, Munekazu Ohno, Yasushi Shibuta, Shinji Sakane, and Takayuki Aoki. Large-scale phase-field simulation of three-dimensional isotropic grain growth in polycrystalline thin films. *Modelling and Simulation in Materials Science and Engineering*, 27(5):054003, 2019.

- [31] Johannes A Koepf, Martin R Gotterbarm, Matthias Markl, and Carolin Körner. 3d multi-layer grain structure simulation of powder bed fusion additive manufacturing. *Acta Materialia*, 152:119–126, 2018.
- [32] Evdokia Popova, Theron M Rodgers, Xinyi Gong, Ahmet Cecen, Jonathan D Madison, and Surya R Kalidindi. Process-structure linkages using a data science approach: application to simulated additive manufacturing data. *Integrating materials and manufacturing innovation*, 6(1):54–68, 2017.
- [33] Ramin Bostanabad. Reconstruction of 3d microstructures from 2d images via transfer learning. *Computer-Aided Design*, 128:102906, 2020.
- [34] Pinar Acar and Veera Sundararaghavan. A markov random field approach for modeling spatio-temporal evolution of microstructures. *Modelling and Simulation in Materials Science and Engineering*, 24(7):075005, 2016.
- [35] Veera Sundararaghavan. Reconstruction of three-dimensional anisotropic microstructures from two-dimensional micrographs imaged on orthogonal planes. *Integrating Materials and Manufacturing Innovation*, 3(1):19, Jun 2014.
- [36] Iman Javaheri and Veera Sundararaghavan. Polycrystalline microstructure reconstruction using markov random fields and histogram matching. *Computer-Aided Design*, 120:102806, 2020.
- [37] Hang Si. On refinement of constrained delaunay tetrahedralizations. In *Proceedings of the 15th international meshing roundtable*, pages 509–528. Springer, 2006.
- [38] HL Wei, JW Elmer, and T DebRoy. Origin of grain orientation during solidification of an aluminum alloy. *Acta Materialia*, 115:123–131, 2016.
- [39] Yanis Balit, Eric Charkaluk, and Andrei Constantinescu. Digital image correlation for microstructural analysis of deformation pattern in additively manufactured 316l thin walls. *Additive Manufacturing*, 31:100862, 2020.
- [40] HL Wei, JW Elmer, and T DebRoy. Crystal growth during keyhole mode laser welding. *Acta Materialia*, 133:10–20, 2017.
- [41] HL Wei, JW Elmer, and T DebRoy. Three-dimensional modeling of grain structure evolution during welding of an aluminum alloy. *Acta Materialia*, 126:413–425, 2017.
- [42] Vahid Fallah, Masoud Alimardani, Stephen F Corbin, and Amir Khajepour. Temporal development of melt-pool morphology and clad geometry in laser powder deposition. *Computational materials science*, 50(7):2124–2134, 2011.
- [43] Xibing Gong and Kevin Chou. Phase-field modeling of microstructure evolution in electron beam additive manufacturing. *Jom*, 67(5):1176–1182, 2015.

- [44] Hongbo Fu, Yichen Wei, Chiew-Lan Tai, and Long Quan. Sketching hairstyles. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling*, pages 31–36, 2007.
- [45] Kenshi Takayama, Makoto Okabe, Takashi Ijiri, and Takeo Igarashi. Lapped solid textures: filling a model with anisotropic textures. In *ACM SIGGRAPH 2008 papers*, pages 1–9. 2008.
- [46] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470, 2000.
- [47] Arnoldo Badillo and Christoph Beckermann. Phase-field simulation of the columnar-to-equiaxed transition in alloy solidification. *Acta Materialia*, 54(8):2015–2026, 2006.
- [48] Črtomir Donik, Jakob Kraner, Irena Paulin, and Matjaž Godec. Influence of the energy density for selective laser melting on the microstructure and mechanical properties of stainless steel. *Metals*, 10(7):919, 2020.
- [49] George Spanos. Foreword: scripta materialia viewpoint set on 3d characterization and analysis of materials. *Scripta Materialia*, 1(55):3, 2006.
- [50] Dorte Juul Jensen. *Three-Dimensional Orientation Imaging*, pages 91–104. Springer US, Boston, MA, 2000.
- [51] David M Saylor, Joseph Fridy, Bassem S El-Dasher, Kee-Young Jung, and Anthony D Rollett. Statistically representative three-dimensional microstructures based on orthogonal observation sections. *Metallurgical and Materials Transactions A*, 35(7):1969–1979, 2004.
- [52] Michael Groeber, Somnath Ghosh, Michael D Uchic, and Dennis M Dimiduk. A framework for automated analysis and simulation of 3d polycrystalline microstructures.: Part 1: Statistical characterization. *Acta Materialia*, 56(6):1257–1273, 2008.
- [53] Michael Groeber, Somnath Ghosh, Michael D Uchic, and Dennis M Dimiduk. A framework for automated analysis and simulation of 3d polycrystalline microstructures. part 2: Synthetic structure generation. *Acta Materialia*, 56(6):1274–1287, 2008.
- [54] C Zhang, H Li, P Eisenlohr, W Liu, CJ Boehlert, MA Crimp, and TR Bieler. Effect of realistic 3d microstructure in crystal plasticity finite element analysis of polycrystalline ti-5al-2.5 sn. *International Journal of Plasticity*, 69:21–35, 2015.

- [55] Mohammadreza Yaghoobi, Sriram Ganesan, Srihari Sundar, Aaditya Lakshmanan, Shiva Rudraraju, John E Allison, and Veera Sundararaghavan. Prisms-plasticity: An open-source crystal plasticity finite element software. *Computational Materials Science*, 169:109078, 2019.
- [56] J-Y Buffiere, E Maire, C Verdu, P Cloetens, M Pateyron, G Peix, and J Baruchel. Damage assessment in an al/sic composite during monotonic tensile tests using synchrotron x-ray microtomography. *Materials Science and Engineering: A*, 234:633–635, 1997.
- [57] Reeru Pokharel, Jonathan Lind, Anand K Kanjarla, Ricardo A Lebensohn, Shiu Fai Li, Peter Kenesei, Robert M Suter, and Anthony D Rollett. Polycrystal plasticity: comparison between grain-scale observations of deformation and simulations. *Annu. Rev. Condens. Matter Phys.*, 5(1):317–346, 2014.
- [58] Paul G Kotula, Gregory S Rohrer, and Michael P Marsh. Focused ion beam and scanning electron microscopy for 3d materials characterization. *MRS Bulletin*, 39(4):361–365, 2014.
- [59] U. Lienert, S. F. Li, C. M. Hefferan, J. Lind, R. M. Suter, J. V. Bernier, N. R. Barton, M. C. Brandes, M. J. Mills, M. P. Miller, B. Jakobsen, and W.” Pantleon. High-energy diffraction microscopy at the advanced photon source. *JOM*, 63(7):70–77, Jul 2011.
- [60] Donald W. Brown, Levente Balogh, Darrin Byler, Chris M. Hefferan, James F. Hunter, Peter Kenesei, Shiu Fai Li, John Lind, Stephen R. Niezgoda, and Robert M. Suter. Demonstration of near field high energy x-ray diffraction microscopy on high-z ceramic nuclear fuel material. In *Mechanical Stress Evaluation by Neutrons and Synchrotron Radiation VII*, volume 777 of *Materials Science Forum*, pages 112–117. Trans Tech Publications, 4 2014.
- [61] John F Bingert, Robert M Suter, Jonathan Lind, Shiu Fai Li, Reeru Pokharel, and Carl P Trujillo. High-energy diffraction microscopy characterization of spall damage. In *Dynamic Behavior of Materials, Volume 1*, pages 397–403. Springer, 2014.
- [62] Philip Eisenlohr, Pratheek Shanthraj, Brendan R Vande Kieft, Thomas R Bieler, Wenjun Liu, and Ruqing Xu. Subsurface grain morphology reconstruction by differential aperture x-ray microscopy. *JOM*, 69(6):1100–1105, 2017.
- [63] AR Shiveley, PA Shade, AL Pilchak, JS Tiley, and R Kerns. A novel method for acquiring large-scale automated scanning electron microscope data. *Journal of microscopy*, 244(2):181–186, 2011.
- [64] David J Heeger and James R Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. Citeseer, 1995.

- [65] Jeremy S De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 361–368. ACM Press/Addison-Wesley Publishing Co., 1997.
- [66] Eero P Simoncelli and Javier Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proc 5th IEEE Int'l Conf on Image Proc*, volume 1, 1998.
- [67] V Sundararaghavan and N Zabaras. Classification and reconstruction of three-dimensional microstructures using support vector machines. *Computational Materials Science*, 32(2):223 – 239, 2005.
- [68] Salvatore Torquato. *Random heterogeneous materials: microstructure and macroscopic properties*, volume 16. Springer Science & Business Media, 2013.
- [69] CLY Yeong and Salvatore Torquato. Reconstructing random media. *Physical Review E*, 57(1):495, 1998.
- [70] Yang Jiao, Eric Padilla, and Nikhilesh Chawla. Modeling and predicting microstructure evolution in lead/tin alloy via correlation functions and stochastic material reconstruction. *Acta Materialia*, 61(9):3370–3377, 2013.
- [71] David T Fullwood, Stephen R Niezgod, and Surya R Kalidindi. Microstructure reconstructions from 2-point statistics using phase-recovery algorithms. *Acta Materialia*, 56(5):942–948, 2008.
- [72] Hongyi Xu, Dmitriy A Dikin, Craig Burkhart, and Wei Chen. Descriptor-based methodology for statistical characterization and 3d reconstruction of microstructural materials. *Computational Materials Science*, 85:206–216, 2014.
- [73] Tristan N Ashton, Donna Post Guillen, and William H Harris. An algorithm to generate synthetic 3d microstructures from 2d exemplars. *JOM*, pages 1–10, 2019.
- [74] Celal Soyarslan, Swantje Bargmann, Marc Pradas, and Jörg Weissmüller. 3d stochastic bicontinuous microstructures: Generation, topology and elasticity. *Acta materialia*, 149:326–340, 2018.
- [75] AD Rollett, D Saylor, J Fridy, BS El-Dasher, A Brahme, S-B Lee, C Cornwell, and R Noack. Modeling polycrystalline microstructures in 3d. In *AIP Conference Proceedings*, volume 712, pages 71–77. AIP, 2004.
- [76] Anthony D Rollett, S-B Lee, R Campman, and GS Rohrer. Three-dimensional characterization of microstructure by electron back-scatter diffraction. *Annu. Rev. Mater. Res.*, 37:627–658, 2007.

- [77] 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):5, 2014.
- [78] Xiaohui Tu, Ahmad Shahba, Jinlei Shen, and Somnath Ghosh. Microstructure and property based statistically equivalent rves for polycrystalline-polyphase aluminum alloys. *International Journal of Plasticity*, 115:268–292, 2019.
- [79] A Brahme, MH Alvi, D Saylor, J Fridy, and AD Rollett. 3d reconstruction of microstructure in a commercial purity aluminum. *Scripta Materialia*, 55(1):75–80, 2006.
- [80] Simone Falco, Jiawei Jiang, Francesco De Cola, and Nik Petrinic. Generation of 3d polycrystalline microstructures with a conditioned laguerre-voronoi tessellation technique. *Computational Materials Science*, 136:20–28, 2017.
- [81] Swantje Bargmann, Benjamin Klusemann, Jürgen Markmann, Jan Eike Schnabel, Konrad Schneider, Celal Soyarslan, and Jana Wilmers. Generation of 3d representative volume elements for heterogeneous materials: A review. *Progress in Materials Science*, 96:322–384, 2018.
- [82] Tarasankar DebRoy, HL Wei, JS Zuback, T Mukherjee, JW Elmer, JO Milewski, Allison Michelle Beese, A Wilson-Heid, A De, and W Zhang. Additive manufacturing of metallic components—process, structure and properties. *Progress in Materials Science*, 92:112–224, 2018.
- [83] Ramin Bostanabad, Yichi Zhang, Xiaolin Li, Tucker Kearney, L Catherine Brinson, Daniel W Apley, Wing Kam Liu, and Wei Chen. Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques. *Progress in Materials Science*, 95:1–41, 2018.
- [84] David T Fullwood, Stephen R Niezgoda, Brent L Adams, and Surya R Kalidindi. Microstructure sensitive design for performance optimization. *Progress in Materials Science*, 55(6):477–562, 2010.
- [85] A Kumar, L Nguyen, M DeGraef, and V Sundararaghavan. A markov random field approach for microstructure synthesis. *Modelling and Simulation in Materials Science and Engineering*, 24(3):035015, 2016.
- [86] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999.
- [87] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 795–802. ACM, 2005.

- [88] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. Solid texture synthesis from 2d exemplars. In *ACM transactions on graphics (TOG)*, volume 26, page 2. ACM, 2007.
- [89] Pinar Acar and Veera Sundararaghavan. A markov random field approach for modeling spatio-temporal evolution of microstructures. *Modelling and Simulation in Materials Science and Engineering*, 24(7):075005, 2016.
- [90] Ramin Bostanabad, Anh Tuan Bui, Wei Xie, Daniel W Apley, and Wei Chen. Stochastic microstructure characterization and reconstruction via supervised learning. *Acta Materialia*, 103:89–102, 2016.
- [91] Xingchen Liu and Vadim Shapiro. Random heterogeneous materials via texture synthesis. *Computational Materials Science*, 99:177–189, 2015.
- [92] David M Turner and Surya R Kalidindi. Statistical construction of 3-d microstructures from 2-d exemplars collected on oblique sections. *Acta Materialia*, 102:136–148, 2016.
- [93] Dinu Coltuc, Philippe Bolon, and J-M Chassery. Exact histogram specification. *IEEE Transactions on Image Processing*, 15(5):1143–1152, 2006.
- [94] Dori Shapira, Shai Avidan, and Yacov Hel-Or. Multiple histogram matching. In *2013 IEEE International Conference on Image Processing*, pages 2269–2273. IEEE, 2013.
- [95] Shang Sun and Veera Sundararaghavan. A probabilistic crystal plasticity model for modeling grain shape effects based on slip geometry. *Acta Materialia*, 60(13):5233 – 5244, 2012.
- [96] Marco A. Lopez-Sanchez and Sergio Llana-Fúnez. An extension of the saltykov method to quantify 3d grain size distributions in mylonites. *Journal of Structural Geology*, 93:149 – 161, 2016.
- [97] Veera Sundararaghavan and Nicholas Zabaras. Linear analysis of texture?property relationships using process-based representations of rodrigues space. *Acta Materialia*, 55(5):1573 – 1587, 2007.
- [98] Pinar Acar and Veera Sundararaghavan. Utilization of a linear solver for multiscale design and optimization of microstructures. *AIAA Journal*, 54(5):1751 – 1759, 2016.
- [99] T Quested. Micrograph 712 available under creative commons license. https://www.doitpoms.ac.uk/miclib/micrograph_record.php?id=712, 2003. [Online; accessed 19-July-2019].
- [100] V. Sundararaghavan, A. Kumar, and S. Sun. Crystal plasticity simulations using nearest neighbor orientation correlation function. *Acta Materialia*, 93:12 – 23, 2015.

- [101] Veera Sundararaghavan and Nicholas Zabaras. A statistical learning approach for the design of polycrystalline materials. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 1(5):306–321, 2009.
- [102] Pinar Acar and Veera Sundararaghavan. Utilization of a linear solver for multiscale design and optimization of microstructures. *AIAA Journal*, pages 1751–1759, 2016.
- [103] Shang Sun and Veera Sundararaghavan. A probabilistic crystal plasticity model for modeling grain shape effects based on slip geometry. *Acta Materialia*, 60(13-14):5233–5244, 2012.
- [104] S. Sun and V. Sundararaghavan. A peridynamic implementation of crystal plasticity. *International Journal of Solids and Structures*, 51(19):3350 – 3360, 2014.
- [105] Veera Sundararaghavan and Siddhartha Srivastava. Microfract: An image based code for microstructural crack path prediction. *SoftwareX*, 6:94–97, 2017.
- [106] Adam J Schwartz, Mukul Kumar, Brent L Adams, and David P Field. *Electron backscatter diffraction in materials science*. Springer, 2014.
- [107] Guodong Fang, Bassam El Said, Dmitry Ivanov, and Stephen R. Hallett. Smoothing artificial stress concentrations in voxel-based models of textile composites. *Composites Part A: Applied Science and Manufacturing*, 80:270 – 284, 2016.
- [108] Romain Quey, PR Dawson, and Fabrice Barbe. Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, 200(17-20):1729–1745, 2011.
- [109] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2):291–307, 1970.
- [110] Charles M Fiduccia and Robert M Mattheyses. A linear-time heuristic for improving network partitions. In *Papers on Twenty-five years of electronic design automation*, pages 241–247. ACM, 1988.
- [111] Peter Sanders and Christian Schulz. High quality graph partitioning. *Graph Partitioning and Graph Clustering*, 588(1), 2012.
- [112] Mark EJ Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822, 2013.
- [113] Aydın Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In *Algorithm Engineering*, pages 117–158. Springer, 2016.

- [114] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.
- [115] Peter Sanders and Christian Schulz. Think locally, act globally: Highly balanced graph partitioning. In *International Symposium on Experimental Algorithms*, pages 164–175. Springer, 2013.
- [116] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.
- [117] Lawrence E Murr, Sara M Gaytan, Diana A Ramirez, Edwin Martinez, Jennifer Hernandez, Krista N Amato, Patrick W Shindo, Francisco R Medina, and Ryan B Wicker. Metal fabrication by additive manufacturing using laser and electron beam melting technologies. *Journal of Materials Science & Technology*, 28(1):1–14, 2012.
- [118] Dirk Herzog, Vanessa Seyda, Eric Wycisk, and Claus Emmelmann. Additive manufacturing of metals. *Acta Materialia*, 117:371–392, 2016.
- [119] Bernd Baufeld, Omer Van der Biest, and Rosemary Gault. Additive manufacturing of ti–6al–4v components by shaped metal deposition: microstructure and mechanical properties. *Materials & Design*, 31:S106–S111, 2010.
- [120] Roland Lvovíc Dobrushin, Roman Kotecký, and Senya Shlosman. *Wulff construction: a global shape from local interaction*, volume 104. American Mathematical Society Providence, 1992.
- [121] Frank R. Schmidt, Eno Töppe, and Daniel Cremers. Efficient planar graph cuts with applications in computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, Jun 2009.
- [122] A Kelly, WR Tyson, and AH Cottrell. Ductile and brittle crystals. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics*, 15(135):567–586, 1967.
- [123] Michael Ortiz and Anna Pandolfi. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *International journal for numerical methods in engineering*, 44(9):1267–1282, 1999.
- [124] Igor Simonovski and Leon Cizelj. Cohesive zone modeling of intergranular cracking in polycrystalline aggregates. *Nuclear Engineering and Design*, 283:139–147, 2015.
- [125] M Samimi, JAW Van Dommelen, and MGD Geers. An enriched cohesive zone model for delamination in brittle interfaces. *International journal for numerical methods in engineering*, 80(5):609–630, 2009.

- [126] Yan Li, DL McDowell, and Min Zhou. A multiscale framework for predicting fracture toughness of polycrystalline metals. *Materials Performance and Characterization*, 3(3):157–172, 2014.
- [127] YF Gao and AF Bower. A simple technique for avoiding convergence problems in finite element simulations of crack nucleation and growth on cohesive interfaces. *Modelling and Simulation in Materials Science and Engineering*, 12(3):453, 2004.
- [128] Charlotte Kuhn and Ralf Müller. A continuum phase field model for fracture. *Engineering Fracture Mechanics*, 77(18):3625–3634, 2010.
- [129] Vincent Hakim and Alain Karma. Laws of crack motion and phase-field models of fracture. *Journal of the Mechanics and Physics of Solids*, 57(2):342–368, 2009.
- [130] Christian Miehe, Fabian Welschinger, and Martina Hofacker. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field fe implementations. *International Journal for Numerical Methods in Engineering*, 83(10):1273–1311, 2010.
- [131] S Murakami and Y Liu. Mesh-dependence in local approach to creep fracture. *International Journal of Damage Mechanics*, 4(3):230–250, 1995.
- [132] Arezoo Emdadi and Mohsen Asle Zaeem. Phase-field modeling of crack propagation in polycrystalline materials. *Computational Materials Science*, 186:110057, 2021.
- [133] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1):131–150, 1999.
- [134] Javier Oliver, Alfredo Edmundo Huespe, and Pablo Javier Sánchez. A comparative study on finite elements for capturing strong discontinuities: E-fem vs x-fem. *Computer methods in applied mechanics and engineering*, 195(37-40):4732–4752, 2006.
- [135] Shardul Panwar, Shang Sun, and Veera Sundararaghavan. Modeling fatigue failure using the variational multiscale method. *Engineering Fracture Mechanics*, 162:290–308, 2016.
- [136] Shang Sun and Veera Sundararaghavan. Modeling crack propagation in polycrystalline microstructure using variational multiscale method. *Mathematical Problems in Engineering*, 2016(4715696):1 – 14, 2016.
- [137] Shang Sun, Ali Ramazani, and Veera Sundararaghavan. A hybrid multi-scale model of crystal plasticity for handling stress concentrations. *Metals*, 7(9):345, 2017.

- [138] Gilles A Francfort and J-J Marigo. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids*, 46(8):1319–1342, 1998.
- [139] Gowri Srinivasan, Jeffrey D Hyman, David A Osthus, Bryan A Moore, Daniel O’Malley, Satish Karra, Esteban Rougier, Aric A Hagberg, Abigail Hunter, and Hari S Viswanathan. Quantifying topological uncertainty in fractured systems using graph theory and machine learning. *Scientific reports*, 8(1):1–11, 2018.
- [140] Abigail Hunter, Bryan A Moore, Maruti Mudunuru, Viet Chau, Roselyne Tchoua, Chandramouli Nyshadham, Satish Karra, Daniel O’Malley, Esteban Rougier, Hari Viswanathan, et al. Reduced-order modeling through machine learning and graph-theoretic approaches for brittle fracture applications. *Computational Materials Science*, 157:87–98, 2019.
- [141] Frederic E. Bock, Roland C. Aydin, Christian J. Cyron, Norbert Huber, Surya R. Kalidindi, and Benjamin Klusemann. A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials*, 6:110, 2019.
- [142] Shmuel Osovski, Alan Needleman, and Ankit Srivastava. Intergranular fracture prediction and microstructure design. *International Journal of Fracture*, 216(2):135–148, 2019.
- [143] Abhilash Molkeri, Ankit Srivastava, Shmuel Osovski, and Alan Needleman. Influence of Grain Size Distribution on Ductile Intergranular Crack Growth Resistance. *Journal of Applied Mechanics*, 87(3), 11 2019. 031008.
- [144] Vladimir Kolmogorov and Ramin Zabini. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–159, 2004.
- [145] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
- [146] George C Sih. Strain-energy-density factor applied to mixed mode crack problems. *International Journal of fracture*, 10(3):305–321, 1974.
- [147] George C Sih. *Mechanics of fracture initiation and propagation: surface and volume energy density applied as failure criterion*, volume 11. Springer Science & Business Media, 2012.
- [148] RJ Nuismer. An energy release rate criterion for mixed mode fracture. *International journal of fracture*, 11(2):245–250, 1975.
- [149] Karuppagounder Palaniswamy. *Crack propagation under general in-plane loading*. PhD thesis, California Institute of Technology, 1972.

- [150] F. Erdogan and G. C. Sih. On the Crack Extension in Plates Under Plane Loading and Transverse Shear. *Journal of Basic Engineering*, 85(4):519–525, 12 1963.
- [151] Abbas Azhdari and Sia Nemat-Nasser. Energy-release rate and crack kinking in anisotropic brittle solids. *Journal of the Mechanics and Physics of Solids*, 44(6):929–951, 1996.
- [152] S Teichtmeister, D Kienle, F Aldakheel, and M-A Keip. Phase field modeling of fracture in anisotropic brittle solids. *International Journal of Non-Linear Mechanics*, 97:1–21, 2017.
- [153] Siddhartha Srivastava and Veera Sundararaghavan. Graph coloring approach to mesh generation in multiphase media with smooth boundaries. *AIAA Journal*, 58(1):198–205, 2020.
- [154] F. R. Schmidt, E. Toppe, and D. Cremers. Efficient planar graph cuts with applications in computer vision. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 351–356, 2009.
- [155] Atsushi Takei, Benoît Roman, José Bico, Eugenio Hamm, and Francisco Melo. Forbidden directions for the fracture of thin anisotropic sheets: an analogy with the wulff plot. *Physical review letters*, 110(14):144301, 2013.
- [156] L. B. Freund. *Energy concepts in dynamic fracture*, page 221–295. Cambridge Monographs on Mechanics. Cambridge University Press, 1990.
- [157] Egon Orowan. Fracture and strength of solids. *Reports on progress in physics*, 12(1):185, 1949.
- [158] Hong-Kai Zhao, Tony Chan, Barry Merriman, and Stanley Osher. A variational level set approach to multiphase motion. *Journal of computational physics*, 127(1):179–195, 1996.
- [159] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [160] Virginia Estellers, Dominique Zosso, Rongjie Lai, Stanley Osher, Jean-Philippe Thiran, and Xavier Bresson. Efficient algorithm for level set method preserving distance function. *IEEE Transactions on Image Processing*, 21(12):4722–4734, 2012.
- [161] Yuri Boykov, Vladimir Kolmogorov, Daniel Cremers, and Andrew Delong. An integral solution to surface evolution pdes via geo-cuts. In *European Conference on Computer Vision*, pages 409–422. Springer, 2006.
- [162] Yuri Boykov and Vladimir Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *null*, page 26. IEEE, 2003.

- [163] C. L. Epstein and Michael Gage. *The Curve Shortening Flow*, pages 15–59. Springer US, New York, NY, 1987.
- [164] Ron Kimmel. *Numerical geometry of images: Theory, algorithms, and applications*. Springer Science & Business Media, 2012.
- [165] Luis A. Santaló and Mark Kac. *Crofton’s Formulas and the Kinematic Fundamental Formula in Noneuclidean Spaces*, page 316–329. Cambridge Mathematical Library. Cambridge University Press, 2 edition, 2004.
- [166] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.
- [167] Vladimir Kolmogorov and Yuri Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, volume 1, pages 564–571. IEEE, 2005.
- [168] Paul Sweeney Jr. Crofton formulæ. 2018. [Unpublished Senior thesis], The University of Notre Dame.
- [169] Joachim Ohser and Frank Mücklich. *Statistical analysis of microstructures in materials science*. Wiley, 2000.
- [170] Gaëtan Lehmann and David Legland. Efficient n-dimensional surface estimation using crofton formula and run-length encoding. *Efficient N-Dimensional surface estimation using Crofton formula and run-length encoding*, 2012.
- [171] Ellen LS Solomon, Anirudh Raju Natarajan, Arunabha Mohan Roy, Veera Sundararaghavan, Anton Van der Ven, and Emmanuelle A Marquis. Stability and strain-driven evolution of β' precipitate in mg-y alloys. *Acta Materialia*, 166:148–157, 2019.
- [172] Atsuyuki Okabe. *Spatial tessellations*. Wiley Online Library, 1992.
- [173] David G Evans and Steven M Jones. Detecting Voronoi (area-of-influence) polygons. *Mathematical geology*, 19(6):523–537, 1987.
- [174] Franz Aurenhammer. Recognising polytopical cell complexes and constructing projection polyhedra. *Journal of Symbolic Computation*, 3(3):249–255, 1987.
- [175] Frederic Paik Schoenberg, Thomas Ferguson, and Cheng Li. Inverting dirichlet tessellations. *The Computer Journal*, 46(1):76–83, 2003.
- [176] D Trinchet-Almaguer and H Pérez-Rosés. Algorithm for Solving the Generalized Inverse Voronoi Problem. *Revista Cubana de Ciencias Informaticas*, 1(4):58–71, 2007.

- [177] Sandip Banerjee, Bhargab B Bhattacharya, Sandip Das, Arindam Karmakar, Anil Maheshwari, and Sasanka Roy. On the Construction of Generalized Voronoi Inverse of a Rectangular Tessellation. In *Transactions on Computational Science XX*, pages 22–38. Springer, 2013.
- [178] R Panozzo and H Hurlimann. A simple method for the quantitative discrimination of convex and convex-concave lines. *Microscopica Acta*, 87(2):169–176, 1983.
- [179] Renée Heilbronner and Nynke Keulen. Grain size and grain shape analysis of fault rocks. *Tectonophysics*, 427(1):199–216, 2006.
- [180] Michael A Sutton, N Li, DC Joy, Anthony P Reynolds, and Xiaodong Li. Scanning electron microscopy for quantitative small and large deformation measurements part I: SEM imaging at magnifications from 200 to 10,000. *Experimental mechanics*, 47(6):775–787, 2007.
- [181] Andrea Nicolas, Alberto W Mello, Yiwei Sun, David R Johnson, and Michael D Sangid. Reconstruction methods and analysis of subsurface uncertainty for anisotropic microstructures. *Materials Science and Engineering: A*, 760:76–87, 2019.
- [182] Abhishek Bhattacharyya, Ehab El-Danaf, Surya R Kalidindi, and Roger D Doherty. Evolution of grain-scale microstructure during large strain simple compression of polycrystalline aluminum with quasi-columnar grains: OIM measurements and numerical simulations. *International Journal of Plasticity*, 17(6):861–883, 2001.
- [183] Julie Escoda, Dominique Jeulin, François Willot, and Charles Toulemonde. Three-dimensional morphological modelling of concrete using multiscale Poisson polyhedra. *Journal of microscopy*, 258(1):31–48, 2015.
- [184] Yang Jiao and Nikhilesh Chawla. Three dimensional modeling of complex heterogeneous materials via statistical microstructural descriptors. *Integrating Materials and Manufacturing Innovation*, 3(1):3, 2014.
- [185] R Bostanabad, Wei Chen, and DW Apley. Characterization and reconstruction of 3d stochastic microstructures via supervised learning. *Journal of microscopy*, 264(3):282–297, 2016.
- [186] Sriram Ganesan, Mohammadreza Yaghoobi, Alan Githens, Zhe Chen, Samantha Daly, John Allison, and Veera Sundararaghavan. The effects of heat treatment on the response of we43 mg alloy: crystal plasticity finite element simulation and sem-dic experiment. *International Journal of Plasticity*, page 102917, 2020.
- [187] A Githens, S Ganesan, Z Chen, J Allison, V Sundararaghavan, and S Daly. Characterizing microscale deformation mechanisms and macroscopic tensile

- properties of a high strength magnesium rare-earth alloy: A combined experimental and crystal plasticity approach. *Acta Materialia*, 186:77–94, 2020.
- [188] A Zeghadi, F N’guyen, S Forest, A-F Gourgues, and O Bouaziz. Ensemble averaging stress–strain fields in polycrystalline aggregates with a constrained surface microstructure–Part 1: Anisotropic elastic behaviour. *Philosophical Magazine*, 87(8-9):1401–1424, 2007.
 - [189] Ricardo A Lebensohn, Renald Brenner, Olivier Castelnau, and Anthony D Rollett. Orientation image-based micromechanical modelling of subgrain texture evolution in polycrystalline copper. *Acta Materialia*, 56(15):3914–3926, 2008.
 - [190] Mohammadreza Yaghoobi, Sriram Ganesan, Srihari Sundar, Aaditya Lakshmanan, Shiva Rudraraju, John E Allison, and Veera Sundararaghavan. Prisms-plasticity: An open-source crystal plasticity finite element software. *Computational Materials Science*, 169:109078, 2019.
 - [191] Atsuo Suzuki and Masao Iri. Approximation of a tessellation of the plane by a Voronoi diagram. *J. Oper. Res. Soc. Japan.*, 29(1):69–97, 1986.
 - [192] Pinar Acar and Veera Sundararaghavan. Uncertainty quantification of microstructural properties due to variability in measured pole figures. *Acta Materialia*, 124:100 – 108, 2017.
 - [193] L Anand and M Kothari. A computational procedure for rate-independent crystal plasticity. *Journal of the Mechanics and Physics of Solids*, 44(4):525–558, 1996.
 - [194] Ellen LS Solomon and Emmanuelle A Marquis. Deformation behavior of β and β precipitates in mg-re alloys. *Materials Letters*, 216:67–69, 2018.
 - [195] Adam D Kammers and Samantha Daly. Digital image correlation under scanning electron microscopy: methodology and validation. *Experimental Mechanics*, 53(9):1743–1761, 2013.
 - [196] Siddhartha Srivastava, Veera Sundraraghavan, and Mohammadreza Yaghoobi. A graph-theoretic approach for multiscale modeling and prediction of crack propagation in polycrystalline materials. *Engineering Fracture Mechanics*, page 107406, 2020.
 - [197] Sangmin Lee and Veera Sundararaghavan. Calibration of nanocrystal grain boundary model based on polycrystal plasticity using molecular dynamics simulations. *International Journal for Multiscale Computational Engineering*, 8(5):509–522, 2010.
 - [198] Nicholas Fasanella and Veera Sundararaghavan. Atomistic modeling of thermo-mechanical properties of swnt/epoxy nanocomposites. *Modelling and Simulation in Materials Science and Engineering*, 23(6):065003, 2015.

- [199] Nicholas A. Fasanella and Veera Sundararaghavan. Atomistic modeling of thermal conductivity of epoxy nanotube composites. *JOM*, 68(5):1396–1410, May 2016.
- [200] A Lakshmanan, S Srivastava, A Ramazani, and V Sundararaghavan. Thermal conductivity of pillared graphene-epoxy nanocomposites using molecular dynamics. *Applied Physics Letters*, 112(15):151902, 2018.
- [201] S. Ganesan and V. Sundararaghavan. An atomistically-informed energy based theory of environmentally assisted failure. *Corrosion Reviews*, 33(6):455 – 466, 2015.
- [202] Shardul Panwar and Veera Sundararaghavan. Dislocation theory-based cohesive model for microstructurally short fatigue crack growth. *Materials Science and Engineering: A*, 708:395–404, 2017.
- [203] S Panwar, JF Adams, JE Allison, JW Jones, and V Sundararaghavan. A grain boundary interaction model for microstructurally short fatigue cracks. *International Journal of Fatigue*, 113:401–406, 2018.
- [204] D Greeley, M Yaghoobi, D Pagan, V Sundararaghavan, and J Allison. Using synchrotron radiation to improve understanding of deformation of polycrystalline metals by measuring, modelling and publishing 4d information. *IOP Conference Series: Materials Science and Engineering*, 580(1):012017, 2019.
- [205] Veera Sundararaghavan and Nicholas Zabaras. Design of microstructure-sensitive properties in elasto-viscoplastic polycrystals using multi-scale homogenization. *International Journal of Plasticity*, 22(10):1799 – 1824, 2006.
- [206] Mohsen Taheri Andani, Aaditya Lakshmanan, Mohammadreza Karamooz-Ravari, Veera Sundararaghavan, John Allison, and Amit Misra. A quantitative study of stress fields ahead of a slip band blocked by a grain boundary in unalloyed magnesium. *Scientific reports*, 10(1):1–8, 2020.
- [207] Mohsen Taheri Andani, Aaditya Lakshmanan, Veera Sundararaghavan, John Allison, and Amit Misra. Quantitative study of the effect of grain boundary parameters on the slip system level hall-petch slope for basal slip system in mg-4al. *Acta Materialia*, 200:148–161, 2020.
- [208] Pinar Acar and Veera Sundararaghavan. Do epistemic uncertainties allow for replacing microstructural experiments with reconstruction algorithms? *AIAA Journal*, pages 1–14, 2019.