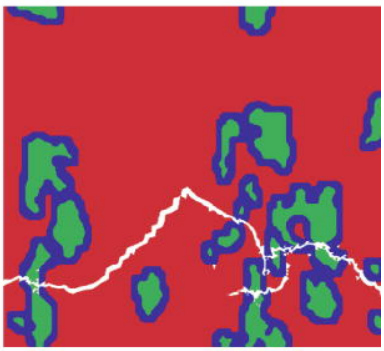
VOLUME 41, issue 2 DECEMBER 2007
http://www.elsevier.com/locate/commatsci

COMPUTATIONAL MATERIALS SCIENCE



Editors:
H. DREYSSÉ
S. SCHMAUDER

Available online at
ScienceDirect
www.sciencedirect.com
ISSN 0927-0256

Available online at www.sciencedirect.com
ScienceDirect
Computational Materials Science 41 (2007) 222–235
COMPUTATIONAL MATERIALS SCIENCE
www.elsevier.com/locate/commatsci

3D polycrystalline microstructure reconstruction from FIB generated serial sections for FE analysis

Y. Bhandari [a], S. Sarkar [a], M. Groeber [a], M.D. Uchic [b], D.M. Dimiduk [b], S. Ghosh [c,*]

[a] Computational Mechanics Research Laboratory, Department of Mechanical Engineering, The Ohio State University, OH, USA
[b] Air Force Research Laboratory, Materials and Manufacturing Directorate, AFRL/MLLMD, Wright-Patterson AFB, OH 45433-7817, USA
[c] Department of Mechanical Engineering, The Ohio State University, Columbus, OH 43202, USA

Received 14 November 2006; received in revised form 24 February 2007; accepted 4 April 2007
Available online 6 June 2007

Abstract

This paper develops a comprehensive methodology for generating realistic 3D polycrystalline microstructures followed by discretization into a 3D tetrahedral mesh for finite element (FE) analysis. With input data on crystallographic orientations for a series of grain sections, created by a dual beam focused ion beam-scanning electron microscope (DB-FIB) system, the reconstruction method uses primitives in CAD modeling based on hierarchical geometrical representation. It involves steps of data cleanup, interface point identification, parametric polynomial and NURBS function based surface patch reconstruction, generalized cell-decomposition, geometric defeaturing and gap-overlap removal. The implementation of the entire procedure is done with the user-programming facilities of a commercial CAD package Unigraphics NX3. The reconstruction algorithms are validated with various error criteria. Subsequently, a finite mesh generator is developed to consistently discretize the reconstructed polycrystalline domain into a finite element mesh with resolution control that is necessary for meaningful computational analysis in microstructure–property evaluation. The mesh generator is enriched with mesh quality improvement and degree of freedom reduction tools.

Keywords: FIB-SEM; 3D microstructure reconstruction; Parametric surface; NURBS; Mesh generation

1. Introduction

Polycrystalline microstructures of metals and alloys are generally quite heterogeneous, consisting of aggregates of grains or multiple phases and constituents. Their morphological characteristics, e.g. size, shape, and spatial distribution, and crystallographic characteristics, e.g. orientation and misorientation distributions often govern their mechanical behavior and fatigue failure response. Significant work has been performed to understand the deformation and damage mechanisms responsible for failure and fatigue phenomena of materials [1–8]. While crystal-plasticity based computational models of polycrystalline materials are making great strides in predicting stress–strain behavior with reasonable accuracy, ductility and fatigue failure predictions are still far from mature. Morphological and crystallographic heterogeneities in the microstructures result in strong anisotropy and localized non-homogeneous deformation, which pose severe challenges to these computational models. For robust predictive capabilities, it is necessary for these models to represent morphological and crystallographic features of the microstructure with high fidelity.

A recently developed approach in [23] to characterize 3D microstructures uses a dual beam focused ion beam-scanning electron microscope (DB-FIB-SEM) system to acquire crystallographic orientation information from electron backscatter diffraction (EBSD) maps of a series of sequentially stacked cross sections. This advancement enables a high-fidelity reconstruction of 3D microstructure,

* Corresponding author. Address: Scott Lab, The Ohio State University, Columbus, OH 43202, USA. Tel.: +1 614 292 2599; fax: +1 614 292 3163.
E-mail address: ghosh.5@osu.edu (S. Ghosh).

0927-0256/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.commatsci.2007.04.007

which can be discretized and analyzed by computational methods, such as the finite element (FE) analysis for material property prediction. The present paper is aimed at developing a CAD-based methodology for 3D grain reconstruction followed by mesh generation for FE analysis, which is characterized by the following key features:

- *Input*: Crystallographic orientation data of sections on a 3D rectangular grid is used to segment individual grains in the microstructure;
- *Output*: The output is a graded FE mesh of the high-fidelity microstructural model;
- *Process automation*: The process is fully automated with minimal additional user input;
- *Generality*: The method is flexible and applicable to different material microstructures;
- *Robustness:* The microstructure reconstruction method is robust, reducing errors or noise in the experimentally acquired data;
- *Optimal FE model*: The FE mesh and model are optimized with respect to the number of nodes and elements while retaining accuracy.

The novelty of this method is in its ability to balance accuracy and efficiency by controlling the resolution of the simulated microstructure. A commercial CAD package, Unigraphics NX3 [28], is used as geometric kernel to perform operations required for the polycrystalline microstructure reconstruction. A special module is developed through an Open C API interface in NX3, which allows direct user access to most of its geometric modeling and manipulation facilities. Section 2 of this paper reviews related work in this general topical area. Section 3 discusses data collection, cleanup and the reconstruction process. Validation of this method is discussed in Section 4 and finally a Delaunay triangulation based mesh generation is discussed in Section 5.

2. Brief review of microstructure reconstruction methods

Various methods of 3D microstructural information acquisition have been reported in the literature. Nondestructive measurement techniques based on ultrasonics or its variants, such as acoustic microscopy or laser ultrasonics [9,10] rely on acoustic wave reflection properties and have limited application for grain imaging in metals. While X-ray based computed tomography [11,12] methods are widely used in 3D solid model generation, they are generally deficient in achieving the desired resolution for detailed study of many materials. Synchrotron based CT technology, which is still not commercially available in general and is quite expensive, has been developed to yield tomographic images with considerable resolution improvement [13]. Recent advances in destructive metallographic methods show considerable promise in microstructure generation. Models derived from statistical interpolations of 2D images [14] and the morphologically 'precise' models derived from 3D reconstruction of serial-section data [15] are gaining considerable attention. While the former approach [14] has the advantage of requiring relatively less experimental effort, its reliability in reproducing important microstructural characteristics depends on the accuracy of the statistical interpolators. It can cause large errors in the extreme values of distributions of key microstructural parameters.

Three-dimensional microstructural information is conventionally represented in two ways, viz. (i) as a collection of voxels (*volumetric pixels)*, each of which may be an array containing selected data, or (ii) as ensembles of grains, for which the grain features (e.g. surfaces, edges) are constructed in terms of parametric functions. Lewis and Geltmacher [16] have developed the voxel-based approach, wherein a typical microstructure containing 138 grains of a stainless steel has been discretized into 3.5 million voxel-based elements. To avoid prohibitive computational costs, they combined four voxels in each direction to construct their FE model, which can result in loss of important geometric information. Further, computational artifacts (e.g. unrealistic local stresses) will occur if the voxel-based sampling becomes too spatially coarse. With respect to the second approach, various methods of parametric reconstruction of individual ensembles have been proposed in the literature. The marching cube method [17] constructs triangular models of constant density in 3D by linear interpolation of density values. Methods developed in [18,19] use Voronoi diagrams for reconstruction of surfaces from random seed points on a smooth surface. Other techniques have been developed in medical imaging [20–22] using serial-sectioned images to construct 3D images by Delaunay triangulation.

A parametric representation of grain vertices, edges, and surfaces is proposed in this paper for representing 3D volumetric domains of polycrystals. The method can have significant advantages over the voxel-based reconstruction methods through control of the desired resolution. Each grain is modeled as a solid bounded by arbitrarily shaped surface patches that can capture the smoothness of actual grain boundary interfaces. Any grain size can also be meshed in this representation.

3. Reconstructing the 3D polycrystalline microstructures

Fig. 1 is a flowchart of the operations necessary to construct the 3D microstructure from the FIB-SEM serial-sectioned data. The steps are discussed next.

3.1. DB FIB-SEM data acquisition and pre-processing

3.1.1. Data acquisition

Data acquisition using the DB FIB-SEM is described in detail in [23]. Microstructural sections are successively sampled with the ion beam and crystallographic orientation maps are acquired for each section using EBSD. The orientation maps contain pixels with specific slip system orienta-

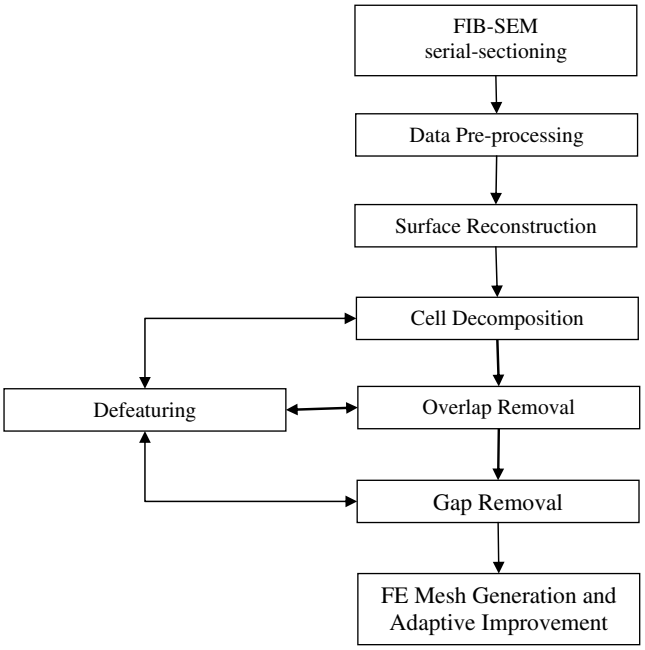


Fig. 1. Flow chart showing the steps for 3D polycrystalline microstructure reconstruction from FIB-SEM generated serial-sectioning data, with subsequent mesh generation.

tion, phase, and reliability index values. For 3D volumetric rendering, each 2D pixel is extruded along the sectioning direction by the section thickness.

3.1.2. Grain segmentation

Crystallographic orientation data, defined by sets of Euler angles $(\varphi_1, \Phi, \varphi_2)$ for each voxel, is used to segment grains of the polycrystalline microstructure. Euler angles correspond to angular rotations that transform the global coordinate system to the local orientation of the crystalline lattice [24,25]. The misorientation between two neighboring voxel points A and B is calculated from their respective orientations using the relation:

$$\theta = \min \left| \cos^{-1} \left\{ \frac{\mathrm{tr}(\mathbf{g}_A \mathbf{g}_B^{-1} \mathbf{O} - 1)}{2} \right\} \right| \tag{1}$$

where g_A and g_B are the orientation matrices of voxels A and B respectively, expressed as

$$\mathbf{g}_i = \begin{pmatrix} \cos\varphi_1 \cos\varphi_2 - \sin\varphi_1 \sin\varphi_2 \cos\Phi & \sin\varphi_1 \cos\varphi_2 + \cos\varphi_1 \sin\varphi_2 \cos\Phi & \sin\varphi_2 \sin\Phi \\ -\cos\varphi_1 \sin\varphi_2 - \sin\varphi_1 \cos\varphi_2 \cos\Phi & -\sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 \cos\Phi & \cos\varphi_2 \sin\Phi \\ \sin\varphi_1 \sin\Phi & -\cos\varphi_1 \sin\Phi & \cos\Phi \end{pmatrix}_i, \quad (i = A, B) \tag{2}$$

Symmetry of the crystal lattice is taken into account through a crystal symmetry operator $\mathbf{O}$. For the cubic system in this work, misorientation is described by 24 equivalent rotations. A tolerance ($\mathrm{mo}_{\mathrm{tol}} \leqslant 4°$) in the minimum misorientation angle between A and B is used to delineate two points belonging to adjacent grains. Steps in the segmentation algorithm are:

1. Mark all voxels as unassigned, i.e. assign their grain number $N = 0$. Initiate the grain currently under consideration as $N_{\mathrm{current}} = 1$.
2. Select an unassigned point P, that has the highest reliability index in the OIM scans, and assign $r_{\mathrm{P}} = r_{\max}$. The reliability index is a scalar value assigned during experimental data collection, quantifies the data quality. The reliability parameter r_{P} is used to reduce noise during grain identification.
3. Choose a suitable tolerance in the reliability index value r_{to}. For a point P, if $r_{\max} < r_{\mathrm{tol}}$ then exit grain segmentation routine, otherwise assign N_{current} to P. If there is no unassigned data point with a reliability index value above r_{tol} then proceed to data cleanup. Otherwise, continue with grain identification.
4. For each unassigned neighbor of P, compute its misorientation with P. If the misorientation is less than $\mathrm{mo}_{\mathrm{tol}}$, assign the grain number N_{current} and add the neighbor to a list of points in N_{current}, otherwise leave unassigned.
5. Continue with step 4 recursively for each point in the list for N_{current} until no new points within $\mathrm{mo}_{\mathrm{tol}}$ can be found.
6. Set $N_{\mathrm{current}} = N_{\mathrm{current}} + 1$ and go to step 1 until all reliable points have been assigned. Some points will be left unsegmented or unassigned after this procedure following the reliability condition.

There is an additional complication with the nickel-base superalloys used as example in this work. Some grains contain relatively narrow crystallographically 'twinned' regions. Due to poor spatial resolution of the OIM scans, their thickness is not usually well-sampled. This makes their reconstruction difficult. Consequently, the twin-related grains are merged with their parent grains in this representation. This issue is irrelevant with adequate data resolution.

3.1.3. Section data cleanup

The experimental procedure of the previous section in segmenting data into regions of contiguous points with similar orientations, leave some of the points un-indexed or with unreliable orientation information. This is commonly observed for points close to or on grain boundaries, pores, and inclusions. The data cleanup procedure heuristically assigns "noise points" to a grain. Different procedures

are implemented depending on the type of noise, classified as follows:

- *Unassigned indexed points*: Points that have reliable orientation information but are not assigned to any grain are assigned to a neighboring grain with the least misorientation.
- *Un-indexed points*: Points that do not have reliable orientation information are assigned to a grain with which they share the most surface area.
- *Contained grains*: Grains having sizes less then a specified tolerance (64 points in this case corresponding to a 1 μm equivalent square grain in the analysis plane) or grains that are contained completely inside any other grain are usually inclusions. These are merged with the parent grain.

Despite the image recognition technology used during data collection, some slice-alignment errors are still present in the resulting data as shown in Fig. 2a. Vertical protrusions continue along the entire length of some sections. Consequently, a heuristic algorithm based on the in-plane translation of sections is developed to minimize misalignment. Each voxel in a given section is compared to the corresponding voxel in the previous section and the number of voxels for calculating the total number of voxels belonging to different grains. The section is translated in-plane to a position for which the number of voxels in different grains is a minimum. Fig. 2b shows a section plot with the result of this algorithm on real data. While some protrusions still exist at some grain boundaries, they are not continuous along the entire vertical length. The protrusions that remain are due to carbide particles at grain boundaries and erroneous data points, not misalignment. Additionally, it may appear that new grains are created in Fig. 2b, but this is only an artifact of the 2D visualization. The sections can be translated in the x (vertical) or y (into the paper) direction. As a result a grain that may not have intersected the section shown in Fig. 2a, may now intersect the same section after the alignment. The grain is not a new grain, but rather an existing grain in a shifted position. The data is cropped after alignment to account for the missing data at slice edges, which results from the physical offset. One shortcoming of this algorithm is that the minimum distance by which each section can be moved is limited to the voxel dimension ΔL. Alignment errors below this limit are not corrected.

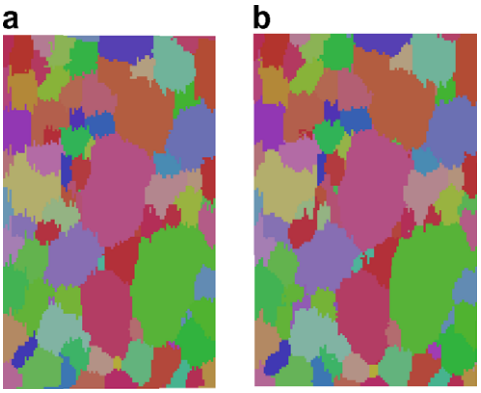


Fig. 2. A pixel-based representation of the microstructure viewed normal to the sectioning direction: (a) before alignment showing continuous protrusions on many grain boundaries in the vertical direction corresponding to misaligned sections; and (b) after alignment showing that protruding features no longer continue fully along the vertical direction.

The data acquisition and pre-processing algorithm is tested on a specimen of a fine-grained polycrystalline nickel-base superalloy, IN100. After alignment, the dimensions of the data cube obtained are 6.25 μm × 41.25 μm × 25.0 μm in the x, y and z directions, respectively, with a voxel size of $\Delta L = 0.25$ μm in each direction, corresponding to a total of 412,500 voxels in the volume. After segmentation and cleanup a total of 196 grains are identified for these data. Alignment is limited to moving the slices in integral multiples of the voxel size ΔL. Since, these processes involve an error of the order of ΔL, this paper will frequently use ΔL as one of the reference length scales for reporting errors.

3.2. Construction of individual grain domain

At the start of the reconstruction process, each grain is represented as a set of contiguous voxels with near-identical orientation values. Metallographic observations reveal that sharp curvatures are often present at the intersection of three or more grains, but the interface shared by two grains is generally locally smooth. In compliance with these observations, smooth interface patches are generated by the interpolation of voxels at the interface of each grain pair. Two types of interpolation functions are used, viz. (a) parametric polynomial and (b) non-uniform rational B-spline (NURBS) [26,27]. Following surface reconstruction, the volumetric domain of each grain is constructed by a cell-decomposition process. The steps involved in this algorithm are discussed next.

3.2.1. Surface reconstruction using parametric representation

Fig. 3 depicts surface points ($\bar{s}$) that are identified at the interface between a pair of grains. The interface may be

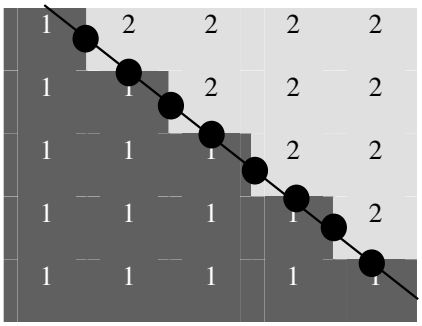


Fig. 3. Interface points identified as black dots between two adjacent grains 1 (dark grey voxels) and 2 (light grey voxels) and, an interpolated surface (in this case a straight line) that is fit to these points.

represented by fitting various parametric surfaces to these points, including ruled surfaces, parametric polynomial surfaces, and NURBS surfaces. The requirement on these surfaces is that they are smooth, accurate, and extendable. In this work, parametric polynomial and NURBS surfaces are fit to each set of surface points due to their amenability in the CAD/FEA environment. Surfaces with the best fit are selected.

3.2.1.1. Parametric polynomial surface-fitting. A parametric polynomial surface patch of order n is defined as a polynomial expression of the coordinates (x, y, z) of a point on the surface in terms of parametric coordinates (u, w). i.e.,

$$\begin{aligned} x &= x(u,w) = x_{00} + x_{01}u + x_{10}w + x_{11}uw + \cdots + x_{nn}u^n w^n \\ y &= y(u,w) = y_{00} + y_{01}u + y_{10}w + y_{11}uw + \cdots + y_{nn}u^n w^n \\ z &= z(u,w) = z_{00} + z_{01}u + z_{10}w + z_{11}uw + \cdots + z_{nn}u^n w^n \\ &\forall u, w \in [0,1] \end{aligned} \tag{3}$$

or in a more compact form,

$$\overline{Q}(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{n} \overline{Q}_{ij} u^i w^j \tag{4}$$

where $\overline{Q} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$ and $\overline{Q}_{ij} = \begin{Bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{Bmatrix}$. An adaptive algorithm is implemented for surface fitting using steps outlined next.

1. Use the least-square method to fit a plane P, through the given set of data-points $\overline{\mathbf{S}} = \{\overline{\mathbf{s_1}}, \overline{\mathbf{s_2}}, \ldots, \overline{\mathbf{s_N}}\}$.
2. Project the points in $\overline{\mathbf{S}}$ onto P to obtain an initial guess of parameters $\mathbf{U} = \{(u_1, w_1), (u_2, w_2), \ldots, (u_N, w_N)\}$ in Eq. (4). Since a complete polynomial is used, the choice of the orthogonal directions will only affect the coefficients of $\overline{Q}_{ij}$ and not the surface itself.
3. Set $n = 1$. Use the parameters $\mathbf{U}$ and data-points $\overline{\mathbf{S}}$, to obtain the polynomial coefficients $\overline{Q}_{ij}$ in Eq. (4) using the least-square method.
4. Calculate the average error E_{initial} for all points. The absolute normal error ε_d is the smallest distance to a point along the surface normal, while E is its average, i.e.

$$\varepsilon_d = \left\{ \min(|\lambda|) : \bar{D}_d = \overline{Q}(u,w) + \lambda \frac{\overline{N}(u,w)}{|\overline{N}(u,w)|} \right\} \tag{5a}$$

$$E = \frac{\sum_{d=1}^{N} \varepsilon_d}{N} \tag{5b}$$

Here, $\overline{N}(u,w)$ is the normal to the surface at the parametric coordinates (u,w) and is the cross product of two tangents at that point on the polynomial surface as:

$$\overline{N}(u,w) = \overline{T}_u(u,w) \times \overline{T}_w(u,w) \tag{6}$$

where $\overline{T}_u(u,w) = \sum_{i=1}^{n}\sum_{j=0}^{n} i\overline{Q}_{ij} u^{i-1} w^j$ and $\overline{T}_w(u,w) = \sum_{i=0}^{n}\sum_{j=1}^{n} j\overline{Q}_{ij} u^i w^{j-1}$.

5. Solve Eq. (4) using the Newton–Raphson iterative method for better estimates of $\mathbf{U}$.
6. Calculate average error E_{final} for each fit. If $|E_{\text{initial}} - E_{\text{final}}| \leqslant$ tolerance then proceed; Otherwise go to step 3.
7. If $E_{\text{final}} >$ tolerance then $n = n + 1$ and go to step 3; Otherwise exit.

The program starts with a first order polynomial and adaptively increases the order until convergence of the average normal error in Eq. (5b). Fig. 4 depicts the effectiveness of this algorithm. A third-order polynomial surface is adaptively fit to a set of points and the normal distance is calculated by projecting each point onto the surface. In Fig. 4b each bar corresponds to the distance of a point from the parametric surface. The height of the each bar is the absolute normal error in comparison with the step-size ΔL of the serial-sectioning experiment, as shown. The convergence of the normalized averaged normal error $E^{\text{normalized}} = \frac{\sum_{d=1}^{N} \frac{\varepsilon_d}{\Delta L}}{N}$ for different order polynomials is shown in Fig. 4c.

3.2.1.2. Surface-fitting with non-uniform rational B-spline or NURBS functions. Non-uniform rational B-spline or NURBS [29] are the second type of parametric surface that are fit to the grain interface points. The NURBS expression for a point coordinates (x, y, z) are written in terms of parametric coordinates (u, w) of the surface as

$$\overline{Q}(u,w) = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} \overline{B_{i,j}^{h}} N_{i,k}(u) M_{j,l}(w) \tag{7}$$

where the coefficients $\overline{B_{i,j}^{h}}$ corresponding to vertices of a four-dimensional polytope, (k, l) are the order of the surface spline functions and $(n+1)$, $(m+1)$ correspond to the number of points in each parametric direction. $N_{i,k}$, $M_{j,l}$ are NURBS basis functions defined by the Cox–de-Boor recursion formulae as

$$\begin{aligned} N_{i,1}(u) &= \begin{Bmatrix} 1 & \text{if } x_j \leqslant u \leqslant x_{j+1} \\ 0 & \text{otherwise} \end{Bmatrix}, \\ N_{i,k}(u) &= \frac{(u - x_i) N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \end{aligned} \tag{8}$$

where $[X] = [x_1, x_2, \ldots, x_{n+k+1}]$ is called the knot-vector with $x_i \leqslant x_{i+1}$, and $u \in [x_1, x_{n+k+1}]$. The same iterative algorithm described in Section 3.2.1.1 is implemented within the NX3 based routine [28] to fit the NURBS surface. The polynomial or NURBS-based surface with the lowest error $E^{\text{normalized}}$ and with no self-intersection is retained. Fig. 5 shows a histogram of the percentage of surfaces with a given value of $E^{\text{normalized}}$. Approximately 97% of the surfaces generated for the entire polycrystalline domain have low normalized errors. These error can be further reduced if higher-order functions (less than third order) are considered for the polynomial and NURBS surfaces. However, higher-order surfaces show sharp local variations and tend

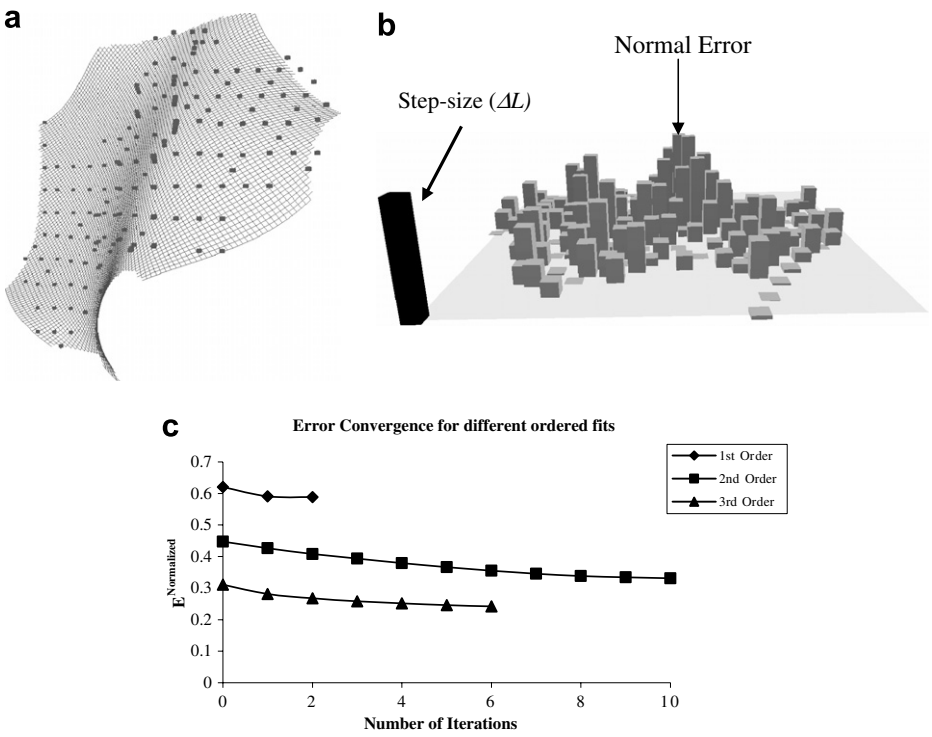


Fig. 4. (a) A representative third-order polynomial surface with a least-squares fit to a set of points delineating a grain boundary (only points above the surface can be seen), (b) point-wise 3D histogram of the point-wise error ε_d, and (c) convergence of average normalized error ($E^{\text{normalized}}$) with different order polynomial interpolation functions.

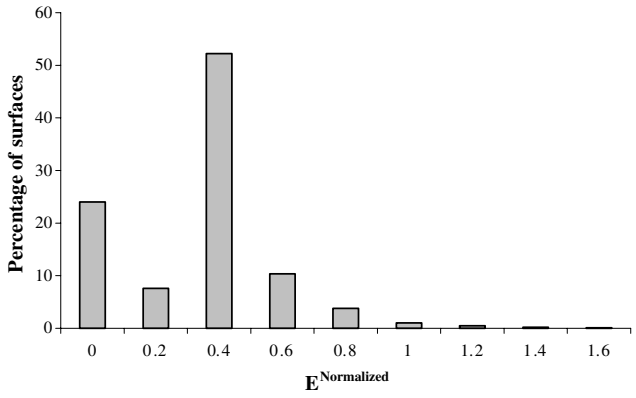


Fig. 5. Percentage of surfaces having the designated average error $E^{\text{normalized}}$.

to self-intersect if used unconditionally. A few larger error points may also be due to misindexed points near the boundary, an experimental attribute. A better resolution data is likely to improve the surface fitting results.

3.2.2. Grain definition via cell decomposition using fit surfaces

This step involves creation of individual grain domains, upon creation of the surfaces delineating grain boundaries. Spatial partitioning algorithms, such as quad-tree, octree, or BSP trees have been extensively used in geometric modeling for solid body representation. These methods partition a large region or 'universe' into a number of cells. Each cell that is completely contained in a grain domain is assigned to the grain. The desired grain is represented by the aggregate of cells. Methods differ in the choice of the universe cell and the choice of partitioning surfaces. One of the most general methods is the cell-decomposition method, where an arbitrarily shaped solid is defined as the "universe cell" and arbitrarily shaped surfaces are considered as the partitioning surfaces. Fig. 6 shows how this technique reconstructs individual grains from grain surfaces obtained by the surface fitting algorithm using the following steps:

1. *Construct the 'Universe Cell' (W)* to wrap the set of voxels of a grain G, having an offset of at-least twice the step-size, i.e. $2 \times \Delta L$. In NX3, the wrap of a set of points is defined as the offset solid of the convex planar solid containing all the points (see Fig. 7). It is ensured that W remains within the bounds of the experimental domain.
2. *Select the partitioning surface (S)* with the largest number of surface points from the set of unused surface functions for the grain G. Once extrapolated to the boundaries of W, S will partition W into cells, CP, and these cells are assigned as either G or GN cells (Fig. 6b) depending on their location.
3. *Perform cell partitioning* by selecting the unassigned cells, $\mathbf{C}$ ($\mathbf{C} = \{W\}$ for the initial cell), which contain at-least one of the points interpolated by S. Extrapolate S to partition $\mathbf{C}$ into sub-cells $\mathbf{CP}$ (Fig. 6c–e).
4. *Assign cells* for each $\text{CP}_i \in \mathbf{CP}$ by performing a containment check inside CP_i for all points belonging to G and

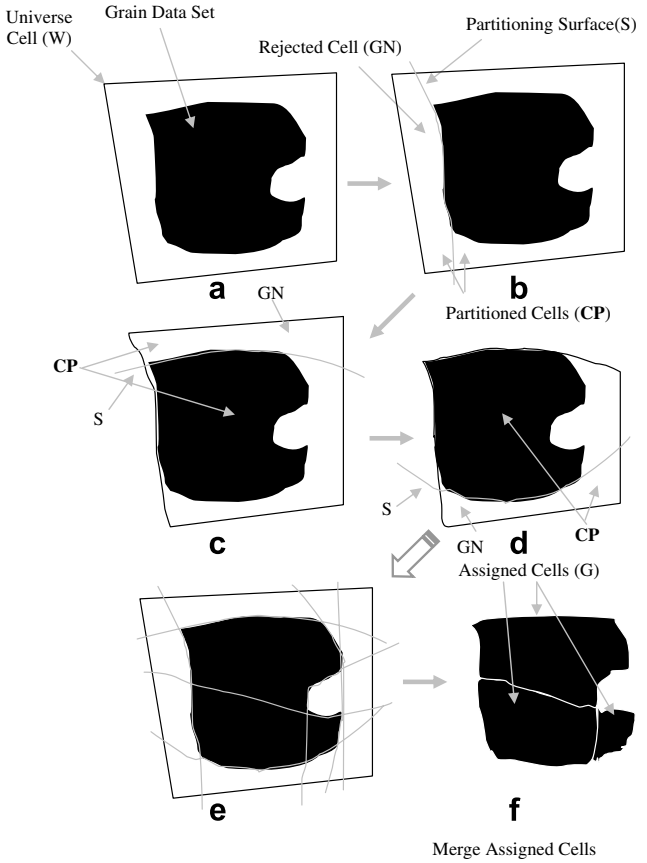


Fig. 6. A 2D schematic representation of the succession of operations in the cell-decomposition methodology. The reconstructed solid grain is the union of the cells that have been assigned to that grain after partitioning of the universe cell.

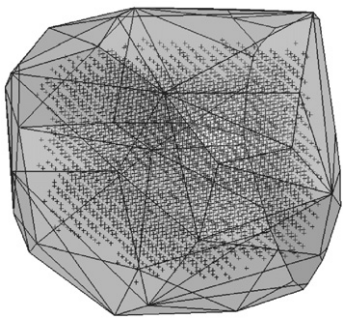


Fig. 7. Universe cell W (semi-transparent) overlaid on the grain volumetric data points.

all its neighbors *GN*. If more than 90% of the contained points in CP_i belong to *G* then assign to *G*. If less than 10% belong to *G*, assign it *GN*. Otherwise leave it unassigned. Repeat steps 2–4 till all surfaces have been considered. Unassigned cells containing more than 10% but less than 90% of their surface voxels in *G* are assigned to *G* or *GN* based on maximum containment of all voxels, not just surface voxels, of *G* or *GN* inside it.

5. *Merge the cells* assigned as *G* for the final grain representation. Delete all other cells (Fig. 6f).

Selection of the "universe cell" *W* in step 1 is critical. If the cell is too large, then difficulties in extrapolation of surfaces will arise during partitioning. If it is too small, then the surfaces may not have room for partitioning. Fig. 7 shows the wrap solid as the universe cell for one such grain. In step 3, the selection of those cells that contain at least one point used in the surface interpolation leads to a more local reconstruction. In this way, a surface fit to one set of points in one region of solid may not influence other regions. Using the largest surfaces described in step 2 in conjunction with this local approach, retains the local nature of the reconstruction. Improvements in both accuracy and also efficiency are achieved as a result. Fig. 8a shows the reconstructed grain for the points in Fig. 7a. The surface points of the grain in the voxelized representation are shown to demonstrate the relative location of the reconstructed solid. The corresponding error plot is shown in Fig. 12a.

3.3. Grain surface compatibility in a polycrystalline aggregate

Overlaps and gaps are expected between individual grains constructed by the solid reconstruction process in Section 3.2, since individual grain surfaces are created in isolation from the other grains. Regions occupied by more

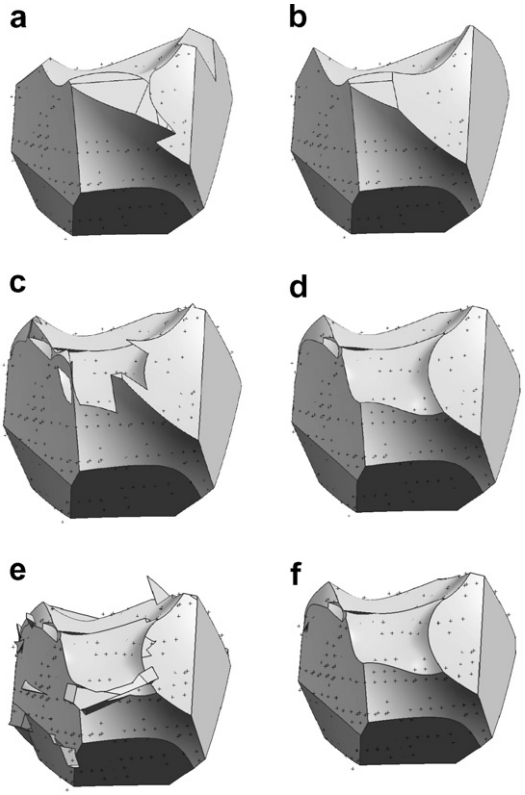


Fig. 8. A single grain reconstruction showing discrete surface data points above the surface: (a) after cell decomposition, (b) after cell decomposition with defeaturing, (c) after overlap removal, (d) after overlap removal with defeaturing, (e) after gap removal, and (f) after gap removal with defeaturing. Note that defeaturing is done several times according to the procedures described in Section 3.4.

than one grain (overlaps) and regions where no grains are present (gaps) are non-physical features that should be removed. Additionally, geometric artifacts that result from the process are removed through a defeaturing process. These representation refinements are described as follows. Note that the defeaturing process described in Section 3.4 (Fig. 8) is applied before, during, and after the overlap and gap filling processes.

3.3.1. Overlap removal

A schematic of the grain-overlap removal algorithm is depicted in Fig. 9. Let $\mathbf{R} = [R_1, R_2, \ldots, R_M]$ be the set of all reconstructed solid grains and $\overline{\mathbf{V}} = \{\overline{\mathbf{V_1}}, \overline{\mathbf{V_2}}, \ldots, \overline{\mathbf{V_M}}\}$ be the set of the voxel points belonging to these grains. The following convention is used to represent set operators. The union and intersection of two sets are represented by the operators $\cup$ and $\cap$ respectively, while the union and intersection of two solids are represented by the Boolean nomenclature: Union(S_1, S_2) and Intersect(S_1, S_2) respectively. The overlap removal algorithm follows the steps outlined below:

1. Set $i = 1$ and $j = 2$.
2. Find $\mathbf{I} = \text{Intersect}(R_i, R_j)$ as the set of all common volumes shared by R_i and R_j. If $\mathbf{I} = \{\varnothing\}$ then go to step 8, otherwise proceed to step 3.
3. For $I_k \in \mathbf{I}$ and for all surface points $\overline{\mathbf{D}^{ij}} = \{D_1^{ij}, D_2^{ij}, \ldots, D_{N_{ij}}^{ij}\}$ belonging to the interface between the grains i and j, perform a containment test to find the points $\mathbf{P}^k = \{P_1^k, P_2^k, \ldots, P_A^k\}$ that are contained in I_k.
4. Fit a surface S^k to the set of points $\overline{\mathbf{P}^k} = \{\overline{P_1^k}, \overline{P_2^k}, \ldots, \overline{P_A^k}\}$.
5. Partition I_k with the surface S^k into the solids $\mathbf{IP} = \{\text{IP}_1^k, \text{IP}_2^k, \ldots, \text{IP}_B^k\}$.
6. For each $\text{IP}_l^k \in \mathbf{IP}$, count the number of points of $\overline{\mathbf{V_i}}$ and $\overline{\mathbf{V_j}}$ that lie inside IP_l^k as n_i and n_j respectively. If $n_i > n_j$ then assign IP_l^k to R_i by applying the operator $R_i = \text{Union}(R_i, \text{IP}_l^k)$, otherwise perform $R_j = \text{Union}(R_j, \text{IP}_l^k)$.
7. Repeat steps 3–6 for all $I_k \in \mathbf{I}$.
8. If $j = M$ then go to step 9, otherwise, perform $j = j + 1$ and go to step 2.
9. If $i = M$ then exit, otherwise, set $i = i + 1$, $j = i + 1$ and go to step 2.

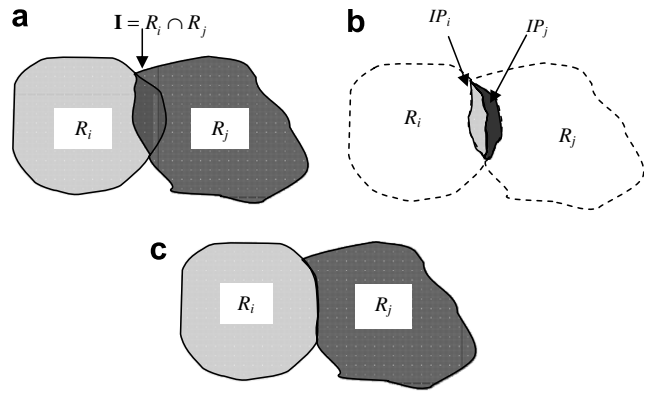


Fig. 9. 2D schematic representation of the grain-overlap removal process: (a) two overlapping grains and identification of overlap region **I** between R_i and R_j, (b) partitioning of **I** into IP_i and IP_j using a fitted surface and (c) assignment of IP_i and IP_j to R_i and R_j, respectively and, creation of the resulting non-intersecting grains.

By fitting a separate surface in step 4, only those points that are local to the overlapping region are included in the partitioning process. The accuracy of the interface is better, since the fitting of surface S^k is not influenced by interface points from other regions. In overlapping regions that are small enough to not contain any surface points, no partitioning is done. Fig. 8c shows the effect on the geometry of a representative grain after overlap removal.

3.3.2. Gap removal

As shown in Fig. 10, gap identification relies on successive subtraction of the reconstructed grains from the entire ensemble space. Some of the gaps produced by this process may be too large for a-priori assignment to an individual grain. An additional step of subdividing larger gaps through the wrap surfaces of individual grains is used to account for such cases. The wrap surface is defined as the surface bounding a wrap solid as defined earlier. Choosing wrap surfaces in this step ensures the condition that when these gaps are assigned to appropriate grains, no grain exceeds its initial wrap-boundary. After partitioning, many of the gaps produced are so small they may not contain a single voxel. Consequently, the choice of grain to assign the gap is not straight forward. Rather than assign the gap arbitrarily, the neighbor which shares the maximum common surface area with the gap is chosen. The solids generated are relatively smooth and are easier to mesh. Fig. 8e shows the effect of gap removal. While handling the exceptions and errors during subtraction, care must be taken to produce no new intersections. Otherwise, the overlap removal procedure may have to be repeated.

3.4. Defeaturing spurious artifacts of the reconstructed grains

The processes of grain reconstruction and compatibility enforcement can lead to certain non-physical artifacts in the microstructure. For example, a small face such as a sliver on the grain boundary can cause problems with FE mesh generation, as well as give rise to spurious high local stresses in the analysis phase. Defeaturing is the process of removing these spurious artifacts prior to generating a FE mesh and model. Defeaturing routines are introduced at three places in the process: (i) before overlap removal, (ii) before gap removal, and (iii) before mesh generation.

The defeaturing procedure operates under differing constraints depending upon which of the three steps are being carried out. For example, defeating can allow new gaps or overlaps to form prior to the overlap removal stage. However, defeaturing cannot introduce gaps or overlaps after the gap removal stage. A single procedure for removing the artifacts is implemented in NX3, but with flexibility

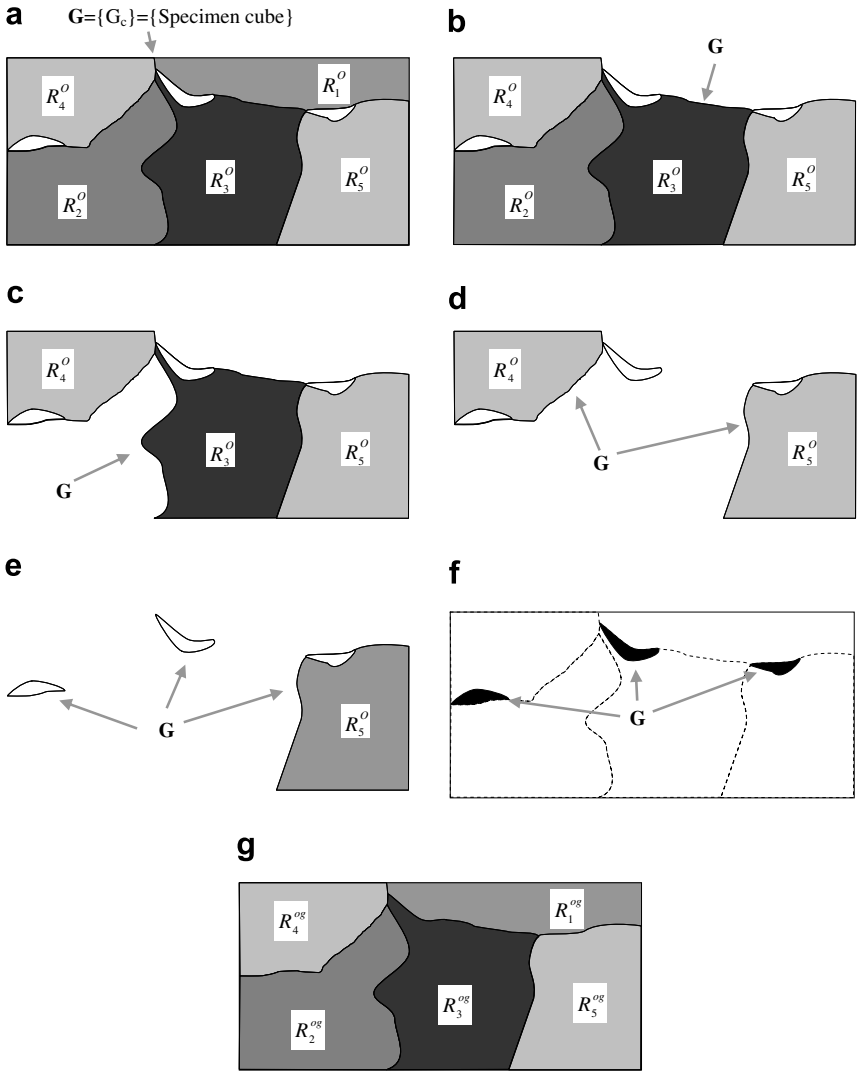


Fig. 10. Steps involved in identification and removal of gaps: (a) two-dimensional reconstructed microstructure with overlaps removed, (b)–(e) process of subtraction of grains R_i^o from **G**, (f) gaps identified in the previous iterations, and (g) microstructure after assigning gaps to appropriate grains. For the purpose of clarity, the step illustrating the partitioning of gaps is skipped.

to adapt to specific compatibility requirements depending on the stage of its application.

The first step in the defeaturing algorithm is identification of artifacts. Although various types of non-physical features are possible, the most common occurrence is 'tiny-faces' that may be defined in NX3 in two ways as (see [28]) (*i*) faces for which the surface area is less than a specified tolerance, and (ii) faces for which the surface-area to perimeter ratio is less than a specified tolerance. The tolerance for grains of different sizes is obtained as a linear variation, ranging from 0 for a grain with zero volume, to $2 \times \Delta L$ for a moderate-sized grain. The tolerance may be adjusted for different microstructures or resolutions.

After identification of the tiny faces, defeaturing is applied in two phases. First, clusters of neighboring tiny faces are removed simultaneously using the '*Simplify Body*' feature in NX3. If this operation fails, automatic removal of the faces corresponding to the failing wound has been found to be useful. In the second step, the remaining individual faces in the simplified geometry are removed by using the NX3 modeling feature '*Replace Face*'. The face to be removed and the neighboring face are specified in this operation. Unlike the '*Simplify Body*' function, specifying different neighboring faces may yield different defeatured solids. The program exploits the generation of multiple defeatured geometries for the same grain by performing all possible '*Replace Face*' operations and choosing the solid with the least number of faces as the correct defeatured solid.

For preventing the introduction of new overlaps, the original solid is first subtracted from the defeatured solid. This difference represents additional solids in the defeatured solid. These additional solids are then subtracted from the neighboring grains. To prevent the insertion of new gaps, the defeatured solid is subtracted from the original solid. The new additional solids regions created are then assigned to the neighboring grains based on the gap removal scheme described earlier. The final step in defea-

turing is to quantify the effect of all the operations on the overall quality. Defeaturing a grain may give it a very simple representation, but its neighbors may become worse after the compatibility operations. The total number of faces of a grain and its neighbors are compared before and after the defeaturing process. If the number is reduced then the defeaturing operation is assumed to be successful, otherwise the older state is restored. Fig. 8b–f shows a defeatured grain after reconstruction, overlap removal, and gap removal respectively.

4. Results and validation of the reconstruction algorithm

The CAD-based reconstruction methodology is applied to the microstructure of a fine-grained polycrystalline nickel-base superalloy, IN100. Fig. 11a shows a surface of the experimental volume and Fig. 11b shows the 3D reconstructed grain microstructure looking at the same face as in Fig. 11a with good agreement. The stepped grain boundaries in the voxel representation are smoothed in the reconstructed representation using low order surfaces. It should be noted that the "low order" surfaces used in this fitting result in some local deviations from the voxel data. When fifth and higher order polynomial and NURBS surfaces are fit to the data, the surfaces match the local undulations quite closely. However, these surfaces are frequently self-intersected or cannot be extended to partition the universe cell. On the other hand, third order and lower order surfaces retain most shape features with low normal errors. It is not possible to comprehend the accuracy of a surface fit by visualizing a 2D section. Also, the surface error in the projected line for a grain section may not correspond to the shortest distance between the voxel-based surface points and the fit surface. The polynomial and NURBS surfaces may change shape significantly just above or below the section being viewed. A representative single grain is shown in Fig. 11c (a) in its voxelized representation, and (b) with smoothened grain boundaries. The overall accuracy of the 3D grain representation is qualitatively obvious from this figure. Some of the local deviations seen in Fig. 11b and c will be rectified during FE mesh generation phase discussed in Section 5. The accuracy of a grain representation, in comparison with the voxelzed representation, is quantitatively evaluated in terms of a surface projection error

$$\varepsilon_i^g = \begin{Bmatrix} 0 & \text{if } \overline{s_i} \subset R_g \\ \frac{\text{distance}(R_g, \overline{s_i})}{d_g} & \text{otherwise} \end{Bmatrix} \tag{9}$$

where $d_g = 2 \times \sqrt[3]{\frac{3V_g}{4\Pi}}$ is the average diameter of a grain g, V_g is the volume of the voxelized grain, and $\overline{s_i}$ is the shortest projected distance from a grain interface point to the surface of the reconstructed grain (R_g). Fig. 12a shows a plot of the normalized surface error for the reconstructed grain of Fig. 8. The plot shows that a majority of the points are within 50% of the experimental step-size. Fig. 12b shows good results of the average value of this error for all grains. Table 1 shows the best and worst normalized

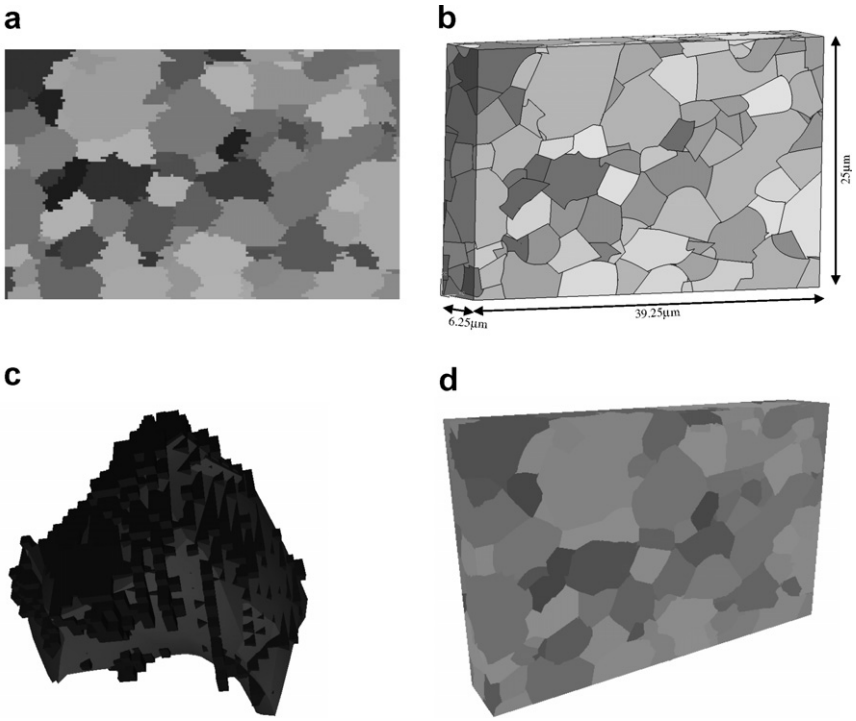


Fig. 11. (a) A single face of the voxelized volume from the FIB-SEM data, (b) the CAD-based 3D reconstruction of the polycrystalline microstructure that contains the section shown in (a) as one face (the dimensions are μm), (c) voxelized and CAD representation of a single grain, superimposed to illustrate the overall closeness of fit, and (d) microstructure after element switching in which local errors are corrected by overlaying elements on the voxel data and switching elements assigned to the wrong grain.

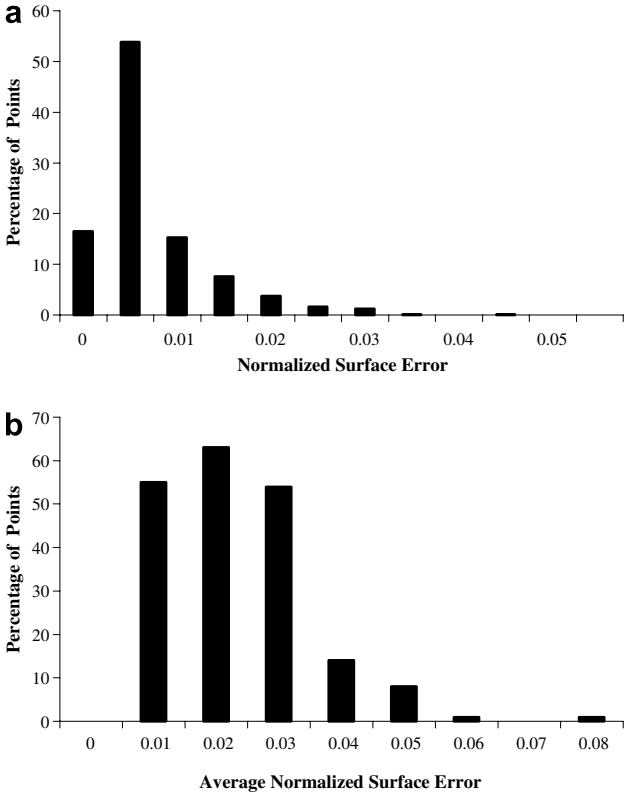


Fig. 12. Distributions of (a) the normalized surface error (dimensionless units) ε_i^g for a simulated grain, and (b) average normalized surface error $\varepsilon_{\text{average}}$ for all grains of Fig. 17.

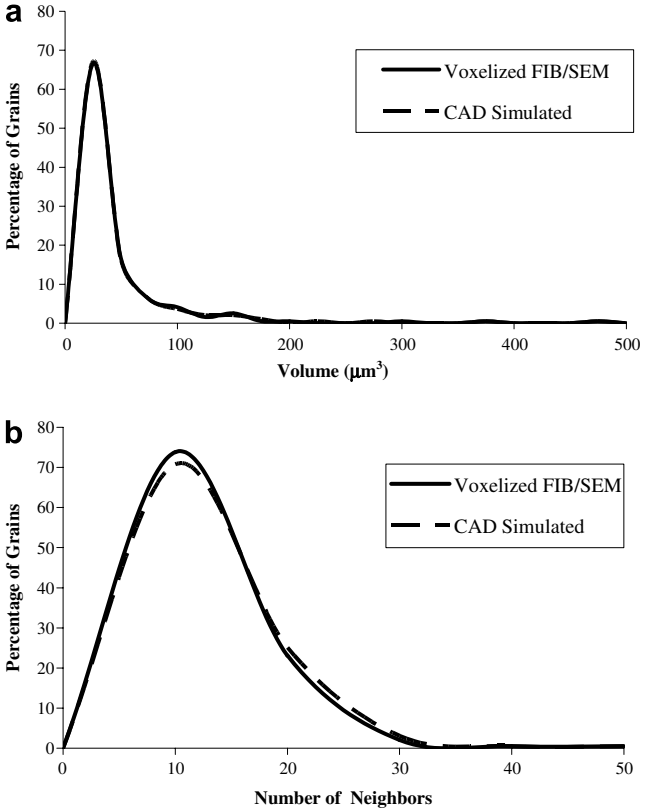


Fig. 13. Comparison of the distributions of (a) grain volume and (b) number of neighbors obtained from the experimental voxel-based microstructure, with that resulting from the for the reconstructed microstructure.

Table 1
Average normalized surface error for grains in the CAD-reconstructed microstructure

Grain	Average normalized surface error	Grain volume (μm^3)
Best grain	0.002264	472.1875
Median grain	0.018129	29.89538
Worst grain	0.07111	0.71875

surface error values for the data, revealing that larger grains with better relative resolution tend to have lower errors than the smaller ones. This result shows the importance of high-resolution experimental data for the process of reconstruction. Figs. 13 and 14 compare the distributions of grain volume, number of neighbors, and moments of inertia for the voxelized and reconstructed grain aggregates. Excellent match is observed for these parameters. However, as is expected, the distributions of grain surface areas in Fig. 15 show a significant difference between the two microstructure representations. The reconstructed smooth grain surfaces have smaller surface area than in the voxelized representation with the stepped surface.

5. Finite element mesh generation

For the nickel-based superalloy specimen considered, the voxelized microstructure representation contains approximately 125,000 grain faces. The CAD reconstructed representation reduces this to approximately 4000 grain faces. This will significantly lower the number of surface nodes for a FE mesh in comparison with the voxel-based mesh. Also, with the functional representation of the grain surfaces, it is possible to have control over the local density of the finite element mesh. Generating a compatible mesh that conforms to different complex shaped grains in the computational domain is a difficult enterprise with standard commercial software such as Unigraphics NX3 or Hyper-mesh. This section discusses a method for generating a high-quality compatible FE mesh of the grain structure that can be controlled for optimal local mesh density.

5.1. Compatible mesh generation

Let $\overline{N}_i = \textbf{Mesh\_Nodes}(R_i^{\text{og}})$ be a function that returns the set of nodes $\overline{N}_i$ obtained by discretizing a reconstructed grain $R_i^{\text{og}} \in R^{\text{og}}$ using any standard mesh generation routine, as in NX3. The steps in the algorithm for obtaining a mesh that conforms to the simulated polycrystalline microstructure R^{og} are:

1. For each grain $R_i^{\text{og}} \in R^{\text{og}}, i = 1, \ldots, M$, obtain $\overline{N}_i = \textbf{Mesh\_Nodes}(R_i^{\text{og}})$.

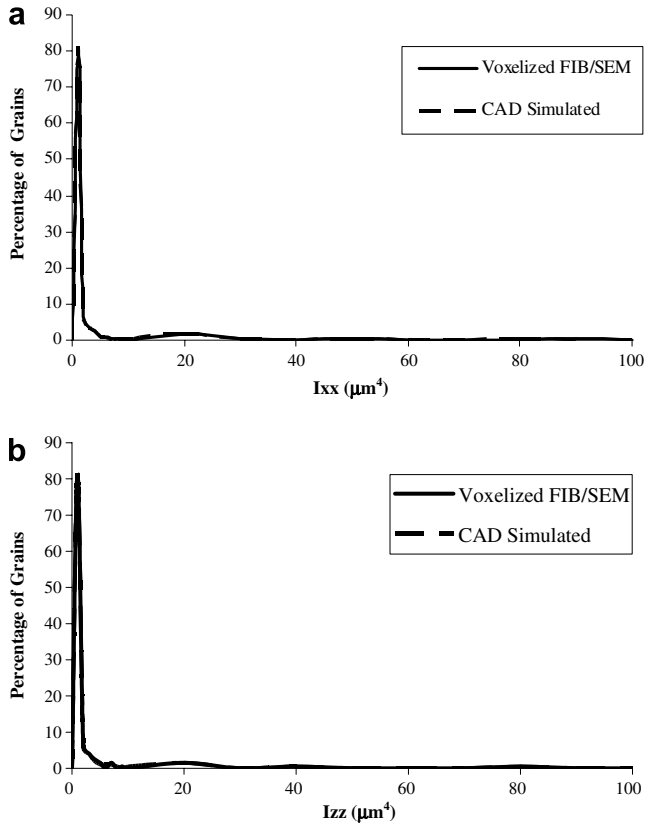


Fig. 14. Comparison of moments of inertia computed for both the voxelized and CAD-based reconstructions of the microstructure.

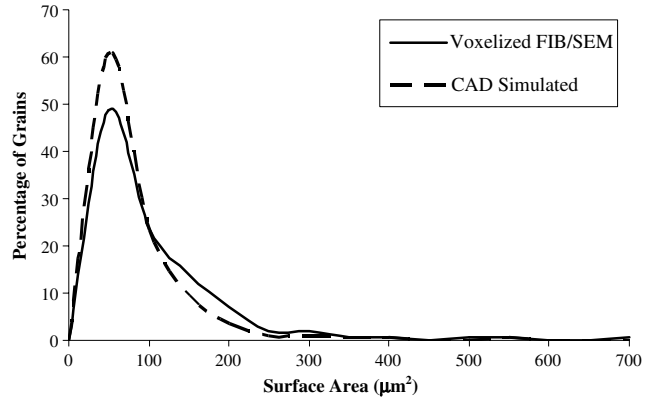


Fig. 15. Comparison of the distributions of grain-surface area for both the experimental voxel-based and the CAD-based microstructure reconstructions.

2. Obtain the assembly of all nodes for the polycrystalline aggregate $\overline{N} = \bigcup_{i=1}^{M} \overline{N}_i$. This is higher than the number of nodes in a compatible mesh with common nodes at the grain interface.
3. Identify and merge coincident nodes in the set $\overline{\mathbf{N}}$. The aggregate of all nodes serve as seed points for 3D Delaunay triangulation, a method that generates a collection of tetrahedra satisfying the empty sphere property [35,36].
4. Perform Delaunay triangulation to obtain a conforming mesh of tetrahedral elements in the FEM mesh using a freeware program Qhull [29].

The mesh generation algorithm is applied to the CAD reconstructed microstructure of Fig. 11b with a prescribed average element-size of $10 \times \Delta L$. The Delaunay triangulation produces a total of 40,632 nodes and 235,914 tetrahedral elements. The element quality is measured in terms of 'collapsibility index' (CI), which is the ratio of largest edge length of an element to its smallest altitude. A normalized collapsibility index (NCI) is defined as:

$$\mathrm{NCI} = \frac{\mathrm{CI} - \mathrm{CI}_{\min}}{\mathrm{CI}_{\max} - \mathrm{CI}_{\min}} \tag{10}$$

The NCI for the mesh is plotted in Fig. 16 with $\mathrm{CI}_{\min} = 1.29155$ and $\mathrm{CI}_{\max} = 4.13 \times 10^7$.

5.2. Mesh quality improvement

Local changes are made to the nodes and elements of the initial mesh to improve its quality without affecting the overall geometric representation of the microstructure. The process includes the following operations:

- *Node deletion*: Nodes that correspond to the shortest altitude in bad quality elements with high NCI are deleted. Delaunay triangulation is performed again with the reduced number of nodes.
- *Node relocation*: Element (E_w) with the highest NCI in the mesh is chosen for node relocation. Each node of this element is assigned a scalar value $\frac{1}{\mathrm{NCI}}$ for the worst element contiguous to E_w. The node with the smallest value of $\frac{1}{\mathrm{NCI}}$ is chosen for relocation. This node is then moved in a direction that minimizes the sum of the collapsibility index for all its contiguous elements.
- *Edge collapse*: The remaining bad elements after node deletion and node relocation can be eliminated by collapsing the nearest pair of nodes in these elements. Fig. 17 shows a schematic of the merger of two nodes at a common location. The corresponding common

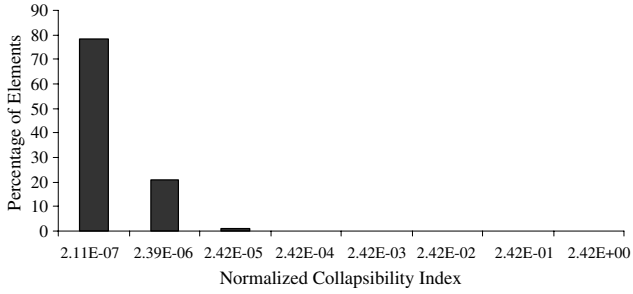


Fig. 16. Percentage of elements having a given value of normalized collapsibility index within the meshed representation of the microstructure.

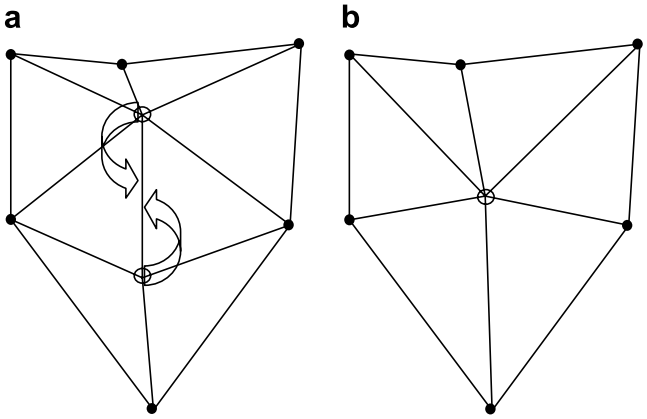


Fig. 17. Schematic illustration of the collapsing of two nodes in a mesh: (a) mesh before and (b) mesh after edge-collapse.

elements are deleted and the connectivity of nodes and elements is modified accordingly, as detailed in [30].

Following mesh quality improvement, each element is superimposed on the original voxel data. All voxels inside of the element are identified and the grain which they belong to, is also noted. The grain with the most voxels inside of the element is determined. If that grain is different than the grain that the element is currently assigned to, then the element is reassigned to the new grain. This process is implemented to correct small local errors seen in Fig. 11. As previously noted, while the surface fits generally match the voxel data well, the use of relatively low order surfaces allowed for some local deviations. The element switching process corrects these small errors, as seen in Fig. 11d. Fig. 18 shows the resulting mesh obtained for the microstructure after completing the quality improvement procedures. The mesh quality improvement data are listed in Table 2.

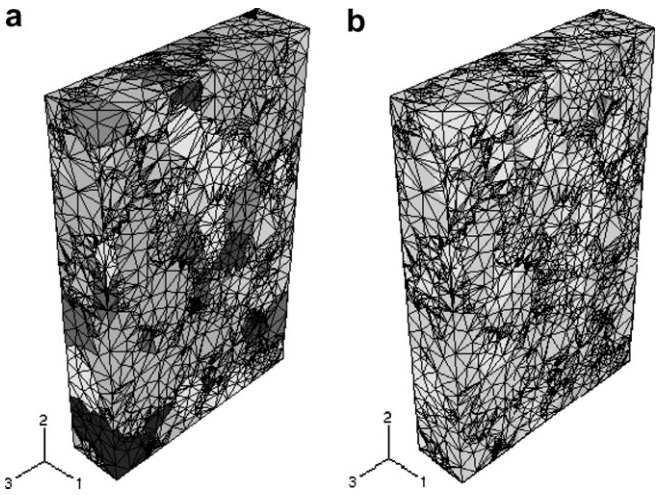


Fig. 18. (a) Meshed microstructure obtained after mesh quality improvement (colors assigned based on orientation of grains) (For interpretation of the references to colour in the figure legend, the reader is referred to the web version of this article.).

Table 2
Degrees of freedom and mesh quality parameters after various stages of mesh generation and quality improvement

Mesh after:	Mean normalized collapsibility index (CI)	No. of Nodes	No. of Elements
Delaunay triangulation	1.59×10^{-5}	40,632	235,914
Node deletion	2.91×10^{-7}	40,448	234,901
Node relocation	2.19×10^{-7}	40,448	234,901
Edge collapsing	1.68×10^{-7}	39,428	228,082
Element collapse	3.98×10^{-7}	36,716	211,768

5.3. Element collapse for enhanced efficiency

The mesh of tetrahedral elements accounting for the grain geometry only, generally has a uniformly high density. A uniformly fine mesh for the entire microstructure will yield good simulation accuracy but at a high computational cost. A coarse mesh, on the other hand, will compromise accuracy for gains in computational efficiency. To strike a balance, an algorithm that collapses smaller elements into larger elements in regions of low solution gradients is introduced. This algorithm takes into account the distribution of the evolving solution variables, and results in a graded mesh having highly refined elements in regions of critical geometry or evolving variables and larger elements elsewhere. For the case studied, the graded mesh is obtained a-posteriori by performing a crystal-plasticity finite element analysis of the polycrystalline microstructure (see Ghosh et al. [32–34]) with the initial mesh. Crystal-plasticity based finite element analysis is conducted for a uniaxial constant strain rate compression test by using the commercial package ABAQUS [31]. The local gradients of the Von Mises stress after a few steps in the analysis are shown in Fig. 18b. Even though the stress is generally uniform over large regions of the microstructure, stress concentrations exhibiting high gradients due to material inhomogeneity are observed at a few locations. The local stress gradients for each neighboring element-pair are calculated as the difference in the value of von Mises stress at the centroids divided by the distance between the centroids. For each element edge, the stress gradient associated with all contiguous elements is quantified. If these gradients are all less than a critical value, this edge is collapsed by the edge-collapse algorithm in Fig. 17. The consequent reduction in the degrees of freedom of the mesh is shown in Table 2.

6. Summary

This paper describes a cradle to grave methodology for creating a finite element mesh of a 3D polycrystalline material microstructure, starting from experimentally acquired data and incorporating CAD-based reconstruction tools. The method uses crystallographic orientation maps of consecutive serial sections of the material created by a DB FIB-SEM system. The microstructure reconstruction

methodology uses a hierarchical geometrical representation in conjunction with common primitives within CAD modeling. Grains are segmented as a collection of voxels using the crystallographic orientation data. Parametric surface–function segments are constructed using polynomial and NURBS functions to smooth the grain interfaces. A cell-decomposition algorithm generates 3D grains, bounded by the surface segments. Grain compatibility is ensured by the use of Boolean operations for gap and overlap removal. Error measures, including those in projected distances to the grain interfaces, distribution of volume, surface area and number of neighbors etc., are used to validate the accuracy of the simulated microstructure. The tests demonstrate that reasonably good microstructure representations can be generated by this method. Finally, the reconstructed domain is discretized into a volumetric mesh consisting of tetrahedral elements for use in FE simulations. A compatible mesh is generated by Delaunay triangulation of the microstructural domain. The triangulation process is followed mesh-quality improvement and element collapse for reducing degrees of freedom.

A significant strength of the method is that it permits allows monitoring and control of the resolution of the simulated microstructure, for optimal representation with respect to the accuracy and efficiency needed for modeling. These techniques are essential for micromechanical analysis within microstructurally sensitive simulation frameworks.

Acknowledgements

The authors acknowledge support from the Air Force Office of Scientific Research, through Grant # FA9550-05-1-0067 at OSU (Program Director: Dr. Brett Conner). The authors from OSU also acknowledge support from the Office of Naval Research through Grant # N00014-05-1-0504 (Program Director: Dr. Julie Christodolou). Computer support from the Ohio Supercomputer Center through Grant # PAS813-2 is also acknowledged.

References

[1] S.V. Harren, R.J. Asaro, J. Mech. Phys. Solids 37 (2) (1989) 91–232.
[2] G. Venkatramani, D. Deka, S. Ghosh, ASME J. Eng. Mater. Technol. 128 (3) (2006) 356–365.
[3] D. Deka, D.S. Joseph, S. Ghosh, M.J. Mills, Metal. Mater. Trans. A 17A (5) (2006) 1371–1388.
[4] S. Sinha, S. Ghosh, Int. J. Fatigue 28 (2006) 1690–1704.
[5] S.R. Kalidindi, C.A. Bronkhorst, L. Anand, J. Mech. Phys. Solids 40 (1992) 537–569.
[6] S. Balasubramanian, L. Anand, J. Mech. Phys. Solids 50 (2002) 101–126.
[7] P. Dawson, M. Miller, T.S. Han, J. Bernier, Metall. Mater. Trans. A 36A (July) (2005) 1627–1641.
[8] V.P. Bennett, D.L. McDowell, Int. J. Fatigue 25 (2003) 27–39.
[9] S.K. Burke, S. McK Cousland, C.M. Scala, Mater. Forum 18 (1994) 85–109.
[10] R.G. King, P.M. Delaney, Mater. Forum 18 (1994) 21–29.
[11] M. Berman, L.M. Bischof, E.J. Breen, G.M. Peden, Mater. Forum 18 (1994) 1–19.
[12] J.Y. Buffiere, E. Maire, C. Verdu, P. Cloetens, M. Pateyron, G. Peix, J. Baruchel, Mater. Sci. Eng. A 234–236 (1997) 633–635.
[13] R.N. Yancey, G.Y. Baaklini, Computed tomography evaluation of metal–matrix composites for aeropropulsion engine applications, in: Proceedings of ASME International Gas Turbine Aeroengines Congress, Cincinnati, OH, 1993.
[14] D.M. Saylor, J. Fridy, B.S. El-Dasher, K.Y. Jung, A.D. Rollett, Metall. Mater. Trans. A 35A (July) (2004) 1969–1979.
[15] M. Groeber, M.D. Uchic, D.M. Dimiduk, Y. Bhandari, S. Ghosh, A framework for automated 3D microstructural analysis & representation, in: Conference Proceedings Symposium 4, Multi-scale materials modeling, 2006.
[16] A.C. Lewis, A.B. Geltmacher, Scripta Mater. 55 (2006) 81–85.
[17] W.E. Lorensen, H.E. Cline, Comput. Graphics 21 (4) (1987) 163–169.
[18] N. Amenta, M. Bern, M. Kamvysselis, A new voronoi-based surface reconstruction algorithm, in: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, 1998, pp. 415–421.
[19] T.K. Dey, J. Giesen. Detecting undersampling in surface reconstruction, in: Proceedings of 17th ACM Symposium on Computational Geometry 2001, pp. 257–263.
[20] B. Geiger, Three-dimensional modeling of human organs and its applications to diagnosis and surgical planning, Technical Report, 2105, INRIA, France, 1993.
[21] G. Barequet, M. Sharir, Ann. Symp. Comput. Geometr. (1994) 93–102.
[22] C.L. Bajaj, E.J. Coyle, K.N. Lin, Graph. Models Image Process. 58 (6) (1996) 524–543.
[23] M.A. Groeber, B.K. Haley, M.D. Uchic, D.M. Dimiduk, S. Ghosh, 3D reconstruction and characterization of polycrystalline microstructure using a FIB-SEM system, Mater. Char. 57 (2006) 259–273.
[24] U.F. Kocks, C.N. Tomé, H.R. Wenk, Texture and Anisotropy: Preferred Orientations in Polycrystals and Their Effect on Materials Properties, Cambridge University Press, Cambridge, 1998.
[25] C.L. Xie, S. Ghosh, M. Groeber, J. Eng. Mater. Technol. 126 (October) (2004) 339–352.
[26] L. Piegl, IEEE Comput. Graphics Appl. 11 (1) (1991) 55–71.
[27] G. Farin, IEEE Comput. Graphics Appl. 12 (5) (1992) 78–86.
[28] Unigraphics NX3.0.4.2, User Manual, UGS corporation, 2004.
[29] C.B. Barber, D.P. Dobkin, Huhdanpaa H. Qhull, ACM Trans. Math. Soft. (December) (1996) 469–493.
[30] H. Hoppe, Progressive meshes, in: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, 1996, pp. 99–108.
[31] Abaqus reference manuals, Hibbit, Karlson and Sorenson Inc. Providence, RI, 2001.
[32] S. Balasubramanian, Polycrystalline Plasticity: Application to Deformation Processing of Lightweight Metals, Ph.D. dissertation, MIT, Cambridge, MA, 1998.
[33] V. Hasija, S. Ghosh, M.J. Mills, D. Joseph, Acta Mater. 51 (2003) 4533–4549.
[34] C.A. Bronkhorst, S.R. Kalidindi, L. Anand, Philos. Trans. R. Soc. London, Ser. A 341 (1992) 443–477.
[35] D.T. Lee, B.J. Schachter, Int. J. Comput. Informat. Sci. 9 (1980) 219–242.
[36] A. Okabe, B. Boots, K. Sugihara, Spatial Tessellations: Concepts and applications of Voronoi diagrams, Wiley, New York, 1992.