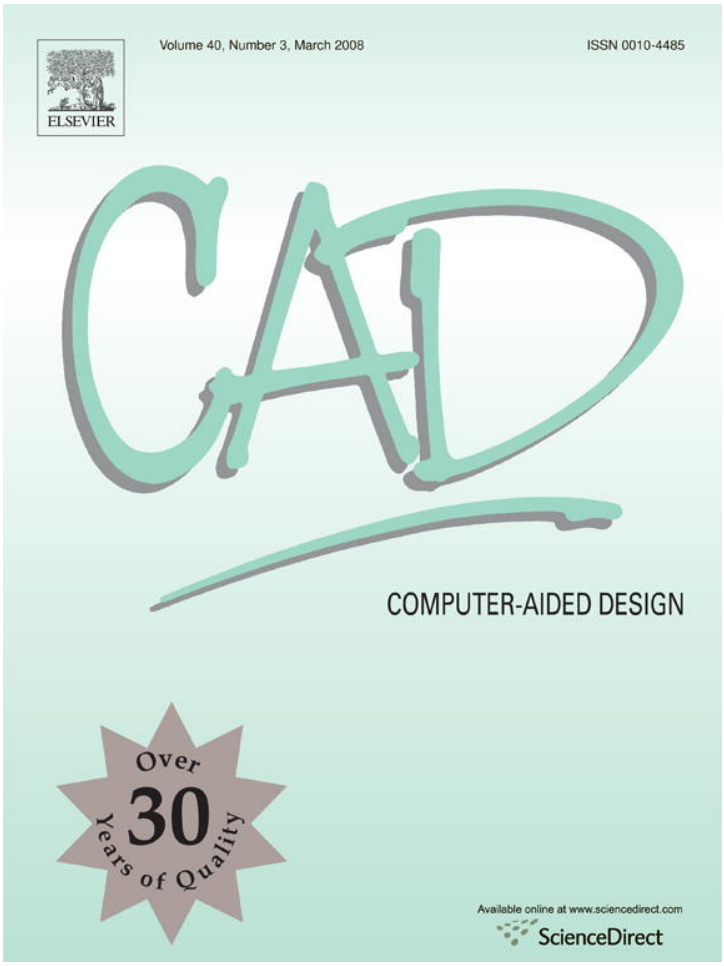

Available online at www.sciencedirect.com
ScienceDirect
Computer-Aided Design 40 (2008) 293–310
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

CAD-based reconstruction of 3D polycrystalline alloy microstructures from FIB generated serial sections

S. Ghosh*, Y. Bhandari, M. Groeber

Mechanical Engineering, The Ohio State University, Columbus, OH-43210, USA

Received 6 October 2006; accepted 3 November 2007

Abstract

This paper develops a robust CAD-based methodology for simulating 3D microstructures of polycrystalline metals using crystallographic input data on sections created by a focused ion beam (FIB)–scanning electron microscopy (SEM) system. The method is able to construct consistent polycrystalline microstructures with control on the resolution necessary for meaningful computational analysis in microstructure-property estimation. The microstructure simulation methodology is based on a hierarchical geometrical representation using primitives used in CAD modeling. It involves steps of data cleanup, interface point identification, polynomial and NURBS function-based parametric surface segments construction, generalized cell decomposition, geometric defeaturing, and gap and overlap removal. The implementation of the entire procedure described above is performed with the aid of user-programming facilities of a commercial CAD package Unigraphics NX3. The microstructure simulation algorithm is validated using various error criteria and measures for an extracted microstructure of a nickel superalloy.

Keywords: FIB–SEM; 3D microstructure simulation; Parametric surface; NURBS; Gap-overlap removal

1. Introduction

Advanced metallic materials used in many industrial applications have complex multi-colony, multi-phase polycrystalline aggregates in their microstructure as shown in Fig. 1. The mechanical behavior and fatigue failure response are intricately governed by microstructural features that include morphological and crystallographic characteristics, e.g. shape, size and location of phases in the colony structure, relative colony size and locations, crystal orientations and misorientations, grain boundary geometry etc. Detailed micromechanical computational models are being used to understand deformation and damage mechanisms and throw light on the stochastic nature of failure and fatigue phenomena of these materials [1–10]. While, the computational models of polycrystalline materials implementing crystal plasticity models are making great strides in predicting the stress–strain behavior with reasonable accuracy, ductility and fatigue failure predictions with high fidelity are still far from mature. Morphological and crystallographic heterogeneities in the microstructures result in strong anisotropy and localized non-homogeneous deformation, which impose severe challenges to these computational models. Experimental studies [11] suggest that the growth of crystallographic microslip bands along active slip systems of plastic flow causes localized instability due to compatibility requirements between interacting grains. They continue to grow across grain boundaries due to grain structure instability and eventually manifest as macroscopic shear bands. The interaction of microscopic shear bands with transverse grain boundaries also leads to grain boundary microcracking, which grows in size and merge to cause fracture.

It is important for computational models to capture the 3D geometric and crystallographic details of grain morphology, as well as their distribution in the polycrystalline aggregate for robust prediction of their properties. An automated approach of characterizing 3D microstructure using a dual beam focused ion beam (FIB)–SEM system has been recently developed [26] to acquire 3D orientation data of a succession of sections in the material microstructure. Using a FIB column in the microscope, highly localized micromachining and ion imaging is conducted. Following this, high resolution electron back-scatter diffraction (EBSD) images are acquired by a SEM column for grain orientations. These experimental advancements have made

* Corresponding author. Tel.: +1 614 292 2599; fax: +1 614 292 3163.
E-mail address: ghosh.5@osu.edu (S. Ghosh).

0010-4485/$ - see front matter
doi:10.1016/j.cad.2007.11.003

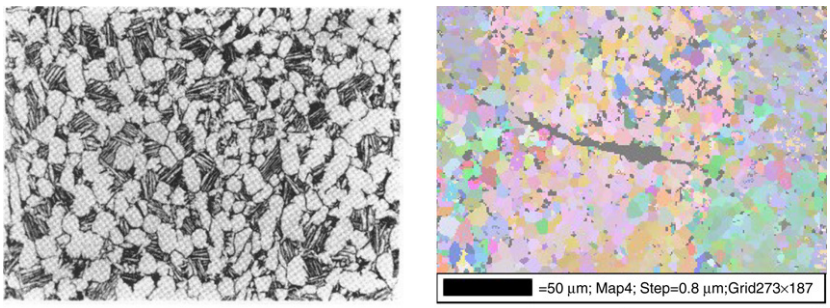

Fig. 1. (a) Optical micrograph α/β forged Ti-6242 alloy; (b) an orientation imaging microscopy image of the alloy showing a fatigue crack.

it possible to seamlessly reconstruct high fidelity 3D grain and subgrain microstructures of polycrystalline materials. The high fidelity 3D microregions can then be discretized and analyzed by computational methods like the finite element analysis methods for an accurate and reliable prediction of material properties. This paper develops a CAD-based method of creating 3D grain structures through post-processing of the FIB–SEM generated OIM data. A seamless reconstruction process will have certain characteristics and features that are summarized below.

- *Input/output data*: The input to the process is crystallographic orientation data of a metallic specimen in a 3D square grid. The output is a collection of solid bodies, with each body representing an individual grain. A requirement is that there be no overlap or gaps between them.
- *Data uncertainty*: Uncertainty corresponding to un-indexed points, incorrectly indexed points, misalignment, scatter marks etc. are to be expected in the experimentally acquired data, and has to be effectively dealt with.
- *Process automation*: The entire process from reading of the experimentally generated orientation maps to the creation of grain geometries in the aggregate should be automated, such that no, or minimum, additional user input is required.
- *Robustness*: The grain ensemble reconstruction procedure should be adequately robust to deal with different material microstructures, for which the sectional data is available. This requires identification of unstable operations and their removal.
- *Requirements for finite element mesh*: Typically crystal plasticity simulations of polycrystalline microstructures require prohibitively high computations, especially for models that represent microstructural details. It is therefore desirable to generate the microstructural details, keeping in mind both accuracy and efficiency considerations. Optimal representation with respect to the number of nodes and elements in the finite element mesh should be generated to retain both accuracy and efficiency of the eventual computational analysis.

This paper addresses the development of a seamless methodology for simulating polycrystalline metal microstructures from FIB–SEM generated serial sections using primitives used in CAD methods. A unique strength of this method is that it is entirely possible to monitor and control the resolution of the simulated microstructure for accuracy and efficiency needed for materials modeling. A commercial CAD package Unigraphics NX3 [31] (henceforth referred to as NX3) is used to perform all operations in the polycrystalline microstructure reconstruction. NX3 allows direct access to most of its geometric modeling and manipulation facilities through Open C API interface. A special module has been developed through this interface to reconstruct microstructure without any user intervention. Section 2 reviews some of the related work in this general area. Section 3.1 discusses steps for data collection and cleanup procedures, while Sections 3.2–3.4 describes the reconstruction process. Finally validation of this method with respect to microstructural characteristics is discussed in Section 4.

2. Brief review of microstructure reconstruction methods

Since polycrystalline deformation is predominantly 3D in nature, it is essential that the microstructural models be developed with detailed 3D information. Techniques based on ultrasonics or its variants, such as acoustic microscopy or laser ultrasonics [12,13] rely on good reflection properties and have limited application in metals. While X-ray-based computed tomography [14,15] methods are widely used in 3D solid model generation, they are generally deficient in achieving the resolution desired for the detailed study of polycrystalline metals. Synchrotron-based CT technology have been developed to yield tomographic images with considerably high resolution [16]. However, this method is still not commercially available and in generally is quite expensive. A few notable recent developments in microstructure representation are showing considerable promise. Of these, models that involve statistical extrapolations from 2D surface or section images [17], and the morphologically 'precise' models of 3D reconstruction from FIB–SEM generated serial-section data [18] are gaining considerable attention. The former approach [17] has the advantage of not relying on exhaustive (often destructive) experiments to obtain the crystallographic information. However, the reliability of these methods in reproducing important microstructural characteristics depends on the accuracy of the statistical interpolators and methods. This is a non-trivial task and can sometimes cause large errors if sufficient constraints are not developed in the statistical interpolation space. Reconstruction methods from 3D sections [18], on the other hand can be experimentally challenging and are generally destructive in nature.

3D microstructures are conventionally represented in two ways, viz. (i) as a collection of voxels (fusion of the words *volumetric* and *pixel)*, each containing the local orientation data, and (ii) as ensembles of grains, in which the grain features (e.g. surfaces, edges) are constructed in terms of parametric functions. Lewis and Geltmacher [19] have developed the voxel-based approach, where a typical microstructure containing 138 stainless steel grains are discretized into 3.5 million voxels. To avoid prohibitive computational costs incurred due to the large number of voxel elements in a finite element mesh, they combined four voxels in each direction to construct their finite element model. Important geometric information can be lost in this process if the voxel collapsing is not done in relation to the local grain boundary geometry. Various methods of parametric reconstruction have been proposed in the literature. The marching cube method [20] constructs triangular models of constant density in 3D by linear interpolation of density values in the image. Methods developed in [21,22] use Voronoi diagrams and for reconstruction based on unstructured sample points on a smooth surface. Other techniques have been developed in medical imaging [23–25] using serial-sectioned images to construct 3D images by Delaunay triangulation.

A hierarchical parametric representation of grain vertices, edges and surfaces is proposed in the present work to construct the 3D volumetric domains of polycrystalline microstructures. The method can have significant advantages over the voxel-based reconstruction methods by adaptively controlling the required resolution. Each grain is modeled as a solid bounded by large arbitrarily-shaped surface patches. These surface patches can capture the smoothness of actual grain boundary interfaces well, while still allowing mesh of almost any size. The resulting finite element model can enjoy accuracy with optimal computational efficiency.

3. Reconstructing the 3D polycrystalline microstructures

Fig. 2 is a flowchart of the sequence of operations that are necessary to construct the 3D microstructure from experimentally acquired section data. Each of the steps is discussed in this section.

3.1. FIB–SEM data acquisition and pre-processing

The basic steps involved in data pre-processing include collection of data from experiments, segmentation of individual grains, data cleanup and alignment. Only after performing all of these steps, the data will be usable enough for any form of geometric reconstruction.

3.1.1. Data acquisition

The sequence of operation involved in 3D data acquisition by the dual beam focused ion beam–scanning electron microscope (FIB–SEM) system is described in detail in Groeber et al. [26]. As shown in Fig. 3, the experiment entails moving a metallic specimen repeatedly between the two microscopic stage positions, viz. the "sectioning" position and the "electron back scatter diffraction (EBSD)" position. Image recognition techniques are used for automated alignment of the sample. When the sample is in the "sectioning" position, the FIB is used to mill a cross-section surface. After sectioning, the sample stage is rotated and translated to bring the cross-section face into the EBSD analysis position. In this position, crystallographic orientation measurements are made at points on a square grid in each section. Fig. 4(a) shows a series of orientation maps that are stacked together to form a 3D rendering of the crystallographic orientation in Fig. 4(b). In this rendering, each voxel is constructed by 3D extrusion of each pixel on the 2D orientation maps through the sectioning thickness of the FIB process.

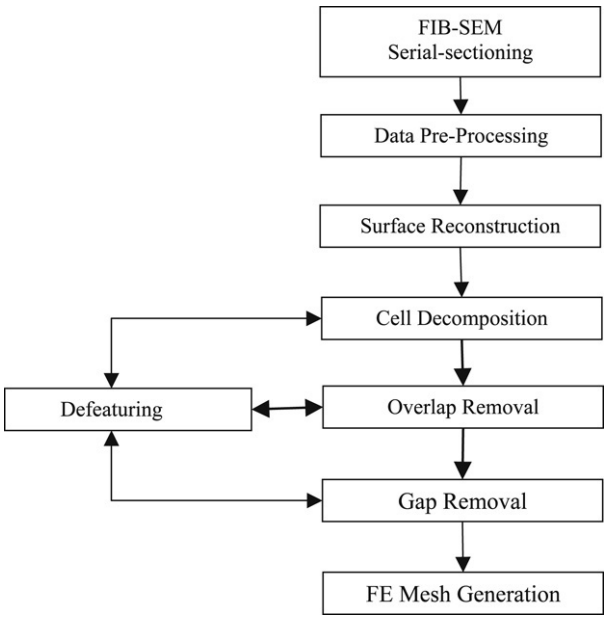

Fig. 2. A flow chart showing the steps for the reconstruction of 3D polycrystalline microstructure from FIB–SEM generated serial-sectioned data.

3.1.2. Grain segmentation

This 3D orientation data set can be used to perform segmentation of the individual grains in the microstructure. At each point of the microstructure, the data provides information on the set of Euler angles (φ_1, Φ, φ_2) that define the crystallographic orientations. In [27,28] the significance of Euler angles is described in terms of the angular rotations of the global coordinate system to obtain the local coordinate system of the crystalline lattice. The latter can be obtained by rotating the global coordinate system first by a rotation by φ_1 about z-axis, then Φ about x-axis and φ_2 about z-axis again. The misorientation between two neighboring voxel points A and B can be calculated from their respective orientations as [27,28]:

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

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

Several identical rotation operations (24 for cubic systems, 12 for hexagonal close packed systems) result when symmetry

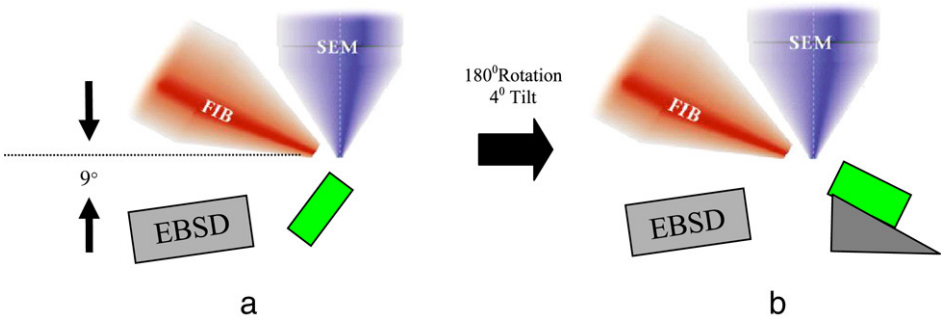

Fig. 3. Schematic of the FIB–SEM-based serial-sectioning process: (a) sectioning position (b) EBSD position.

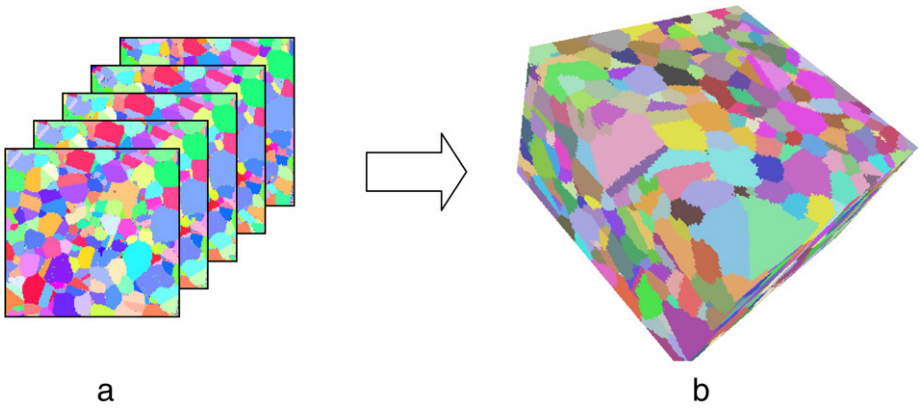

Fig. 4. (a) Stacking of individual 2D orientation maps to obtain the reconstructed 3D maps; and (b) 3D volume rendering from the 2D images.

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

Box I.

of crystal lattices is taken into account through the crystal symmetry operator O. The misorientation angle between A and B is taken to be the minimum θ in Eq. (1). A preset misorientation tolerance (~4° in this study), is used to delineate a grain boundary between two points belonging to different grains. The steps in the segmentation algorithm are given below:

1. Mark all points in the microstructure as unassigned. Tag a grain number, currently under consideration as $N_{\text{current}} = 1$
2. Select an unassigned point P, that has the highest reliability of EBSD measurement $r_P = r_{\max}$. Reliability of the EBSD measurement is a scalar value obtained during the experiment corresponding to the noise in the EBSD patterns. Higher lattice distortion, common near the grain boundaries, results in more noise and worse reliability. Using the reliability in identifying the starting point for grain assignment can reduce noise in grain identification.
3. If $r_{\max} < r_{\text{tolerance}}$ then exit, otherwise assign N_{current} to P.
4. For each unassigned neighbor of P, find out if its misorientation with P is within the preset tolerance. If this is met, then assign it a grain number N_{current}.
5. Continue with 4 recursively till no new points can be found.
6. Set $N_{\text{current}} = N_{\text{current}} + 1$ and go to 1 till all points have been assigned.

Some grains may contain extremely narrow 'twin' regions with a special misorientation relationship with respect to the parent grain. Due to poor data resolution of the experiment, their thickness is not usually well sampled, making their reconstruction extremely difficult. Due to this reason, all of the twins are merged with their parent grain at this stage.

3.1.3. Section data cleanup

In Section 3.1.2, all the contiguous points having similar orientations are identified as belonging to a single grain. However, limitations of the experimental procedure may leave some of the points un-indexed or with unreliable orientation information. This is commonly observed for points that are close to the grain boundaries. The lattice structure at these points may be distorted enough to render the readings unreliable or unusable. Presence of inclusions may sometimes also generate this noise in the data. The data cleanup procedure heuristically involves assigning these points to the appropriate reliable grains. Different procedures are implemented corresponding to the noise.

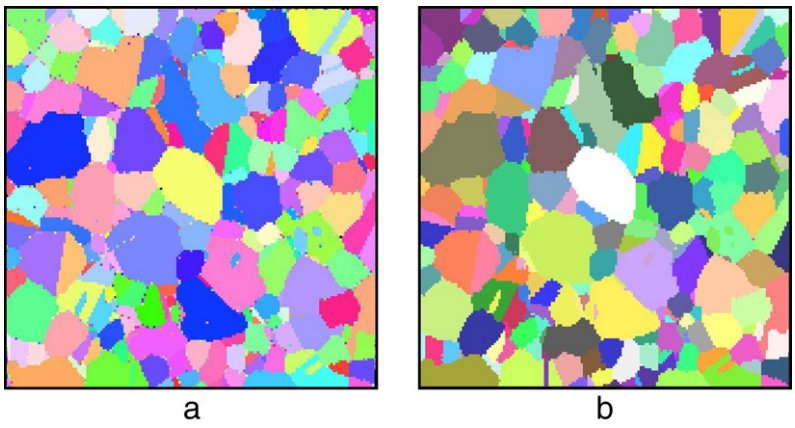

Fig. 5. Slice of data (a) as received from the experiment; and (b) after data cleanup where unreliable and un-indexed data, in the form of speckles have been removed.

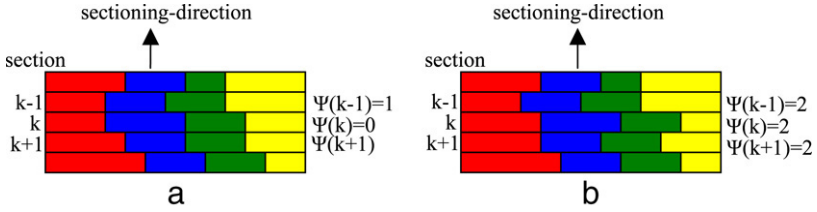

Fig. 6. Schematic of grain orientation measures in sequence of sectioned surfaces showing (a) aligned data and (b) misaligned data along the kth section. Values of $\Psi(k)$ are larger for misaligned sections.

- *Unassigned indexed points*: Points which have orientation information but are unassigned to any grain are assigned to that neighboring grain with which they have least misorientation.
- *Un-indexed points*: Points which do not have any orientation information are assigned to a grain with which they share the most surface area.
- *Enclosed grains*: Grains with sizes less then a given tolerance (64 data points in this paper corresponding to a 1 μm^3) or grains that are contained completely inside any other grain are usually inclusions and therefore are merged.

The data cleanup procedure, when applied to the experimental data of Fig. 5(a) results in the clean image of Fig. 5(b). Noisy points, usually observed as speckles are completely absent in the later image.

3.1.4. Section alignment

Serial-sectioning involves the rotation of the stage between the scanning and cutting positions as shown in Fig. 3. Although the alignment errors are kept in check through the use of optical recognition technologies, some alignment errors may still be present in the resulting data. These errors can be minimized by moving the misaligned slices of data in x- and y-directions, orthogonal to the sectioning direction, by appropriate distances. Since the section data is in the form of a 3D square grid, these corrections must correspond to the multiples of step-size or sectioning thickness (ΔL) of the experiments. Schematics of aligned and misaligned sections corresponding to individual grains are shown in Fig. 6. The misalignment of a pixel in a given direction of the sectioned plane causes the grain boundary to appear jagged. A measure of this jaggedness can be defined as:

$$\Psi(i,j,k) = \begin{cases} 1 & \text{if } G(i,j,k) \neq G(i,j,k-1) \\ & \text{and } G(i,j,k) \neq G(i,j,k+1) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Here, $G(i,j,k)$ corresponds to the grain number at the (i,j,k)th data point corresponding to the x, y and z-directions respectively. The measure of jaggedness for the section (k) is given as:

$$\Psi(k) = \sum_{j=0}^{y_{\max}} \sum_{i=0}^{x_{\max}} \Psi(i,j,k), \tag{3}$$

where, $x_{\max}$ and $y_{\max}$ are the total number of data points in the corresponding directions. The measure of the section jaggedness parameter $\Psi(k)$ is high for a misaligned section (see Fig. 6). Grain boundary energetics makes it difficult to observe a 'bump' on most of the grains along a particular direction on a given section. This observation is the basis of the algorithm to perform automatic alignment of the 3D serial-sectioned data. In this process, the following steps are performed for each section $k = 3$ to $z_{\max} - 2$.

1. Calculate $\Psi(k)$, $\Psi(k-1)$, $\Psi(k+1)$ using Eq. (3).
2. Set $\Delta\Psi^{x+}_{\max} = 0$ and $n = 1$, where $\Delta\Psi^{x+}_{\max}$ is the maximum value of $(\Psi(k) - \Psi^{x+n}(k))$ and n corresponds to voxel data points.
3. Offset the slice k by n data points in the positive x-direction and calculate $\Psi^{x+n}(k)$, $\Psi^{x+n}(k-1)$, $\Psi^{x+n}(k+1)$. Here $\Psi^{x+n}(k)$ is the jaggedness measure of section k, when it is displaced by n voxels in the positive x-direction.

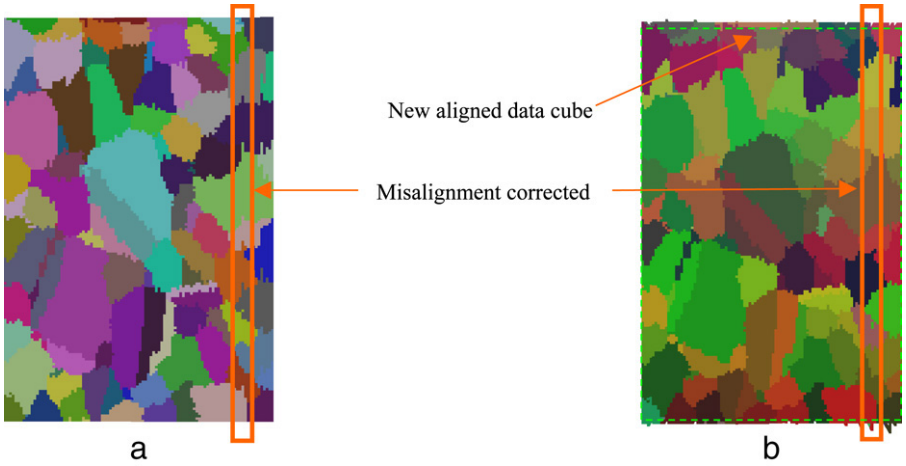

Fig. 7. A pixel-based representation of microstructure: (a) before alignment with bumps on all grains along the misaligned section; and (b) after alignment with little or no bumps.

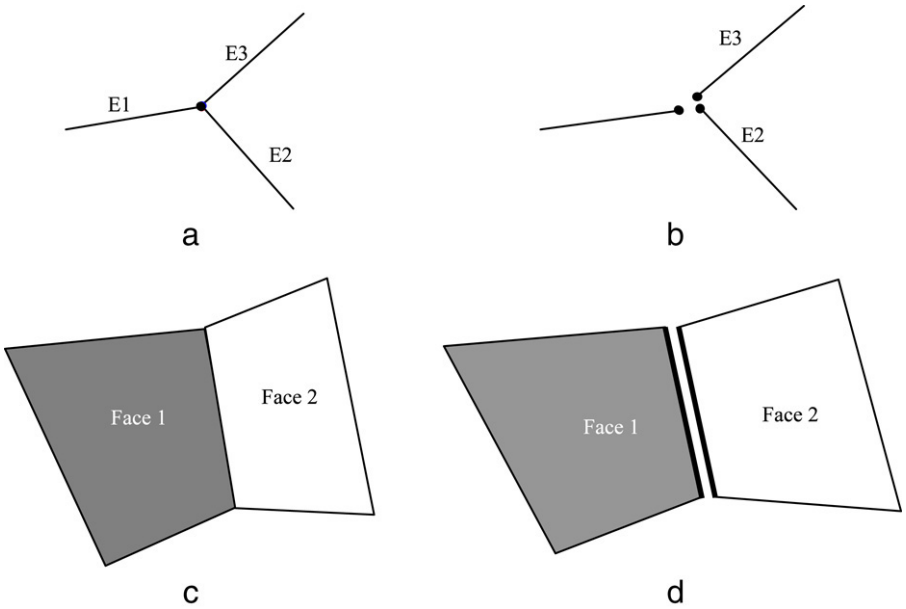

Fig. 8. Additional validity checks introduced for reconstructed grains: (a) vertex-consistent geometry; (b) vertex-inconsistent geometry; (c) edge-consistent geometry and (d) and edge-inconsistent geometry.

4. Reset the section k to original position. If $\Psi^{x+n}(k) < \Psi(k)$, $\Psi^{x+n}(k-1) < \Psi(k-1)$, $\Psi^{x+n}(k+1) < \Psi(k+1)$ and $\frac{\Psi(k)-\Psi^{x+n}(k)}{\Psi(k)} >$ preset tolerance then proceed to step 5, otherwise skip to step 6.
5. If $\Delta\Psi^{x+}_{\max} \geq (\Psi(k) - \Psi^{x+n}(k))$, then $n-1$ is the positive x-offset required for the section k and proceed to step 7. Otherwise set $\Delta\Psi^{x+}_{\max} = (\Psi(k) - \Psi^{x+n}(k))$, $n = n+1$ and go to step 3.
6. Repeat steps 2–5 in the negative x-direction.
7. Repeat steps 2–6 for the positive and negative y-direction.

The main idea behind the alignment algorithm is that a misaligned section, when moved in the direction opposite to the misalignment, will reduce $\Psi(k)$ for section itself, as well as for sections above and beneath it. With this alignment criterion, the algorithm generates a list of distances along the x- and y-directions, by which each section must be offset. Fig. 7 shows the result of this algorithm on actual data, which is physically offset to show the improvement in alignment. To account for the missing data during this physical offset, the new data cube after alignment has to be cropped down by the amount offset. Also, to prevent a few local natural bumps to be replaced by artificial bumps due to alignment, the preset tolerance in step 4 is set to 10%. One of the shortcomings of this algorithm is that the minimum distance by which each section can be moved is limited to ΔL. Hence the alignment errors below this limit may not be completely corrected.

3.1.5. Preprocessing applied to a polycrystalline microstructure

To demonstrate the validity of the pre-processing algorithm, it is tested on a specimen of a fine-grained polycrystalline nickel-base superalloy, IN100. After alignment, the dimension of the data cube obtained is 6.25 μm × 41.25 μm × 25.0 μm in

the x-, y- and z-directions with step-size $\Delta L = 0.25$ μm in each direction, corresponding to a total of 412 500 data points. After segmentation and cleanup a total of 196 grains are identified for this data. Alignment is limited to move the data sections in integral multiples of the step-size ΔL. Even, the process of collection of data itself is limited to the step-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 scale for reporting errors.

3.2. Domain construction for individual grains

At the start of this reconstruction process, each grain in the microstructure is represented as a set of contiguous voxels in 3D raster. Metallographic observations show that while it is common to have grain interfaces with significant undulations and sharp curvatures when multiple grains are involved, the interface shared by two grains can be smooth. Consistent with this observation, smooth interface patches are first generated by the interpolation of interface points shared by each pair of grains. Two types of interpolation functions, viz parametric polynomial functions or Non-Uniform Rational B-Spline or NURBS functions are used [29,30]. Following surface reconstruction, the volumetric domain of each grain is constructed by a cell-decomposition process. Compatibility and defeaturing of these grains will then be discussed.

3.2.1. Valid geometric entities in the reconstruction

Before discussing the steps of grain surface and volume generation, it is important to delineate valid geometric operations that yield topologically consistent geometric constructs. Examples of invalid operations include exceptional cases that may arise due to ill-conditioned equations, instability of the geometric kernel etc. To avoid such inconsistencies, checks available in NX3 are activated for the following.

1. Data structure problems, such as corruption
2. *Face–face intersections*, to see if all the faces of a solid meet at their edges only
3. *Consistency*, to see if (i) the topological structure is consistent, (ii) the geometrical objects are valid and that faces and edges have G1 continuity, and (iii) the geometrical objects are consistent, i.e., point geometry lies on the edges that it is linked to, edges on the faces and so forth.
4. *Smoothness* of the B-surface of each face along the patch boundaries
5. *Self-intersection* of each face of the solid
6. *Spike or cuts* on each face of the solid
7. *End-point consistency,* such that the solid has edges meeting at a single point as shown in Fig. 8(a), An end-point inconsistent solid, shown in Fig. 8(b), will have at least one pair of edges that do not meet a common point.
8. Edge consistency, such that each face-edge is shared between exactly two faces as shown in Fig. 8(c). If at least one face-edge exists that belongs to only one face as seen in Fig. 8(d), then it is an edge-inconsistent solid.

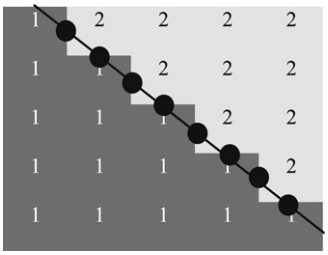

Fig. 9. Interface points identified as black dots between two adjacent grains 1 (dark grey voxels) and 2 (light grey voxels), and an interpolated surface (in this case a straight line) that is fit to these points.

The last two checks are unavailable in NX3 and are additionally introduced to further improve the robustness of the geometric operations. If the result of any modeling operation generates a solid that violates any of these checks, then the operation is considered as invalid. In these special cases, alternative methods of performing the same operation may have to be developed. For example, if subtraction of *solid B* from *solid A* fails, then B is adaptively broken into smaller pieces, each of which is then subtracted from A.

3.2.2. Surface reconstruction using parametric interpolation

Fig. 9 depicts surface points ($\overline{s}$) that are identified as belonging to a pair of grains. The grain-pair interfaces can be represented by different types of parametric surfaces including ruled surfaces, parametric polynomial surfaces or NURBS surfaces, fit through these surface points. The only constraint on these surface patches is that they have to be smooth, accurate and extendable. Due to the amenability of their representation in a standard CAD/FEA environment, parametric polynomial and NURBS surface-fitting are chosen, and the better fit among them is selected for representing a grain surface.

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

$$\begin{aligned} x &= x(u,w) = x_{00} + x_{01}u + x_{10}w \\ &\quad + x_{11}uw + \cdots + x_{nn}u^n w^n \\ y &= y(u,w) = y_{00} + y_{01}u + y_{10}w \\ &\quad + y_{11}uw + \cdots + y_{nn}u^n w^n \\ z &= z(u,w) = z_{00} + z_{01}u + z_{10}w \\ &\quad + z_{11}uw + \cdots + z_{nn}u^n w^n \quad \forall u, w \in [0,1] \end{aligned} \tag{4}$$

or in a more compact form,

$$\bar{Q}(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{n} \bar{Q}_{ij} u^i w^j \tag{5}$$

where, $\bar{Q} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$ and $\bar{Q}_{ij} = \begin{Bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{Bmatrix}$. A least square fit of the resulting surface, obtained for a set of N points, $\overline{\mathbf{S}} = \{\overline{s_1}, \overline{s_2}, \ldots, \overline{s_N}\}$ with parametric coordinates $\mathbf{U} = \{(u_1, w_1), \ldots, (u_N, w_N)\}$ can be derived by minimizing a

residual R_d, defined for each point (x_d, y_d, z_d) as:

$$R_d^2 = \left(x_d - \sum_{i=0}^{n}\sum_{j=0}^{n} x_{ij} \times u_d^i \times w_d^j\right)^2 + \left(y_d - \sum_{i=0}^{n}\sum_{j=0}^{n} y_{ij} \times u_d^i \times w_d^j\right)^2 + \left(z_d - \sum_{i=0}^{n}\sum_{j=0}^{n} z_{ij} \times u_d^i \times w_d^i\right)^2. \quad (6)$$

The square of residual error can be minimized by setting $\frac{\partial \sum_{d=1}^{N}(R_d^2)}{\partial x_{lk}} = \sum_{d=1}^{N} \frac{\partial (R_d^2)}{\partial x_{lk}} = 0$ with respect to each of the unknown variables x_{lk}, resulting in

$$\sum_{d=1}^{N}\left(x_d \times \left(u_d^l w_d^k\right)\right) = \sum_{d=1}^{N}\left(\left(\sum_{i=0}^{n}\sum_{j=0}^{n} x_{ij} \times u_d^i \times w_d^j\right) \times \left(u_d^l w_d^k\right)\right). \quad (7)$$

Eq. (7) can be represented in a matrix form as

$$\mathbf{A^T A X} = \mathbf{A^T S_x} \quad \text{or} \quad \mathbf{X} = (\mathbf{A^T A})^{-1}\mathbf{A^T S_x} \quad (8)$$

where $\mathbf{A}$ is a rectangular matrix containing exponents of the parametric coordinates (u_d, w_d), $\mathbf{S_x}$ is a vector of known coordinates (x_d, y_d, z_d), and $\mathbf{X}$ is the solution vector of the unknown coefficients in Eq. (4). The parametric coordinates (u_d, w_d) for each point $[x_d, y_d, z_d]^T$ in Eq. (7) are not known in advance and hence an iterative scheme is implemented for their determination. The Newton–Raphson has been used to minimize the residual in Eq. (9) by varying $\mathbf{U} = \{(u_1, w_1), (u_2, w_2), \ldots, (u_N, w_N)\}$, i.e.

$$\text{Minimize} \left(\left(x_d - \sum_{i=0}^{n}\sum_{j=0}^{n} x_{ij} u^i w^j\right)^2 + \left(y_d - \sum_{i=0}^{n}\sum_{j=0}^{n} y_{ij} u^i w^j\right)^2 + \left(z_d - \sum_{i=0}^{n}\sum_{j=0}^{n} z_{ij} u^i w^j\right)^2\right). \quad (9)$$

An initial guess of the parametric coordinates (u_d, w_d) is needed to start this adaptive fitting. The projection of $[x_d, y_d, z_d]^T$ onto the best-fit plane P of the N points is taken as an initial guess in this iterative method, for which the sequence of steps is given below:

1. Fit a plane $ax + by + cz = 1$ over the set of data points $\overline{\mathbf{S}} = \{\overline{s_1}, \overline{s_2}, \ldots, \overline{s_N}\}$
2. Project the data points onto the plane by calculating for all data points
$$x_d^{\text{proj}} = \frac{(b^2+c^2)x_d - aby_d - acz_d + a}{a^2+b^2+c^2},$$
$$y_d^{\text{proj}} = \frac{(a^2+c^2)y_d - abx_d - bcz_d + b}{a^2+b^2+c^2} \quad (10)$$
$$z_d^{\text{proj}} = \frac{(b^2+a^2)z_d - cby_d - acx_d + c}{a^2+b^2+c^2}.$$
3. Translate the plane so that it passes through origin and its new equation takes the form $ax_d^{\text{trans}} + by_d^{\text{trans}} + cz_d^{\text{trans}} = 0$ with $x_d^{\text{trans}} = x_d^{\text{proj}}$, $y_d^{\text{trans}} = y_d^{\text{proj}}$ and $z_d^{\text{trans}} = z_d^{\text{proj}} - 1/c$.
4. Rotate this plane such that the z-coordinate becomes zero. i.e.,
$$\begin{bmatrix} x_d^{\text{rot}} \\ y_d^{\text{rot}} \\ z_d^{\text{rot}} \end{bmatrix} = \begin{bmatrix} x_d^{\text{rot}} \\ y_d^{\text{rot}} \\ 0 \end{bmatrix} = \begin{bmatrix} d & 0 & c_x \\ 0 & 1 & 0 \\ -c_x & 0 & d \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_z/d & c_y/d \\ 0 & -c_y/d & c_z/d \end{bmatrix} \begin{bmatrix} x_d^{\text{trans}} \\ y_d^{\text{trans}} \\ z_d^{\text{trans}} \end{bmatrix}. \quad (11)$$
Here $c_x = \frac{a}{\sqrt{a^2+b^2+c^2}}$, $c_y = \frac{b}{\sqrt{a^2+b^2+c^2}}$, $c_z = \frac{c}{\sqrt{a^2+b^2+c^2}}$, $d = \sqrt{c_y^2 + c_z^2}$.
5. Scale the coordinates, so that the maximum parametric coordinate of a point is 1 and minimum is 0. i.e.,
$$u_d = \frac{x_d - x_{\min}}{x_{\max} - x_{\min}}, \qquad w_d = \frac{y_d - y_{\min}}{y_{\max} - y_{\min}} \quad (12)$$
where $x_{\min} = \min\{x_1, \ldots, x_N\}$, $x_{\max} = \max\{x_1, \ldots, x_N\}$ and $y_{\min} = \min\{y_1, \ldots, y_N\}$, $y_{\max} = \max\{y_1, \ldots, y_N\}$.

The overall adaptive algorithm to fit a surface over a set of surface points may be summarized in the following steps.

1. Use the least square scheme to fit a plane P, over the given set of data points $\overline{\mathbf{S}} = \{\overline{s_1}, \overline{s_2}, \ldots, \overline{s_N}\}$
2. Project the points in $\overline{\mathbf{S}}$ onto P to obtain the initial guess of parameters $\mathbf{U} = \{(u_1, w_1), (u_2, w_2), \ldots, (u_N, w_N)\}$. Since a complete polynomial is used, the choice of the orthogonal directions shall only affect the coefficients of $\bar{Q}_{ij}$, 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 $\bar{Q}_{ij}$ in Eq. (5)
4. Calculate average error E_{initial} for all the surface points. The absolute normal error is the smallest distance of a point along the normal to the surface. The absolute error ε_d and the average normal error E are evaluated as
$$\varepsilon_d = \left\{\min(|\lambda|) : \bar{D}_d = \bar{Q}(u, w) + \lambda \frac{\bar{N}(u, w)}{|\bar{N}(u, w)|}\right\} \quad (13a)$$
$$E = \frac{\sum_{d=1}^{N} \varepsilon_d}{N}. \quad (13b)$$
Here, $\bar{N}(u, w)$ is the normal to the surface at parametric coordinates (u, w) and is calculated as cross product of two

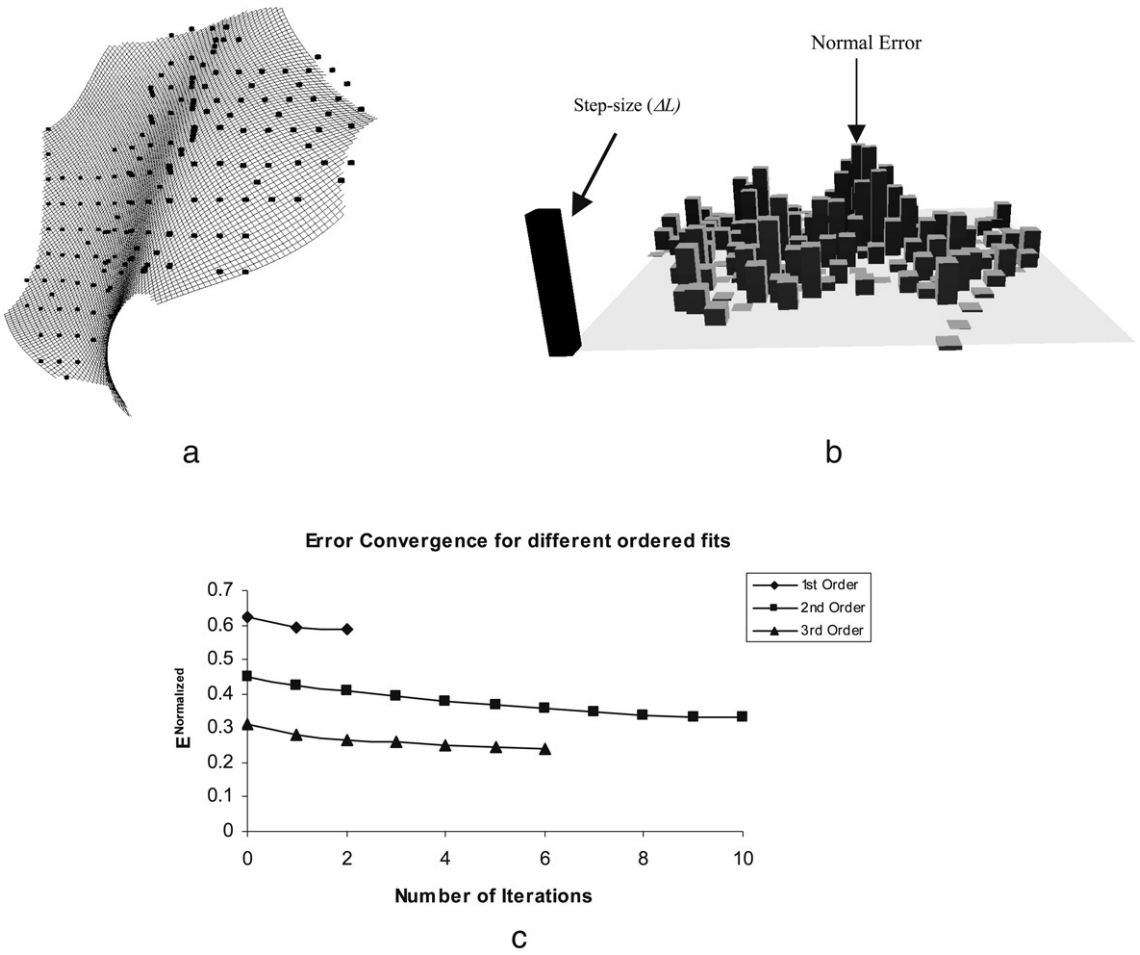

Fig. 10. (a) A representative third-order polynomial surface with a least square fit to a set of data points delineating a grain boundary, (b) point-wise 3D histogram of the point-wise error ε_d, and (c) convergence of average normalized error ($E^{\text{Normalized}}$) with different order interpolation functions.

tangents at that point on the polynomial surface as:

$$\bar{N}(u, w) = \bar{T}_u(u, w) \times \bar{T}_w(u, w) \tag{14}$$

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

5. Use Newton–Raphson iterative method for better estimates of **U** at each point
6. Calculate average error E_{final} for this fit. If $|E_{\text{initial}} - E_{\text{final}}| \leq$ *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 the first-order polynomial and adaptively increases the order till there is convergence with respect to the average normal error of Eq. (13b). Fig. 10 is a demonstration of the effectiveness of this algorithm, where a third-order polynomial surface is adaptively fit to a scattering data points. Each of these data points are then projected onto the surface and the normal distance is calculated. The 3D bar chart is shown in Fig. 10(b), in which each bar corresponds to the projected distance of each point to the surface of the non-planar parametric surface. Thus the height of each bar is equal to the absolute normal error. For comparison, the step-size in the serial-sectioning experiment from which the data is generated, is also shown. The convergence of this error for different order polynomials is shown in Fig. 10(c). The normalized averaged normal error, with the step-size ΔL as the normalizing factor is defined as:

$$E^{\text{normalized}} = \frac{\sum_{d=1}^{N} \frac{\varepsilon_d}{\Delta L}}{N}. \tag{15}$$

3.2.2.2. Non-uniform rational B-spline or NURBS surface-fitting. Non-uniform rational B-spline surfaces or NURBS are the second type of parametric surface that are fit to the set of data points corresponding to the grain boundaries. The NURBS expression for physical coordinates (x, y, z) of a point on the surface in terms of parametric coordinates (u, w) is given as

$$\bar{Q}(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} \overline{B_{i,j}^{h}} N_{i,k}(u) M_{j,l}(w) \tag{16}$$

where the coefficients $\overline{B_{i,j}^{h}}$ correspond to the vertices of a 4D polytope, (k, l) are the order of the surface spline functions and $(n+1)$ and $(m+1)$ are the number of points in each parametric direction. $N_{i,k}$,$M_{j,l}$ are NURBS basis functions defined by the

Cox–deBoor recursion formulae as

$$N_{i,1}(u) = \begin{Bmatrix} 1 & \text{if } x_j \le u \le x_{j+1} \\ 0 & \text{otherwise} \end{Bmatrix}, \quad \text{and}$$

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \quad (17)$$

where, $[X] = [x_1, x_2, \ldots, x_{n+k+1}]$ is called the knot-vector with $x_i \le x_{i+1}$, and $u \in [x_1, x_{n+k+1}]$. The NURBS routine in NX3 [31] to fit a NURBS surface for a point cloud is used to obtain a "smooth" surface for the set of grain boundary points. The same iterative technique mentioned in Section 3.2.2.1 is implemented for least square fit of the NURBS function. Both the polynomial and NURBS-based surfaces are checked for self-intersection and lowest average normalized error, and the one with the least error and which does not self intersect is retained. Fig. 11 is a histogram of the percentage of surfaces with a given value of $E^{\text{normalized}}$ for those that are chosen from the two types. Almost all the surfaces have low errors with very few of them having errors greater than the step-size itself. These errors can be attributed to the fact that the only lower-order functions (third order or lower) have been considered in this example for both the polynomial and NURBS surfaces. Higher-order surfaces can be self-intersecting or may show unusually sharp variations and are avoided. The larger errors may also due to locally incorrect experimental data, such as a spike left out by a carbide particles.

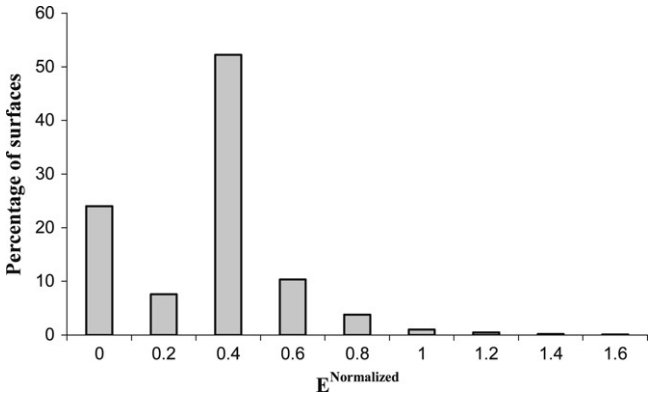

Fig. 11. Percentage of surfaces as a function of the average error $E^{\text{normalized}}$.

3.2.3. Cell decomposition

Spatial partitioning algorithms, such as quad-tree, octree or BSP trees, have been extensively used in geometric modeling for solid body representation. All these methods involve the partitioning of a 'universe' solid into a number of cells, each of which are assigned to the solid based on their containment. The merging of these cells produces a representation of the desired solid. These methods differ in the choice of the universe cell and the choice of partitioning surfaces. One of the most general methods in this class of methods is the cell-decomposition method. In this method, an arbitrary-shaped solid is defined as the universe cell and arbitrary-shaped surfaces are considered as the partitioning surface. As illustrated in Fig. 12, this space partitioning technique is used to reconstruct individual grains from the grain surfaces constructed in the previous section. The following sequence of steps is pursued in this reconstruction.

1. *Construct the 'Universe Cell' (W)* to wrap the set of data points contained in a grain G, with an offset of at least twice the step-size i.e. $2 \times \Delta L$. In NX3, W is a planar offset solid containing the data points. Ensure that W should remain within the bounds of the actual specimen domain.
2. *Select partitioning surfaces (S)* as the yet unused surface consisting of the largest number of surface points belonging to this grain for the next partition.
3. *Perform cell partitioning* by selecting all the unassigned cells, **C** ($\mathbf{C} = \{W\}$ for the initial cell), which contain at least one of the points interpolated by S. Extrapolate S to partition **C** into subcells **CP**.

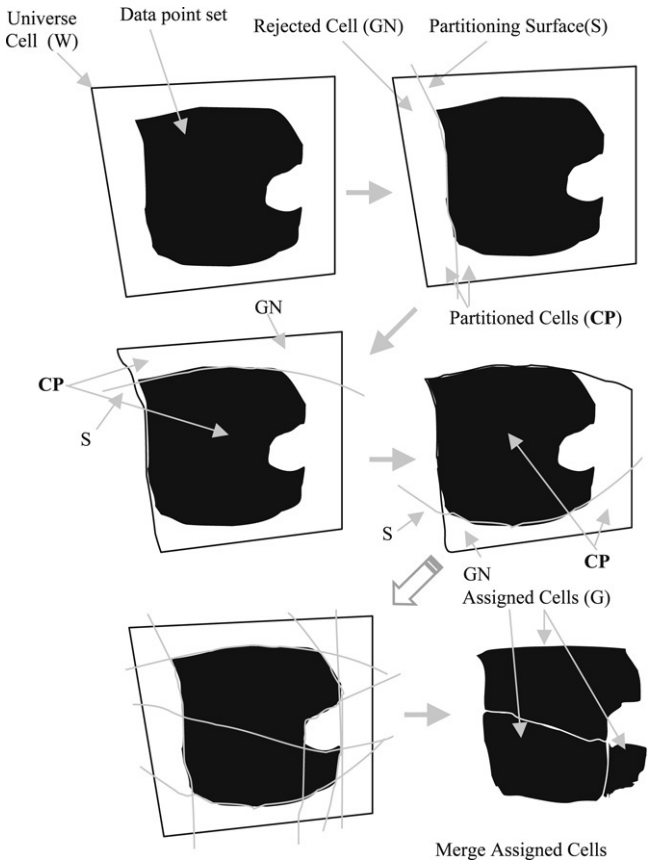

Fig. 12. Schematic representation of the cell-decomposition methodology corresponding to the steps in Section 3.2.3. The reconstructed solid grain is the union of the assigned cells after partitioning of the universe cell.

4. *Assign cells* for each $CP_i \in \mathbf{CP}$ by performing containment check inside CP_i for all the points belonging to G and all its neighbors GN. If more than 90% of the contained points in CP_i belong to G then assign it to G. If less than 10% belong to G, assign it as GN. Otherwise leave it unassigned. Repeat steps 2–4 till all surfaces are exhausted. Leftover unassigned cells are assigned to G or GN based on maximum containment of points of G and GN inside it.
5. *Merge the cells* assigned as G for the final grain representation. Delete all other cells.

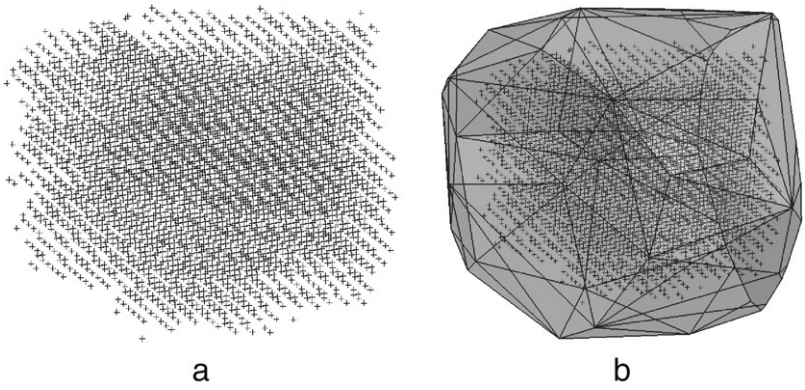

Fig. 13. (a) Volumetric data points belonging to voxels constituting a grain, (b) Universe call (semi-transparent) overlaid on the grain data points.

The selection of the size of the universe cell 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. 13 shows the wrap solid as universe cell for one such grain. In step 3, the selection of only those cells that have points belonging to surface contained in them leads to a more local reconstruction. 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 step retains the local nature of the reconstruction. Improvements in both accuracy and also speed are achieved as a result. Fig. 14(a) shows the reconstructed grain for the data points in Fig. 13(a). The surface points of this grain are also shown in the figure to demonstrate the closeness of the reconstructed solid to the voxelized representation.

3.3. Compatibility requirements in a polycrystalline aggregate

Individually constructed grains by the hierarchical solid reconstruction process described in Section 3.2, when assembled in a polycrystalline aggregate, may incur gaps and overlaps between them. This is because the individual grain surfaces are created in isolation from the other grains in the ensemble. Regions occupied by more than one grain (overlaps) and regions where there are no grains present (gaps) are non-physical and should be removed. Additionally, geometric artifacts generated in the process of reconstruction and gap and overlap removal are removed through a subsequent defeaturing process in Section 3.4.

3.3.1. Overlap removal

A schematic of the grain overlap removal algorithm is presented in Fig. 15. Let $\mathbf{R} = [R_1, R_2, \ldots, R_M]$ be the set of all reconstructed solid grains and $\bar{\mathbf{V}} = \{\overline{\mathbf{V}_1}, \overline{\mathbf{V}_2}, \ldots, \overline{\mathbf{V}_M}\}$ be the set of the voxel points belonging to each grain. To avoid confusion, the following convention is used. The union and intersection of two sets is represented by the set operators $\cup$ and $\cap$ respectively, while the union and intersections of two solids are represented by the Boolean nomenclature ***Union***(S_1, S_2) and ***Intersect***(S_1, S_2) etc. The overlap algorithm is outlined next.

1. Set $i = 1$ and $j = 2$.

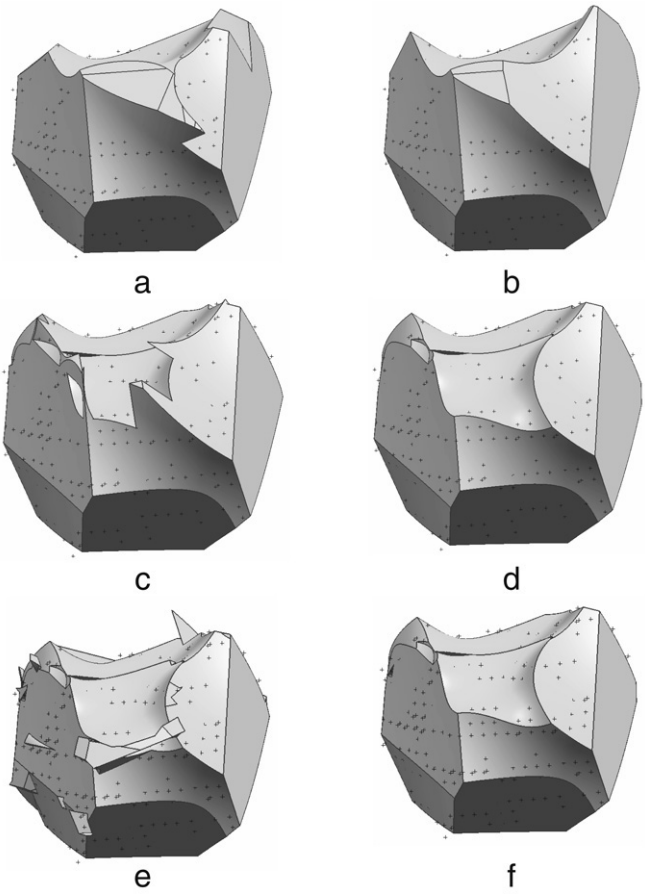

Fig. 14. A single grain reconstruction showing discrete surface data points; (a) after cell decomposition, (b) after cell decomposition with defeaturing, (c) after overlap removal, (d) after overlap removal with defeaturing, (e) after gap removal, and (f) after gap removal with defeaturing.

2. Find $\mathbf{I} = \textbf{\textit{Intersect}}(R_i, R_j)$ 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 $\overline{\mathbf{D^{ij}}} = \{\overline{D_1^{ij}}, \overline{D_2^{ij}}, \ldots, \overline{D_{N_{ij}}^{ij}}\}$ belonging to the interface between the grains i and j, perform a containment test to find out points $\overline{\mathbf{P^k}} = \{\overline{P_1^k}, \overline{P_2^k}, \ldots, \overline{P_A^k}\}$ that are contained inside I_k.
4. Fit surface a S^k to the set of points $\overline{\mathbf{P^k}} = \{\overline{P_1^k}, \overline{P_2^k}, \ldots, \overline{P_A^k}\}$.

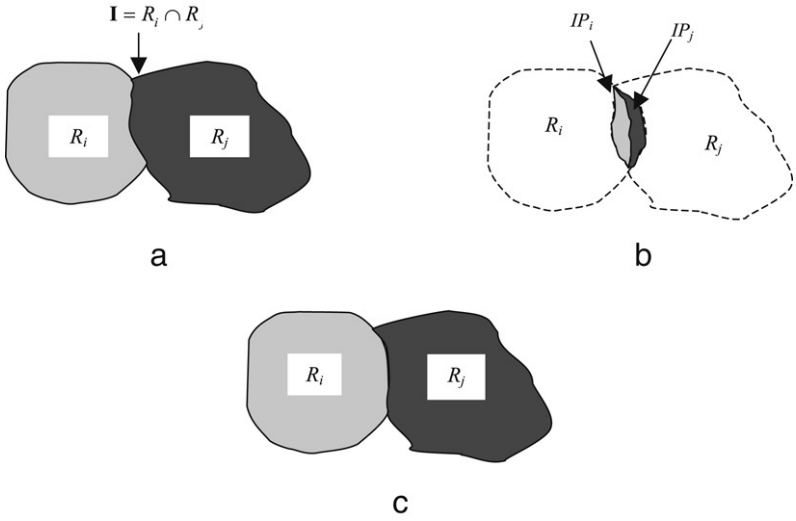

Fig. 15. Schematic representation of the overlap removal process: (a) two overlapping grains with identification of overlapped region **I** between R_i and R_j, (b) partitioning of **I** into IP_i and IP_j using a fitted surface, (c) assignment of IP_i and IP_j to R_i and R_j respectively and creation of the resulting non-intersecting grains.

5. Partition I_k with the surface S^k into the solids $\mathbf{IP} = \left\{IP_1^k, IP_2^k, \ldots, IP_B^k\right\}$.
6. For each $IP_l^k \in \mathbf{IP}$, count the number of points 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 performing $R_i = \textbf{\textit{Union}}(R_i, IP_l^k)$, otherwise perform $R_j = \textbf{\textit{Union}}(R_j, IP_l^k)$.
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.

By fitting a separate surface in step 4, only those points that are local to the overlapping region are included in the partitioning process. Also the accuracy of the interface with respect to the local microstructure is better, since the effect of interface points from other regions is not considered. For the overlapping regions, small enough not to contain any surface points, no partitioning is done. Fig. 14(c) shows the effect on the geometry of a representative grain after the overlaps have been removed.

3.3.2. Gap removal

Fig. 16 shows a schematic of the two basic Boolean operations required for the removal of gaps between contiguous grains, viz. (i) identification of unassigned regions or gaps, and (ii) assigning them to appropriate solid grains. Let $\mathbf{G} = \{G_c\}$ correspond to the current set of gaps in the set of reconstructed grains $\mathbf{R}^o = \left\{R_1^o, R_2^o \ldots, R_M^o\right\}$ that have already undergone overlap removal. G_c is a solid representing the metallic specimen. The following steps are required to remove gaps from $\mathbf{R}^o$, with starting values $i = j = 1$.

1. For $R_i^o \in \mathbf{R}^o$ set $j = 1$.
2. For $G_j \in \mathbf{G}$ determine the set $\mathbf{\Lambda} = \textit{Subtract}(G_j, R_i^o)$.
3. If $\mathbf{\Lambda} = \{\varnothing\}$ then $j = j + 1$ and go to step 2. If $i = M$ go to step 4, otherwise evaluate $\mathbf{G} = \mathbf{G} \cup \mathbf{\Lambda}$, set $i = i + 1$ and go to step 1.
4. Create a set $\mathbf{W}^o = \left\{W_1^o, W_2^o, \ldots, W_M^o\right\}$ as the wrap surfaces of each grain for all grains in the ensemble.
5. Since each of the gaps $G_j \in G$ may be bigger than what is assigned to an individual grain, they are partitioned into smaller gaps $P = \{P_1, P_2, \ldots, P_k\}$ with all of the wrap surfaces $\mathbf{W}^o$.
6. For each $P_i \in \mathbf{P}$, identify the reconstructed grain $R_g^o \in \mathbf{R}^o$ with which it shares the maximum surface area and set $R_g^o = \textbf{\textit{Unite}}(R_g^o, P_i)$.

From step 2, it is obvious that the process of 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 much larger than what can be assigned to an individual grain. To account for such cases an additional step of subdividing larger gaps is introduced in step 5. Choosing wrap surfaces in this step ensures that, when these gaps are assigned to appropriate grains, no grain exceeds its wrap boundary. Many of the gaps are so small that they may not contain even a single voxel inside them. Consequently, the choice of the grain to which the gap should be assigned is not straightforward. Instead of assigning the gap arbitrarily to its neighbors, a neighbor is chosen which shares maximum common surface area with this gap. The solids thus generated are relatively smooth and are easier to mesh. Fig. 14(e) shows the effect of gap removal process on a grain. While handling the exceptions and errors during subtraction, care must be taken to produce no new intersections. Otherwise, the entire overlap removal procedure may have to be repeated.

3.4. Defeaturing spurious artifacts of the reconstructed grains

The entire process of grain reconstruction and compatibility enforcement can lead to certain artifacts in the microstructure that are non-physical and may cause problems at the analysis stage. For example, a small face such as a sliver on the grain boundary can cause problems with respect to finite element mesh generation as well as give rise to spurious

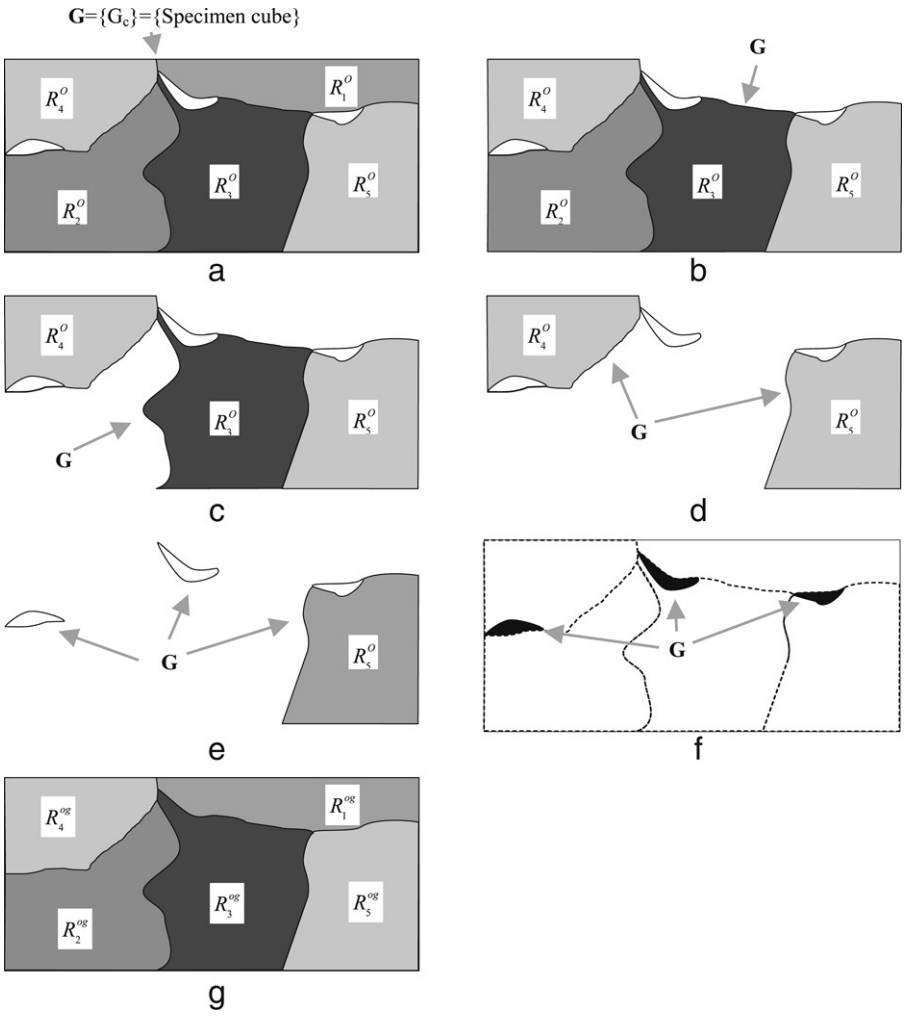

Fig. 16. Steps involved in identification and removal of gaps: (a) reconstructed microstructure with overlaps removed, (b)–(e) process of subtraction of grains R_i^o from **G**, (f) gaps identified in the previous iterations, and (g) microstructure after assigning gaps to appropriate grains. Steps of partitioning gaps are skipped in this illustration.

high local stresses, thus affecting the quality of solutions. Defeaturing is the process of removing these artifacts from the reconstructed grains during their creation process prior to their implementation in a finite element mesh and model. Defeaturing routines are introduced at three places in the program, viz. (i) before overlap removal, (ii) before gap removal and (iii) before mesh generation.

The defeaturing at different stages may have different compatibility requirements. For example, it can allow new gaps or new overlaps to be formed prior to the overlap removal stage. However, it should neither introduce new gaps nor overlaps after the gap removal stage. Thus, the same procedure for removing the artifacts is implemented in a single routine in NX3, with the flexibility to adapt specific compatibility requirements.

The first step in the defeaturing process is the identification of artifacts. Although various types of non-physical features are possible, the most common is the occurrence 'tiny faces' that may be defined in NX3 as (see [31]):

1. Faces for which the surface area is less than a specified tolerance.
2. Faces surface area to perimeter ratio is less than a specified tolerance.

While a looser tolerance of $2 \times \Delta L$ may be used for larger grains, the tolerance for smaller grains is obtained by linearly scaling from 0 for a grain with zero volume to $2 \times \Delta L$ for a moderate grain size.

Once the 'tiny faces' are identified, the defeaturing process is carried out in two phases. In the first phase, cluster of neighboring tiny faces are removed simultaneously using the 'Simplify Body' feature in NX3. If this operation fails, selecting the faces for not removal corresponding to the failing wound has been found to be useful. In the second phase, the leftover individual faces in the simplified geometry are removed by using the modeling feature 'Replace Face' in NX3. The face to be removed, as well as the neighboring tool face are specified in this operation. Unlike 'Simplify Body' function, specifying different tool faces may yield different solids. The program exploits the generation of multiple defeatured geometry for the same solid by performing all possible 'Replace Face' operations and choosing the 'best solid' with the least number of faces.

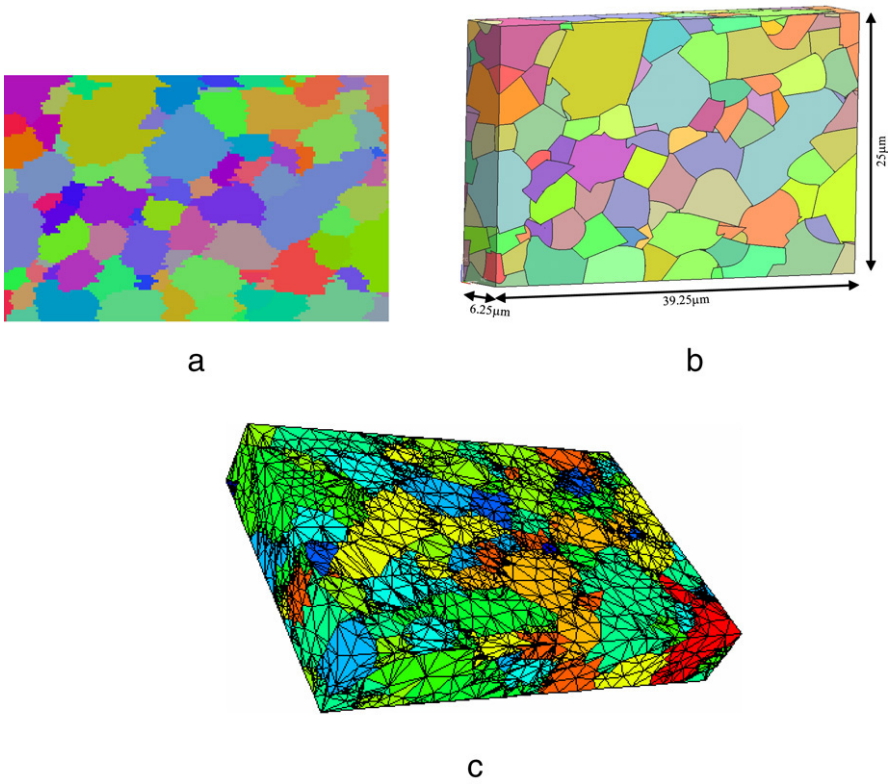

Fig. 17. Comparison of (a) voxel-based grain section representation of the FIB–SEM data, (b) CAD-based 3D reconstruction of the polycrystalline microstructure, (c) meshed microstructure using an adaptive meshing process discussed in [32].

The next step restores compatibility, depending on the constraints under which the defeaturing is being performed. For not letting any new overlap to be introduced, the original solid is first subtracted from the defeatured solid. Any additional solid in the new defeatured solid is 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 solid thus created, is then assigned to the neighboring grains. The final step in the defeaturing process is to quantify the effect of all the operations on the overall 'quality' of the microstructure representation. Defeaturing a grain may give it a very simple representation. On the other hand, its neighbors may become much worse after performing the compatibility restoration operations. To find out the effect of the defeaturing process on the overall quality, the total number of faces of a grain and its neighbors are compared before and after the defeaturing process. If the number has reduced then the defeaturing operation is assumed to be successful, otherwise the older state is restored and the previous step is termed as unsuccessful. The entire defeaturing algorithm is summarized below.

Let R_g^{og} be a reconstructed grain to be defeatured with a wrap solid W_g and with neighboring grains $\mathbf{N}_g = \{R_\alpha^{og}, R_\beta^{og}, \ldots, R_\delta^{og}\}$. Let $NF(S)NF(\mathbf{A})$ be functions that return the number of faces in the solid S and the set of solids in $\mathbf{A}$ respectively. The following steps are performed for the defeaturing operation.

1. Mark faces that should not be removed in this operation as $\mathbf{BF} = \{\varnothing\}$
2. Identify all the tiny faces in R_g^{og} as $\mathbf{TF} = \{TF_1, TF_2, \ldots, TF_L\}$. Remove all elements in **TF** that also belong in **BF**. If $\mathbf{TF} = \{\varnothing\}$ then exit.
3. For each face TF_i, define a neighboring face as one that shares at least one edge with it. Club together all the tiny faces that are neighbors of each other, into a set **C** consisting of the subsets $\mathbf{C} = \{C^1, C^2, \ldots, C^M\}$. The subsets should satisfy the condition $n(C^i) \geq n(C^{i+1})$, where $n(C^i)$ is the number of faces in C^i.
4. Remove the set of faces C^1 from R_g^{og} using the 'Simplify Body' function in NX3 to obtain a new solid grain $\mathbf{D_g} = \{D_g^1\}$. If this process is successful then go to step 7; otherwise proceed to step 5.
5. List edges $\mathbf{E} = \{E_1, E_2, \ldots, E_P\}$ that are causing failure of the previous operation. Remove all faces from C^1 that have at least one edge common with elements of **E**. Add all these faces to the set **BF**.
6. If $C^1 = \{\varnothing\}$ then go to 2; otherwise go to 3.
7. Set $D_g^i = \textbf{\textit{Intersect}}(D_g^i, W_g)$ for all $D_g^i \in \mathbf{D_g}$. When performing this intersection chose only the largest of the resulting solids.
8. For each $D_g^i \in \mathbf{D_g}$, check if the number of faces in $NF(D_g^i) \geq NF(R_g)$. If this is true, then set $\mathbf{D_g} = \mathbf{D_g} - \{D_g^i\}$.
9. If $\mathbf{D_g} = \{\varnothing\}$ then set $\mathbf{BF} = \mathbf{BF} \cup \mathbf{C^1}$, $\mathbf{C^1} = \{\varnothing\}$ and go to 2.
10. If the current mode of defeaturing is executed after gap removal, then for each $D_g^i \in \mathbf{D_g}$, obtain the set of all

solids resulting from subtraction of the defeatured grain from the reconstructed grain, and unite these solids to their copies of the neighboring grains $\mathbf{N_g^i}$. If the current mode of defeaturing is executed after overlap or gap removal then, for each $D_g^i \in \mathbf{D_g}$ obtain the set of all solids resulting from subtraction of reconstructed grain from the defeatured grain, and subtract these solids from their copies of the neighboring grains $\mathbf{N_g^i}$.

11. Sort elements $D_g^i \in \mathbf{D_g}$ in the ascending order of $nf_g^i = (NF(D_g^i) + NF(N_g^i))$.
12. Calculate the number of faces of the reconstructed grain and its neighbors as $nf_g = (NF(R_g) + NF(N_g))$. If $nf_g \leq nf_g^1$ then delete all $D_g^i \in \mathbf{D_g}$ and their corresponding $\mathbf{N_g^i}$. Also set $\mathbf{BF} = \mathbf{BF} \cup C^1$, $C^1 = \{\varnothing\}$. If this condition is not true then replace R_g^{og} with D_g^1, $\mathbf{N_g}$ with $\mathbf{N_g^1}$. Finally go back to step 2.

After execution of the defeaturing algorithm, a slightly altered algorithm is executed again. The differences are listed below.

1. Do not club together any face in step 3, define $C^i = \{TF_i\}$
2. Replace step 4 with the following operations. Remove the set of faces C^1 from R_g^{og} by using the 'Replace Face' function in NX3 to obtain all possible new solids $\mathbf{D_g} = \{D_g^1, D_g^2, \ldots, D_g^A\}$. If process is successful then proceed to step 7; otherwise set $\mathbf{BF} = \mathbf{BF} \cup C^1$, $C^1 = \{\varnothing\}$ and go to step 2.
3. Steps 5 and 6 are not required in this sweep.

Fig. 14(b), (d) and (f) show the defeatured grain after reconstruction, overlap removal and gap removal respectively.

4. Numerical results and algorithm validation

The CAD-based methodology described in Section 3 is now tested for reconstructing the microstructure of a fine-grained polycrystalline nickel-base superalloy, IN100. Fig. 17 (a) shows a section of the microstructure in a voxel-based representation using the experimental data after alignment. Fig. 17(b) shows a 3D representation of the reconstructed grain microstructure with the colors corresponding to a given orientation. Visual comparison of the two figures shows good agreement between the sections of the experimental and simulated microstructures. The high frequency stepped patterns in the voxel representation are smoothened in the simulated representation with lower-order surfaces. This simplification is desirable for convenient finite element mesh generation and analysis. The high frequency undulations necessitate high resolution mesh at regions where the solutions may not require high mesh density. A better control of the local mesh density with an optimal number of nodes and elements are possible because of this smoothened representation. Low frequency undulations, which are responsible for critical local variables, e.g. high stress gradients, are retained in the simulated microstructures. The overall reconstruction method is comprehensive, in that it considers accuracy from an input point of accurate (OIM data) and efficiency from an output point of view (finite element analysis). Fig. 17(c) shows the results of a meshed microstructure as detailed in [32].

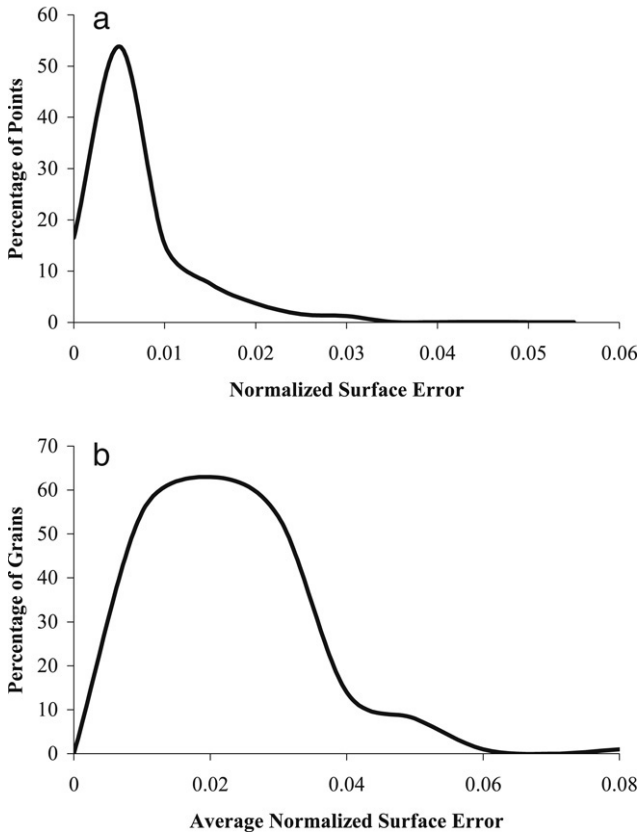

Fig. 18. Distribution of (a) the normalized surface error ε_i^g for a simulated grain, and (b) average normalized surface error $\varepsilon_{\text{average}}$ for all grains of Fig. 17.

For validation, each surface point $\overline{s_i}$ of a grain g, is checked whether it is contained within the reconstructed grain geometry (R_g). If the point lies inside the grain geometry, then the error is zero, while for the points outside the grain it is the projected distance to the grain surfaces normalized with the grain diameter for the grain i.e.,

$$\varepsilon_i^g = \left\{ \begin{array}{cc} 0 & \text{if } \overline{s_i} \subset R_g \\ \dfrac{\text{projected distance } \left(R_g, \overline{s_i}\right)}{d_g} & \text{otherwise} \end{array} \right\} \tag{18}$$

where, the average grain diameter is calculated from the volume of each grain as $d_g = 2 \times \sqrt[3]{\frac{3V_g}{4\Pi}}$. Here V_g is the volume of the voxelized FIB/SEM grain. Fig. 18(a) shows a plot of the normalized surface error for the simulated grain of Fig. 14. Other than a few outlying points still within the limits of experimental error, a majority of the points are within 50% of the experimental step-size. Good results are also seen for the average value of this error for all grains in the microstructure in Fig. 18(b). Table 1 shows the best and worst normalized surface errors. Owing to the better resolution of the grain geometry, it can be clearly observed from the table that the larger grains tend to have lower errors, then the smaller ones. This is an important result for this work, as it shows the importance of high resolution experimental data for the process of reconstruction.

Table 1
Average normalized surface error for the grains in the 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

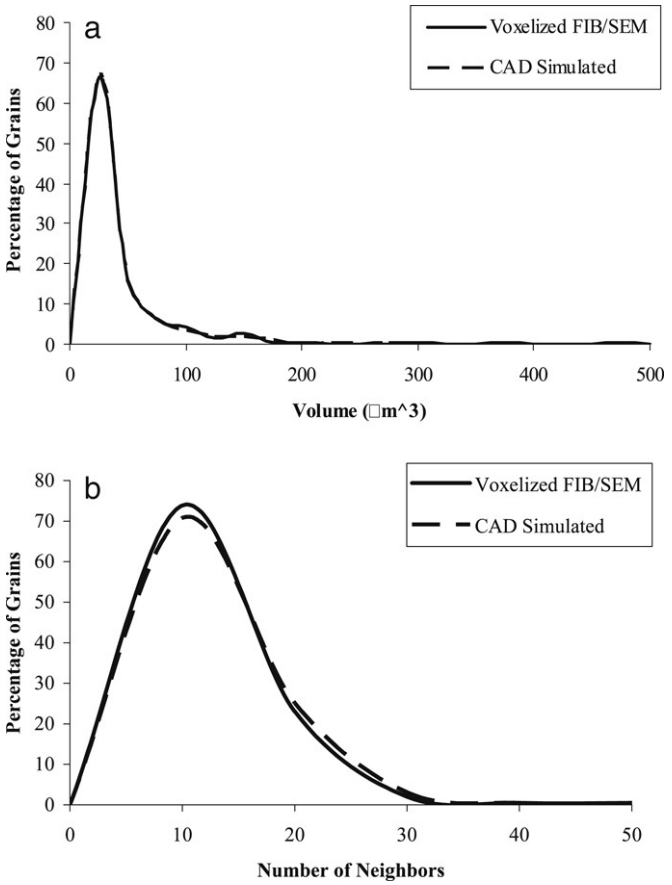

Fig. 19. Comparison of the distribution of (a) grain volume and (b) number of neighbors for the experimental voxel-based microstructure with that for the simulated microstructure.

Figs. 19 and 20 compare the distribution of the volume, number of neighbors of each grain and moment of inertia for the experimental voxel-based microstructure with the simulated microstructure. Excellent match is observed for these parameters. However the distribution of the surface areas in Fig. 21 shows a significant difference between the two microstructures. As expected, the reconstructed grain's smooth surface area is much smaller than that for the stair-case stepped surface of the voxel-based representation.

Over 125,000 faces in voxel-based representation are replaced by about 4000 faces in the CAD simulated microstructure. A significantly lower number of surface nodes are expected in the finite element mesh for the simulated microstructure. Also the simulated microstructure enjoys the flexibility with respect to choice of elements with density control.

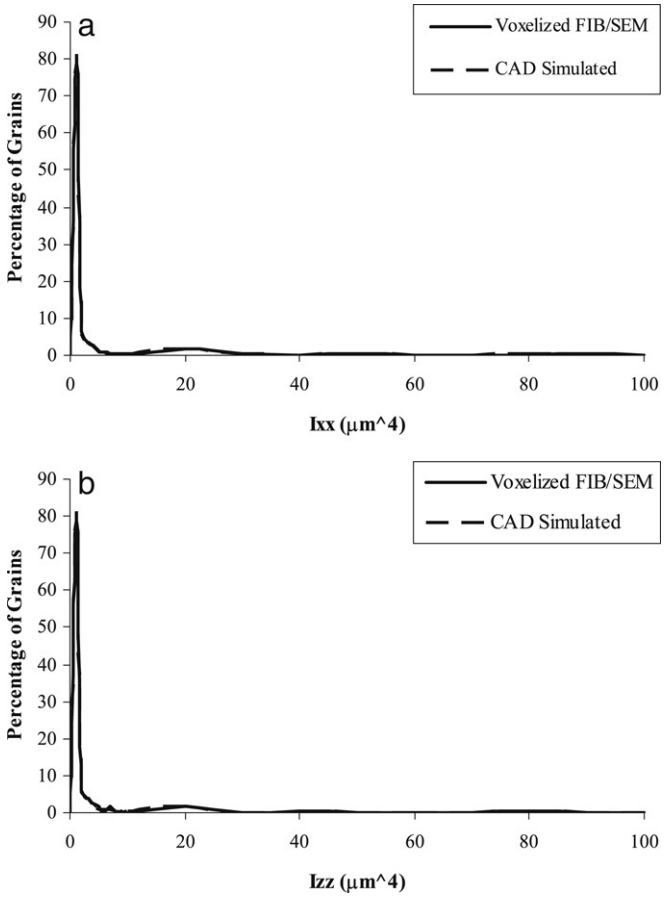

Fig. 20. Comparison of moment of inertia between voxel-based microstructure and simulated microstructure.

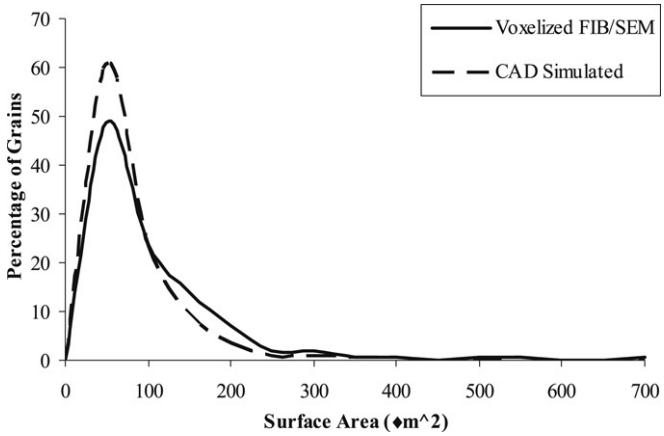

Fig. 21. Comparison of the distribution of surface area for the experimental voxel-based microstructure with that for the simulated microstructure.

5. Conclusion

This paper is aimed at the development of a robust and comprehensive CAD-based methodology for simulating 3D microstructures of polycrystalline metals using crystallographic input data on sections created by a focused ion beam (FIB)–scanning electron microscopy (SEM) system. The 3D dual-beam FIB–SEM is very effective for serial-sectioning micron/submicron scale metallic specimens and subsequently

obtaining crystallographic orientation maps for each section. Orientation maps from each of these sections are then stacked to yield voxelized representation of the microstructure. The method developed in this paper is able to reconstruct consistent polycrystalline microstructures from this data, with control on the resolution and morphological details necessary for meaningful computational analysis in microstructure-property estimation.

The microstructure simulation methodology is based on a hierarchical geometrical representation using primitives used in CAD modeling. With the help of crystallographic orientation information, grains are first segmented as a collection of voxels. Data cleanup is then performed to eliminate any un-indexed or incorrectly indexed measurements. Points on the interface belonging to each pair of neighboring grains are identified from misorientation data in contiguous voxels. Parametric surface segments with least squares are fit to the interface data are then constructed adaptively using polynomial and NURBS functions. Following this, the cell-decomposition algorithm creates volumetric rendering of the grains as a 3D domain enclosed by the surfaces. A geometric defeaturing operation is performed to remove any unwanted kinks from the simulated microstructure. Since the grains are individually created in isolation, their assembly in polycrystalline aggregates may cause problems with respect to inter-grain compatibility. For example there may be residual overlaps and gaps among the assembled grains. Boolean operations are executed for exhaustive gap and overlap removal to create clean 'water-tight' grain boundaries. This is again followed by the defeaturing routine to avoid the creation of any spurious features as a result. The overall polycrystalline microstructure simulation algorithm is validated using various error criteria and measures, for an extracted microstructure of a nickel superalloy. The metrics include local and global errors in projected distances to the grain interfaces, distribution of volume, surface area and number of neighbors etc. All of these tests attest to the excellent microstructures that can be simulated by this method. They provide a benchmark for any stereological calculations that can be subsequently conducted.

Microstructures simulated by this method can find direct application in material modeling for microstructure-property relations. These include statistical quantification in terms of distributions functions and crystal plasticity-based finite element simulations for predicting microstructural variables and response. A significant strength of this method is that it is entirely possible to monitor and control the resolution of the simulated microstructure and retain one that is 'optimal' with respect to the accuracy and efficiency needed for modeling. Hence it can go a long way in the effective analysis and design of microstructures. Details on the finite element mesh generation for these microstructures and their analyses are currently under development will be reported in a future paper.

Acknowledgements

The authors from OSU acknowledge the support of Air Force Office of Scientific with Grant # FA9550-05-1-0067 (Program Director: Dr. James Tiley) and the support of Office of Naval Research with Grant # N00014-05-1-0504 (Program Director: Dr. Julie Christodolou). This support is gratefully acknowledged. Computer support by the Ohio Supercomputer Center through grant # PAS813-2 is also acknowledged.

References

[1] Harren SV, Asaro RJ. Nonuniform deformations in polycrystals and aspects of the validity of the Taylor model. J Mech Phys Solids 1989; 37(2):91–232.
[2] Lemonds J, Asaro RJ, Needleman A. A numerical study of localized deformation in bicrystals. Mech Mater 1985;4:363–73.
[3] Venkatramani G, Deka D, Ghosh S. Crystal plasticity based FE model for understanding microstructural effects on creep and dwell fatigue in Ti-6242. ASME J Engng Mater Tech 2006;128(3):356–65.
[4] Deka D, Joseph DS, Ghosh S, Mills MJ. Crystal plasticity modeling of deformation and creep in polycrystalline Ti-6242. Metal Mater Trans A 2006;17A(5):1371–88.
[5] Sinha S, Ghosh S. Modeling cyclic ratcheting based fatigue life of HSLA steels using crystal plasticity FEM simulations and experiments. Int J Fatigue 2006;28:1690–704.
[6] Kalidindi SR, Bronkhorst CA, Anand L. Crystallographic texture evolution in bulk deformation processing of fcc metals. J Mech Phys Solids 1992;40:537–69.
[7] Balasubramanian S, Anand L. Elasto-viscoplastic constitutive equations for polycrystalline fcc materials at low homologous temperatures. J Mech Phys Solids 2002;50:101–26.
[8] Turkmen HS, Loge RE, Dawson PR, Miller M. On the mechanical behavior of AA 7075-T6 during cyclic loading. Int J Fatigue 2003;25: 267–81.
[9] Dawson P, Miller M, Han TS, Bernier J. An accelerated methodology for the evaluation of critical properties of polyphase alloys. Metallurgical and Mater Trans A 2005;36A:1627.
[10] Bennett VP, McDowell DL. Polycrystal orientation distribution effects on microslip in high cycle fatigue. Int J Fatigue 2003;25:27–39.
[11] Korbel , Martin P. Microscopic versus macroscopic aspect of shear band deformation. Acta Metall Mater 1990;38:1369–80.
[12] Burke SK, Cousland SMcK, Scala CM. Mater Forum 1994;18:85–109.
[13] King RG, Delaney PM. Mater Forum 1994;18:21–9.
[14] Berman M, Bischof LM, Breen EJ, Peden GM. Image Analysis. Mater Forum 1994;18:1–19.
[15] Buffiere JY, Maire E, Verdu C, Cloetens P, Pateyron M, Peix G, et al. Damage assessment in an Al/SiC composite during monotonic tensile tests using synchrotron X-ray microtomography. Mater Sci Engng A 1997;234–36:633–5.
[16] Yancey RN, Baaklini GY. Proc. ASME inter. Gas turbine aeroengines congress. 1993.
[17] Saylor DM, Fridy J, El-Dasher BS, Jung KY, Rollett AD. Statistically representative three-dimensional microstructures based on orthogonal observation sections. Metal Mater Trans A 2004;35A:1969–79.
[18] Groeber M, Uchic M, Dimiduk D, Bhandari Y, Ghosh S. A framework for automated 3D microstructural analysis & representation. In: Conference proceedings symposium 4, multi-scale materials modeling. 2006.
[19] Lewis AC, Geltmacher AB. Image-based modeling of the response of experimental 3D microstructures to mechanical loading. Scripta Mater 2006;55:81–5.
[20] Lorensen WE, Cline HE. Marching cubes: A high resolution 3d surface construction algorithm. Comput Graphics 1987;21(4):163–9.
[21] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques. 1998. p. 415–21.
[22] Dey TK, Giesen J. Detecting undersampling in surface reconstruction. In: Proceedings of 17th ACM symposium on computational geometry. 2001. p. 257–63.

[23] Geiger B. Three-dimensional modeling of human organs and its applications to diagnosis and surgical planning. Technical report, 2105, France: INRIA; 1993.
[24] Barequet G, Sharir M. Piecewise-linear interpolation between polygonal slices. In: Annual symposium on computational geometry. 1994. p. 93–102.
[25] Bajaj CL, Coyle EJ, Lin KN. Arbitrary topology shape reconstruction from planar cross sections. Graphical Models Image Process 1996;58(6): 524–43.
[26] Groeber MA, Haley BK, Uchic MD, Dimiduk DM, Ghosh S. 3D reconstruction and characterization of polycrystalline microstructure using a FIB–SEM system. Material Characterization 2006;57(4–5): 259–73.
[27] Kocks UF, Tomé CN, Wenk HR. Texture and anisotropy: Preferred orientations in polycrystals and their effect on materials properties. Cambridge: Cambridge University Press; 1998.
[28] Xie CL, Ghosh S, Groeber M. Modeling cyclic deformation of HSLA steels using crystal plasticity. J Engng Mater Tech 2004;126:339–52.
[29] Piegl L. On NURBS — A survey. IEEE Comput Graphics Appl 1991; 11(1):55–71.
[30] Farin G. From conics to NURBS — A tutorial and survey. IEEE Comput Graphics Appl 1992;12(5):78–86.
[31] Unigraphics NX3.0.4.2 . User Manual. UGS corporation; 2004.
[32] Bhandari Y, Sarkar S, Groeber M, Uchic M, Dimiduk D, Ghosh S. 3D polycrystalline microstructure reconstruction from FIB generated serial sections for FE Analysis. Comput Mater Sci 2007;41:222–35.