IOP Publishing Modelling and Simulation in Materials Science and Engineering

Modelling Simul. Mater. Sci. Eng. **22** (2014) 025017 (21pp) doi:10.1088/0965-0393/22/2/025017

Three-dimensional digital approximations of grain boundary networks in polycrystals

S-B Lee[1,3], G S Rohrer[2] and A D Rollett[2]

[1] School of Mechanical and Advanced Materials Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea
[2] Department of Materials Science and Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

E-mail: sukbinlee@unist.ac.kr, rohrer@cmu.edu and rollett@andrew.cmu.edu

Received 16 July 2013, revised 29 November 2013
Accepted for publication 10 December 2013
Published 31 January 2014

Abstract

In this work, we offer a set of algorithms that convert a voxellated image to a conformal surface mesh that is targeted for polycrystalline materials containing grains with a wide range of sizes and complex shapes. More specifically, we propose a simple but effective algorithm for approximating the grain boundary networks that are implicit in three-dimensional digital images of polycrystals. The algorithm segments a three-dimensional digital image of a polycrystalline microstructure and then smoothes an interpolated conformal surface mesh of the grain boundary network while maintaining certain characteristic features of the microstructure. It is found that the proposed algorithm successfully approximates the grain boundary network based only on the digital, voxellated images of the polycrystal. Simulated microstructures are used to verify that the resulting mesh qualitatively and quantitatively approximates the true structure, in terms of the displacement of the nodes, the grain volume change and the dihedral angle distribution along triple junctions after smoothing. The effect of the use of the cubic grid for mapping digital microstructures on the grain boundary approximation is also discussed.

Keywords: 3D polycrystals, grain boundaries, surface mesh, segmentation and smoothing

(Some figures may appear in colour only in the online journal)

1. Introduction

In ideal, fully dense, single-phase polycrystals, grains form a complex network of conformal grain boundaries and junctions without leaving any empty space. Many theories on the

[3] Author to whom any correspondence should be addressed.

morphology and evolution of the polycrystalline structures are based on this premise [1–6]. In general, the grain boundaries and junctions in polycrystals have a range of smoothly changing curvatures, depending on the relative grain sizes of the neighboring grains, the number of neighboring grains and the nature of the grain boundaries (misorientations and energies). In contrast, the majority of the microstructural images from simulations and experiments are digitized, resulting in various kinds of aliased, stair-stepped, discrete grain boundaries and junctions that are influenced by the type of the regular grid used to construct the image. With rare exceptions, aliasing of the boundaries is attributed to inadequate resolution in the characterization. Therefore, it is essential to extract physically plausible geometric features from the digitized images before computing such quantities as area, orientation, curvature, volume and connectivity.

In fact, the realistic approximation and characterization of grains and grain boundary networks in polycrystals is of significant interest because many of the properties and behaviors of polycrystals are functions of their microstructural parameters. More interestingly, many materials science phenomena related to polycrystals are three-dimensional in nature. However, microstructures are traditionally characterized by observing two-dimensional plane sections. Even though stereological methods [7] can be used to extract three-dimensional information of the microstructural entities from plane sections, such deductions always require certain assumptions about the shape, distribution and even statistical information about the objects in the microstructure. For example, three-dimensional grain boundary character distributions (GBCDs) have either been stereologically calculated based on two-dimensional electron back-scatter diffraction (EBSD) maps of polycrystals [8–15] or measured by combining information partially from adjacent EBSD maps from serial sectioning experiments [16–18]. However, these methods do not yet use a complete mesh of all the grain boundaries in the given volume of microstructure.

If full, three-dimensional, realistic, grain boundary meshes were available for experimentally reconstructed polycrystals, it would be possible to make more accurate measurements of grain characteristics and better understand the evolution of the polycrystals during coarsening [19, 20]. Rowenhorst *et al* [19] showed that the integral mean curvature of the grain faces and the mean width of grains are sensitive to the method used to smooth the grain boundary surface mesh. Using their own simple and effective smoothing scheme on experimentally reconstructed three-dimensional digital images of β-titanium grains in Ti–21 S, they verified various coarsening theories and numerical models in terms of grain size distribution, number of neighboring grains vs. grain size, mean curvature and width of grains and the growth rates of individual grains. However, their constrained Laplacian smoothing technique still produces locally aliased, unsmoothed triple junctions and curvature-driven motion of the grain boundaries and junctions, resulting in either under- or over-estimation of grain boundary curvature and grain volume.

Recently, efforts have been made to generate conformal surface/volume meshes in polycrystalline materials obtained from both simulations and experiments [21–30]. One of the main purposes of these studies is to generate conformal meshes of a quality that would be sufficient for the finite element method (FEM) simulations. For example, Voronoi cells are segmented using tetrahedral elements to represent the polycrystalline grain network [21, 22]. Although this method produces elements of high quality, the short and long-range curvatures of the grain boundaries and junctions are absent from segmented elements because Voronoi cells, by their nature, contain only flat boundaries and junctions [21, 31–33], which makes it hard to use them as inputs to simulations of the grain boundary dynamics or the grain boundary distribution. Conversely, other studies have tried to extract a more realistic grain boundary mesh by fitting individual grain boundaries to polynomial functions [24–26], followed by

constructing the whole volumetric mesh inside the polycrystal [24, 25]. However, this method adopts additional polynomial surface approximation to get rid of overlaps or gaps between neighboring grains, leading to sporadic, unrealistic grain boundary traces and junctions when compared to the original voxel image of the surface of the microstructure. Other notable attempts use a minimization of surface energies by a gradient descent method [20, 28] or by a local volume conservation scheme [29, 30]. Again, the extracted grain boundaries are either almost flat [20, 28] or have unrealistic, aliased or corrugated curvatures [29, 30] due to the nature of the methods used.

In light of aforementioned challenges, we propose a simple but effective segmentation and smoothing algorithm to approximate the grain boundary networks in three-dimensional digital images of polycrystals. The goal of the proposed algorithm is to extract and smooth the grain boundary surface mesh while maintaining some features of the microstructures (such as the positions of the nodes in the grain boundary mesh, the long-range curvature of the grain boundaries and junctions, the sharpness of triple/quadruple junctions, the volume of individual grains) as far as possible and, at the same time, eliminate the aliasing from the segmentation method. Accordingly, our method uses the following steps: (1) grain boundary network segmentation using the multi-material marching cubes algorithm [34, 35], which results in a conformal triangular mesh of aliased grain boundaries and grain junctions in a polycrystal; (2) smoothing the segmented junctions of grains (i.e. the triple/quadruple junctions inside the polycrystal and the grain boundary traces on the surface of the polycrystal) using a constrained line smoothing (CLS) method; and (3) smoothing the grain boundaries inside the polycrystal using a constrained Laplacian smoothing (CLpS) method, followed by a volume conservation scheme. In this paper, the method is explained in detail using a digital unit cell image (spherical particles grown from the body-centered cubic grid) and a hypothetical test polycrystal obtained from a numerical simulation (the Monte Carlo Potts model for isotropic grain growth [36–39]). The remainder of this paper is structured as follows. In section 2, the preparation of digital test microstructures is discussed. In section 3, the segmentation and smoothing algorithms are presented in detail. In section 4, the results from the grain boundary approximation are presented. Sections 5 and 6 present the discussion and summary, respectively.

2. Preparation of hypothetical test microstructures

To test the capability of the proposed segmentation and smoothing method, two different hypothetical test microstructures were prepared. The first hypothetical test microstructure is a unit cell on a $50 \times 50 \times 50$ cubic grid where 16 half-a-quarter spheres with total volume fraction of $\sim$0.85 are grown from the body-centered cubic (BCC) grid points of the unit cell ('Microstructure I', figure 1). Voxellated images of 'Microstructure I' are shown in figure 1, where (*a*) is the surface, (*b*) are the particles and (*c*) is the matrix of the microstructure. The reason for choosing this microstructure as a test case for the proposed algorithm is because it contains a combination of straight, curved and looping junctions between volumes with flat and curved boundary faces, it also has a matrix phase with a complex geometry. The second hypothetical test microstructure is a polycrystal generated through the Monte Carlo Potts (MCP) model for isotropic grain growth [36–39] on a $100 \times 100 \times 100$ cubic grid ('Microstructure II', figure 2). The grains obtained from the MCP isotropic grain growth model are fairly equiaxed, and the grain boundaries and their junctions relax towards local equilibrium, so that the dihedral angles along the triple junctions are theoretically 120° for such ideal polycrystals. The number of grains in Microstructure II is 375. The voxellated images of the polycrystal are presented in figure 2, where (*a*) is the surface of Microstructure II and (*b*) is the image of selected grains in the corresponding polycrystal.

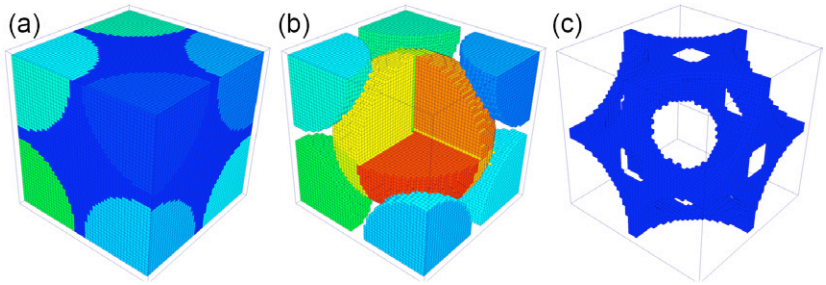

Figure 1. Voxellated images of 'Microstructure I': (*a*) surface, (*b*) particles and (*c*) matrix. Note that the microstructure possesses various kinds of features, such as straight lines, curved lines, loops, flat boundary faces, curved boundary faces and a matrix phase with a complex geometry. Individual volumes (i.e. particles and matrix) in the microstructure are randomly coloured.

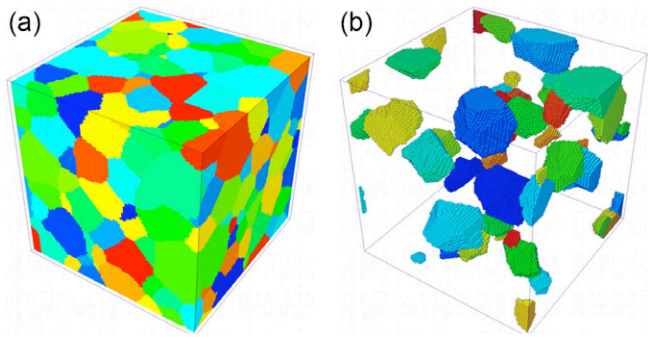

Figure 2. Voxellated images of 'Microstructure II': (*a*) surface and (*b*) selected grains. Individual grains are randomly coloured. Note that the grains in Microstructure II have fairly isotropic shapes (shown in (*b*)) because Microstructure II is generated from the isotropic Monte Carlo grain growth simulation ($100 \times 100 \times 100$ cubic grid).

3. Numerical methods

Each microstructure introduced in section 2 can also be described as a set of surfaces bounding the grains (i.e. grain boundaries inside the microstructures and grain trace areas on the surface of the microstructures) that are linked at grain boundary junctions (i.e. triple/quadruple junctions inside the microstructures and grain boundary traces on the surface of microstructure). Based on these characteristics, the segmentation and smoothing of the grain boundary network proceeds as follows.

3.1. Segmenting grain boundary network

The first step in approximating a grain boundary network is to numerically extract the three-dimensional grain boundaries in a cubic-gridded digital microstructure as a conformal triangular mesh. Such segmentation is accomplished by using the three-dimensional multi-materials marching cubes algorithm [34]. Here, we present the highlights of the multi-materials marching cubes algorithm. The details can be found elsewhere [34, 35]. The marching cubes algorithm was originally designed for the binary case [35], where eight neighboring cubic voxels in three-dimensional images, representing two different regions at maximum, can form

one local marching cube out of 256 ($=2^8$) possible different configurations. However, if there are more than two neighboring regions (eight materials, domains, grains, identifiers (IDs) and spins at maximum), as in polycrystals, then the number of different configurations for a marching cube increases enormously and it is not practical to keep track of all the possible configurations. Wu and Sullivan [34] showed that a conformal triangular boundary mesh between different regions in the three-dimensional digital image is generated by modifying the original marching cubes algorithm. At the beginning, the boundary lines on the faces of each marching cube are determined by introducing the nodes at the centers of the sides and the face of each square. This is trivial because only four neighboring pixels, representing four different materials at maximum, can form one out of 16 possible different configurations. As explained in their original work [34], two ambiguous degenerate cases arise in which the opposing diagonal corners of a face have the same material. Here, this ambiguity is treated such that the connectivity of the regions takes precedence over other conditions. Next, the nodes at the body centers of the marching cubes are introduced, if necessary, and then the loop traces of the boundary lines between different regions inside the marching cubes are identified. Once the loops are identified, triangles are constructed inside each loop. Then, all of the boundary edges and triangles are conformal throughout the microstructure. However, the segmented mesh obtained from this algorithm has aliased, stair-stepped morphology because the nodes in the mesh are the centers of the sides, the faces or the bodies of marching cubes, which requires a smoothing scheme to extract a realistic grain boundary network[4].

For this study, the IDs of the nodes, edges and triangles of the grain boundary mesh are systematically ordered and matched with the neighboring grain IDs during the mesh segmentation so that the characteristics of nodes, edges and triangles can be used easily for future analyses such as grain volume calculations and mesh smoothing. From the in-house multi-material marching cubes segmentation code, the following characteristics of the mesh can be determined.

(1) *Position and type of each node*. Position means the spatial coordinate, (x, y, z), of each node. The type of each node is the number of grains that it belongs to. Therefore, it equals 2 if the node is on a grain boundary, 3 if on a triple junction, 4 if on a quadruple junction and so on. A unique, consecutive integer ID is assigned to each node.
(2) *Two ending nodes, neighboring grain IDs and type of each edge*. The algorithm produces the IDs of the two ending nodes of each edge. It also evaluates the neighboring grain IDs and type number for each edge. Type of an edge is the number of grains it touches. For example, if an edge is surrounded by three different grains as neighbors, then the type of the edge is 3—signifying that the edge is on a triple junction. Each edge has a unique, contiguous integer ID.
(3) *Three nodes and edges, and two neighboring grain IDs of each triangle.* The IDs and sequence of three nodes and edges of each triangle are obtained. Then, the relative locations of two neighboring grains across each triangle are determined based on the sequence of three nodes of the triangle and the neighboring grain configuration in each marching cube.

Using this information, one can define each grain junction and boundary as a contiguous set of edges and triangles with the same pair of neighboring grain IDs, respectively. Then, it is sensible to define each grain in the microstructure as a set of grain boundaries having one of the neighboring grain IDs in common. The segmentation results on two test microstructures will be presented in section 4.

[4] The in-house multi-material marching cubes code was successfully used for three-dimensional characterization of a β-titanium alloy [19].

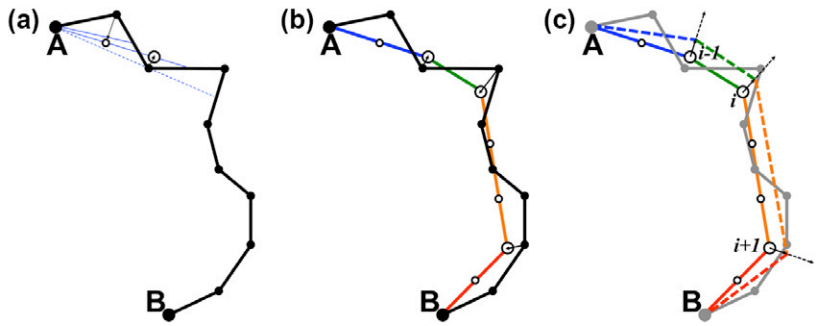

Figure 3. Schematics of: (*a*) the initially segmented grain junctions (bold black line with solid circles), (*b*) the grain junctions after the constrained line smoothing, CLS (coloured lines with open circles) and (*c*) final smoothed grain junctions after CLS + distance minimization scheme (dashed colour lines) as opposed to the original segmentation (gray bold lines with solid circles) and CLS junctions (coloured bold lines with open circles). For simple and effective visual presentation, the junction is drawn in two-dimensions. One can extend directly the two-dimensional CLS algorithm to three-dimensional one by using the three-dimensional vectors and the equation of a straight line.

3.2. Smoothing grain boundary junctions and grain traces

The second step in approximating the grain boundary network is to smooth the grain junctions (i.e. the triple/quadruple junctions inside the microstructures and the grain boundary traces on the surface of the microstructures). Once the information on the nodes and edges of the junctions from the initially segmented, aliased grain boundary mesh is identified, a CLS method is applied to approximate the junctions as a set of straight lines (usually, a set of edges), which collectively represent local and long-range curvatures of the junctions. Consider a junction[5], identified as a set of edges with two ending nodes, in figure 3. The two ending nodes of the junction, A and B, are mainly quadruple points inside the three-dimensional polycrystals and/or triple junction points on the surface of the polycrystals. One of the most important assumptions we make for the smoothing algorithm is that points, such as A and B, are constrained to be fixed during the CLS and the CLpS. In the figure, the originally aliased edges and nodes are presented with thick bold black/gray lines and solid circles, respectively. The CLS method transforms this stair-stepped, aliased, discrete line of edges into a string of smoother line segments using the following procedures[6]:

(1) Construct a straight line from starting node, A (large solid circle marked as A in figure 3(*a*)), to the midpoint of the next edge (the first thin blue solid line in figure 3(*a*)).

(2) If the distance from each of the previous nodes (including node A) to the constructed line is smaller than a user-defined threshold distance, $d_{\mathrm{J,th}}$, then construct the next straight line to the midpoint of the next edge (the second thin blue solid line in figure 3(*a*)).

(3) If the distance from any of the previous nodes to the newly constructed line is longer than $d_{\mathrm{J,th}}$, then discard the current line (for example, dashed blue line in figure 3(*a*)). Instead,

[5] For a simple and effective visual presentation, the junction in figure 3 is drawn in two-dimensions. One can extend directly the two-dimensional CLS algorithm to three-dimensional one by using the three-dimensional vectors and equation of straight line.

[6] The line straightening part in the proposed CLS method is similar to the two-dimensional grain boundary reconstruction method employed in TSL software [40].

use the previously constructed straight line and move the last node before the line to the perpendicular foot on the line (large open circle in figure 3(*a*) on the second thin blue solid line). Then, move all of the intermediate nodes to equidistant points on the newly reconstructed line segment (small open circle in figure 3(*a*) on the second thin blue solid line).

(4) Make the last node (large open circle in figure 3(*a*) on the second thin blue solid line) the starting node for the next line segmentation.

(5) March through all edges in the junction and repeat steps from (1) to (4) until the last node, B, is reached (figure 3(*b*)).

As shown in figure 3(*b*), this procedure yields a string of linear segments (thick solid coloured lines with open circles) that lie on top of the originally aliased junctions (solid circles and thick black lines in the figure).

During the CLS method, the junctions move toward the center of local curvature, leading to changes in the volumes of neighboring grains for most of the three-dimensional cases. Now, it is necessary to move the nodes to the new positions, from which a certain characteristic measure of the neighboring geometry is minimized (or maximized). For two-dimensional cases, an area conservation scheme has been successfully implemented for the heterophase interface boundaries [41]. However, in three dimensions, no analytical approach was found to calculate the new positions of the nodes on the junctions, so that the original volumes of three or more neighboring grains are recovered at the same time. Instead, a distance minimization scheme is applied after each CLS process for this study.

(1) Find unit translation vectors, $\boldsymbol{u}$ (dashed arrows in figure 3(*c*)), for each ending node (large open circles in figure 3(*c*)) of the individual smoothed linear segments (bold coloured lines in figure 3(*c*)) such that $\boldsymbol{u}_i = (\boldsymbol{e}_{i,i-1} + \boldsymbol{e}_{i,i+1})/|\boldsymbol{e}_{i,i-1} + \boldsymbol{e}_{i,i+1}|$, where $\boldsymbol{e}_{i,i-1}$ and $\boldsymbol{e}_{i,i+1}$ are unit vectors from the current ending node i to the neighboring ending nodes, $\tilde{i}1$ and $i+1$. The translation vectors for A and B are zero because they are supposed to be fixed during smoothing procedures. If the previous CLS step produces a straight line from point A to B, then a unit vector of the sum of the vectors from all of the nodes between A and B to the corresponding nodes of the original, aliased mesh, $\boldsymbol{u}_{\mathrm{mid}}$, is calculated and assigned to the midpoint between A and B.

(2) Decide the direction of the unit translation vectors, $[\mathbf{sign}(\boldsymbol{u}_i)]$, such that $\mathbf{sign}(\boldsymbol{u}_i) = 1$ if $[\boldsymbol{u}_{i-1} \cdot \boldsymbol{u}_i] < [\boldsymbol{u}_{i-1} \cdot (-\boldsymbol{u}_i)]$ and $\mathbf{sign}(\boldsymbol{u}_i) = -1$ if not. This maintains the slopes of the approximated linear segments from the previous CLS step as far as possible during the distance minimization process.

(3) Find the value of a parameter t_{root} such that the sum of the distances from the originally segmented nodes (solid gray circles in figure 3(*c*)) to the line segments, newly translated by $t_{\mathrm{root}}\boldsymbol{u}_i$ (dashed coloured lines in figure 3(*c*)), is minimized. In order to find t_{root}, the minimum value of the sum of the distances is tracked as the line sweeps through the neighboring regions using $t\boldsymbol{u}_i$ with a discrete step, $t = \pm(n\mathrm{d}t)$, $n = 0, 1, 2, \ldots n_{\max}$. Here, $\mathrm{d}t = \beta \mathrm{d}x$, where $\mathrm{d}x$ is the real step size of the voxel and β is a user-specified parameter related to the sweeping resolution. $n_{\max}$ is defined as $d_{\mathrm{b}}/\mathrm{d}t$, where d_{b} is the user-defined distance to the bounds of neighboring regions swept. For example, if $\beta = 0.01$ and $d_{\mathrm{b}} = 3\,\mathrm{d}x$, then the approximated line sweeps through the regions bounded by 3 voxel length with 100 times higher resolution than original image. Then, it is sensible to consider that t_{root} is very close to the real solution.

(4) Move all intermediate nodes to the equidistant positions on the new linear segments constructed from $t_{\mathrm{root}}\boldsymbol{u}_i$ (dashed coloured lines in figure 3(*c*)). Because the local curvature

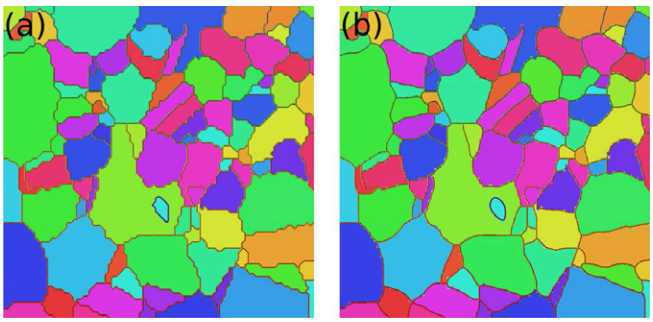

Figure 4. Images of: (*a*) the originally segmented and (*b*) the smoothed grain boundaries on a two-dimensional section of a real three-dimensional microstructure map with $d_{\mathrm{J,th}} = 0.1$ (in units of pixel length), $N_{\mathrm{J,Iter}} = 4$ and $\beta = 0.001$ for the verification of the smoothing method. Individual grains are obtained from an electron back-scatter diffraction image of IN100 (a Ni-based superalloy). It is clear from the figure that the proposed method successfully smoothes the grain boundaries by eliminating the unphysical stair-steps from the original mesh and approximating both local and long-range curvatures of the grain boundaries.

of the newly segmented lines from the CLS is to be maintained during the distance minimization process and the obtained value for t_{root} is usually very small (typically, the average value of t_{root} is less than 0.5 when normalized by dx), it is reasonable to use the $\boldsymbol{u}_i$, as explained above, for local relocation of the nodes.

Note that the proposed algorithm for smoothing grain boundary junctions and traces retains the number of nodes along the lines. An iteration of the combined CLS and the distance minimization process is completed when they are applied in both directions along each junction between two end-points (i.e. from point A to point B and from point B to point A in figure 3). Currently, the smoothing process stops after a user-prescribed number of iterations, however, one can stop the process by setting up a terminal threshold value for the average distance moved by the node. For example, figure 4 shows the images of (*a*) the originally segmented and (*b*) the smoothed grain boundary traces on a two-dimensional surface of a real three-dimensional microstructure image with $d_{\mathrm{J,th}} = 0.1$ (in units of pixel length), number of iterations, $N_{\mathrm{J,Iter}} = 4$, and a sweeping resolution, $\beta = 0.001$ for the verification of the method. It is clear from the figure that the proposed method successfully smoothes the grain boundary traces on the surface of microstructure by eliminating the unphysical aliasing (i.e. stair-steps) from the original mesh and approximating both the local and long-range curvatures of the grain boundary traces. Typically, such a process results in an average displacement of $\sim$0.2 (in pixel length) for the relocated nodes from their original positions. Achieving such a small shift confirms that the newly smoothed, distance-minimized segments are defined inside the regions bounded by unit pixel length from the initial segments. Also, the grain volume changes due to this process in three-dimensional polycrystals are very small. The quantitative results will be presented in section 4.

3.3. Smoothing grain boundaries

As the final step, the nodes on the grain boundaries in the polycrystal are smoothed while the previously smoothed triple/quadruple junctions are fixed in space. Specifically, a CLpS

method, followed by a volume conservation scheme, is used to remove the stair-stepped artifacts from the original marching cubes mesh, and to maintain the local and long-range curvatures of the grain boundaries at the same time. In a conventional Laplacian smoothing of the surface mesh, the new position of each node, X_j^{new}, is calculated using the current position of the node j, X_j, and the positions of its N neighboring nodes, X_{k}, and it is given by

$$X_j^{\text{new}} = X_j + \lambda \left[\frac{1}{N} \sum_{k=1}^{N} \left(X_k - X_j \right) \right] = (1 - \lambda)\, X_j + \lambda \bar{X}_{\text{neigh}}, \tag{1}$$

where $\bar{X}_{\text{neigh}}$ is the average position of the N neighboring nodes, and λ is a weighting factor between 0 and 1. Thus, the conventional Laplacian smoothing forces the node j to move to a position that divides the line segment between X_j and $\bar{X}_{\text{neigh}}$ with a ratio of $\lambda : (1 - \lambda)$.

Conventional Laplacian smoothing is computationally efficient and easy to implement. However, it has some disadvantages, such as: it creates a non-uniform element size and poor triangle quality in general, even after a few iterations; and, it may generate folded or inverted triangles depending on the concavity of the polygons that the neighboring triangles form. For the calculation of metrics associated with grain boundaries, such as the GBCD, the shapes and volumes of grains, the first disadvantage is trivial and its effect on such analyses can be ignored. However, the second disadvantage is problematic. It usually occurs most often at the triangles adjacent to triple/quadruple junctions with significant local curvature variations. To minimize this, we calculate a modified average position of the neighboring nodes, $\bar{X}_{\text{neigh}}$, for the current node j, as given by

$$\bar{X}_{\text{neigh}} = \frac{1}{N} \sum_{k=1}^{N} \left[(1 - \delta) X_k + \delta \{ \alpha X_k + (1 - \alpha) X_j \} \right], \tag{2}$$

where α is a weighting factor between 0 and 1, and $\delta = 0$ if the neighboring node k is on grain boundary and $\delta = 1$ if on triple/quadruple junctions. Choosing a value of $\alpha = 0.3$–0.5 mitigates the effect of the fixed triple/quadruple junctions on the possible folding or inversion of triangles attaching to them during smoothing. Also, a further constraint is imposed on the method so that no node can travel further than a user-specified maximum distance during an iteration of smoothing. This is realized as

$$X_j^{\text{new}} = \begin{cases} \bar{X}_{\text{neigh}} & \text{if } \left| \bar{X}_{\text{neigh}} - X_j \right| \leqslant d_{\text{GB,th}}\, \text{d}x, \\ X_j + (d_{\text{GB,th}}\, \text{d}x) \dfrac{\bar{X}_{\text{neigh}} - X_j}{\left| \bar{X}_{\text{neigh}} - X_j \right|} & \text{otherwise,} \end{cases} \tag{3}$$

where $d_{\text{GB,th}}$ is a user-defined normalized threshold distance between 0 and 1 (in voxel length, dx). Then, $d_{\text{GB,th}}\, \text{d}x$ is the actual maximum distance that a node is allowed to travel during each smoothing iteration. In this way, we have an indirect control over the maximum and average displacement of the nodes after smoothing, combined with proper $d_{\text{GB,th}}$ and number of smoothing iterations, $N_{\text{GB,Iter}}$.

After a user-defined number of the CLpS iterations, $N_{\text{GB,Iter}}$, are applied on each grain boundary, a volume conservation scheme follows. During the CLpS, each grain boundary moves toward the center of its local curvature, leading to changes in the volumes of neighboring

grains across the boundary for most of the cases. In other words, during the CLpS, grain growth occurs. The algorithm for the volume conservation scheme is given as follows:

(1) Calculate the unit normal vectors of triangles on each grain boundary and make all of them point towards one of two neighboring grains in common, $\boldsymbol{n}$.
(2) Calculate a translation vector, $\boldsymbol{s}_i = \eta\ [\boldsymbol{w}_i/|\boldsymbol{w}_i|]$, for each node i on the grain boundary. $\boldsymbol{w}_i$ is an area weighted vector, given by $\boldsymbol{w}_i = [\Sigma_j\ (\boldsymbol{A}_j\boldsymbol{n}_j)]/(\Sigma_j \boldsymbol{A}_j)$, where $\boldsymbol{A}_j$ and $\boldsymbol{n}_j$ are the area and the normal vector of neighboring triangle j of node i. Also, the effect of tension from the fixed grain junctions on each node of the grain boundary is taken into account and realized by setting $\eta = \Delta_j/\Delta_{\max}$, where Δ_j is the minimum distance from node j to the junctions of the grain boundary and $\Delta_{\max}$ is the maximum among Δ_js of all nodes on the grain boundary.
(3) Define a signed volume, V, of each grain boundary consisting of T triangles such that $V = \Sigma_k[(\boldsymbol{r}_k.\boldsymbol{n}_k)\boldsymbol{A}_k/3.0]$, $k = 1, 2, 3, \ldots, T$, where $\boldsymbol{r}_k$ is the position vector at the midpoint of triangle k[7]. Then, find the value of a parameter t_{root} such that the signed volume, V_{new}, defined by each translated grain boundary by $t_{\text{root}}\boldsymbol{s}_j$, are comparable to the signed volume of that grain boundary defined by the initially segmented marching cubes mesh, V_{mcubes}. For this study, t_{root} is chosen when $(|V_{\text{new}}|-|V_{\text{mcubes}}|)/|V_{\text{mcubes}}| < 10^{-6}$. The conventional bisection method is applied for this process.
(4) Move all the grain boundary nodes by the amount of $t_{\text{root}}\boldsymbol{s}_j$.

The proposed grain boundary smoothing algorithm retains the numbers of nodes, edges and triangles from the marching cubes segmentation. Typically, this combined process results in a small average shift for the relocated nodes on grain boundaries. Achieving such a small average shift, together with conserving the grain volumes, and both the local and long-range curvatures of the grain boundaries, is a desirable result for smoothing the grain boundary network from three-dimensional digital images of polycrystals. The relevant quantitative analyses will be given in section 4.

4. Results

The surface meshes of the grain/phase boundary networks of three microstructures are obtained using the in-house multi-material marching cubes code that were described in section 3. For smoothing the originally aliased segmented meshes, the following parameters are used for all three microstructures, consistently: $d_{\text{J,th}} = 0.1$, $N_{\text{J,th}} = 6$ and $\beta = 0.005$ for the CLS of the junctions and the grain boundary traces on the surfaces of the microstructure, and $d_{\text{GB,th}} = 0.1$, $N_{\text{GB,th}} = 6$ and $\alpha = 0.45$ for the CLpS of the grain boundaries. The parameters are chosen so that the average displacement of the shifted nodes is comparable to, or smaller than, unit step size of the digital images (unit voxel length).

4.1. Surface meshes from segmentation and smoothing

In figures 5 and 6, the originally segmented surface mesh generated from the in-house multi-material marching cubes code ((*a*) and (*b*)), and the smoothed mesh obtained from the algorithm ((*c*) and (*d*)) of Microstructure I and II are presented, respectively. In the figures, (*a*) and (*c*) are the images of the surface of each microstructure while (*b*) and (*d*) show selected grains of each microstructure before and after smoothing, respectively. It is evident from the images that the stair-stepped, aliased grain junctions, grain boundary traces and grain boundaries of

[7] For each triangle, k, in a grain boundary, a volume is defined, given by $(\boldsymbol{r}_k \cdot \boldsymbol{n}_k)\boldsymbol{A}_k/3.0$, a signed volume formed by connecting the origin and the vertices of the triangle.

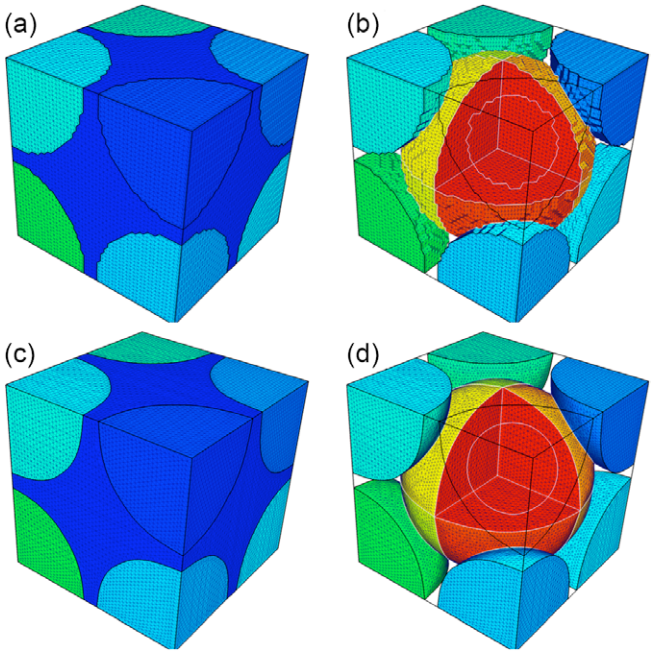

Figure 5. Images of the originally segmented ((*a*) and (*b*)), and the smoothed mesh ((*c*) and (*d*)) of Microstructure I, respectively. The numbers of nodes, edges and triangles are 37281, 114510 and 77240, respectively. In the images, (*a*) and (*c*) show the surface of the microstructure while (*b*) and (*d*) show selected grains of the microstructure before and after smoothing, respectively. As is evident from the images, the stair-stepped, aliased grain junctions, grain boundary traces and grain boundaries of the originally segmented microstructures are successfully approximated as smoothly changing lines of edges and patches of triangles with reasonable long-range curvatures based on the originally segmented images. Each grain boundary in (*b*) and (*d*) is coloured using a random colour scheme.

the originally segmented microstructures are successfully approximated as smoothly changing lines of edges and patches of triangles with reasonable long-range curvatures based on the originally segmented images. In other words, the curvatures of the lines and boundaries from the proposed algorithm are determined only by the segmented images from the voxellated microstructures, not by any assumed laws of motion or mathematical functional. Also, note that the triangles attached to the grain junctions and grain boundary traces are elongated toward the centers of the boundaries they belong to. This is a consequence of the imposition of the parameter α in equation (2) to keep them from folding or inverting as much as possible.

4.2. Displacement of nodes after smoothing

As previously mentioned, the parameters for the smoothing process are designed so that the average displacement of the shifted nodes is smaller than, or comparable to, the unit step size of the digital images (unit voxel length). Now, it is of immediate interest to measure how far the nodes move during the smoothing. To do that, a normalized displacement of the shifted nodes, d'_s, is defined as $d' = d_s/\mathrm{d}x$, where d_s is the distance shifted for each node from its original position and $\mathrm{d}x$ is the length of unit step size of the cubic voxel for the images. For

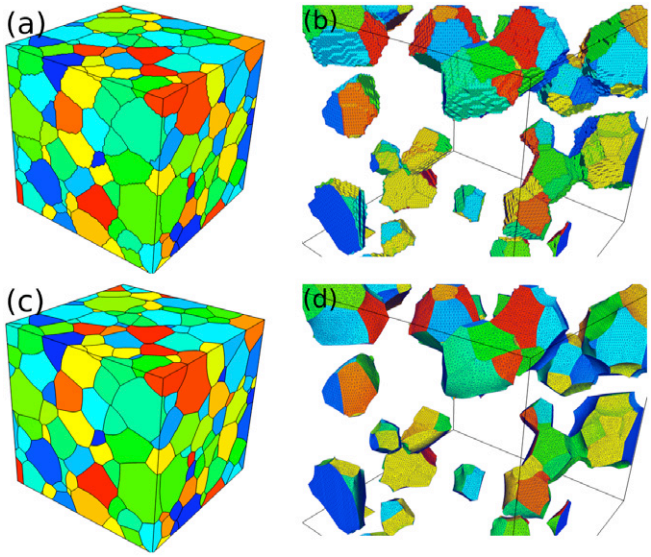

Figure 6. Images of (*a*) and (*b*) originally segmented, and (*c*) and (*d*) smoothed mesh of Microstructure II. The numbers of nodes, edges and triangles are 330655, 1034070 and 703791, respectively. In the images, (*a*) and (*c*) show the surface of the microstructure while (*b*) and (*d*) show selected grains of the microstructure before and after smoothing, respectively. As is evident from the images, the stair-stepped, aliased grain junctions, grain boundary traces and grain boundaries of the originally segmented microstructures are successfully approximated as smoothly changing lines of edges and patches of triangles with reasonable long-range curvatures based on the originally segmented images. Each grain boundary is coloured using one the neighboring grain IDs.

this calculation, the quadruple points inside the microstructure, triple points on the surface of the microstructure and the nodes on the grain trace area on the surface of the microstructure are excluded because they are either fixed or move only laterally on the flat surface of the microstructure during smoothing. In figure 7, the distributions of d'_s for (*a*) Microstructure I, and (*b*) Microstructure II are shown, respectively. The average distance and the standard deviation in the distributions are 0.21 and 0.10 for Microstructure I, and 0.24 and 0.12 for Microstructure II. Note that the proposed algorithm positions the smoothed grain boundary networks near the original segmentation well within a unit voxel length. Also, note that d'_s is an overestimating measure of the shifted distance of each node because d'_s is always equal to or bigger than the distance from its originally segmented position to the smoothed boundary, which once again confirms the claim above.

4.3. Volumes of grains

Each step in the proposed segmentation and smoothing procedure involves changes in the grain volume. During segmentation, the volumes of individual grains in the voxellated microstructure are not maintained because the multi-materials marching cubes algorithm uses the midpoints of sides and face/body centers of each marching cube for triangulation of the boundaries. The maximum grain volume change caused by the segmentation is about 3% for the microstructures examined in this paper. However, we do not attempt to rectify this error because

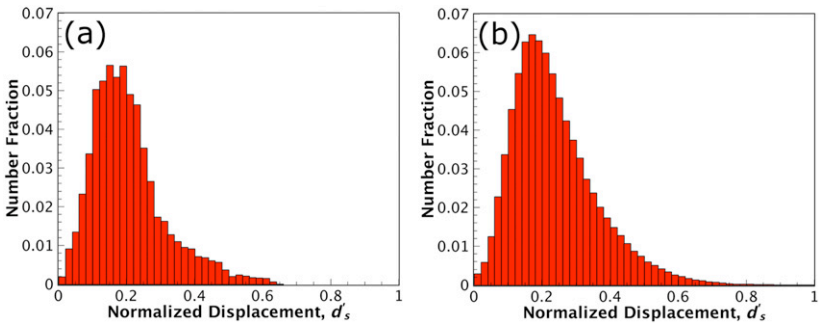

Figure 7. Distributions of normalized displacement, d_s', after smoothing (*a*) Microstructure I and (*b*) Microstructure II, respectively. The bin size is 0.02. The number of nodes, and the maximum distance, the average distance and the standard deviation in the distribution are (*a*) 23280, 0.64, 0.21 and 0.10; and (*b*) 60364, 1.19, 0.24 and 0.12, respectively. Note that the proposed algorithm positions the smoothed grain boundary networks near the original segmentation well within unit voxel length.

we lack information about where the grain boundaries are exactly at this resolution. Instead, we focus on conserving the volume of each grain obtained from the originally segmented mesh using the marching cubes algorithm during smoothing. During grain junction smoothing using the CLS method, the changes in volumes of the neighboring grains are found to be very small (less than 0.2% with a maximum change of ∼0.6%) for the microstructures considered here. This confirms that the distance minimization scheme in the proposed CLS method works successfully in terms of grain volume conservation.

The change in volume of each grain i, ΔV_{i}, between the volume from the originally segmented mesh, $V_{i,\mathrm{mcubes}}$ (labeled as '*mcubes*') and that from the smoothed mesh, $V_{i,\mathrm{smoothed}}$ (labeled as '*smoothed*'), is defined as $\Delta V_{\mathrm{i}} = V_{i,\mathrm{mcubes}} - V_{i,\mathrm{smoothed}}$. In figure 8, the errors of ΔV_{i} created through the smoothing process are plotted against $V_{i,\mathrm{mcubes}}$ for all of the grains in (*a*) Microstructure I, and (*b*) Microstructure II, respectively. Note that most of the grains experience less than 0.2% volume changes during the smoothing procedure and that smaller grains experience larger volume changes due to the relatively poor resolution of both the original images and segmented triangles when compared to the larger ones, as one might reasonably expect [42].

4.4. Dihedral angles along triple junctions

One of the design features of the smoothing algorithm is to prevent the triangles attached to the triple/quadruple junctions from folding or inverting by imposing a parameter α during the CLpS method. This is crucial for future applications of the three-dimensional grain boundary mesh of the polycrystals to microstructural characterization, such as the direct measurement of the grain boundary energy distributions (GBEDs) [43]. The triangles obtained from the original marching cubes algorithm are neither folded nor inverted, forming only discrete values of dihedral angles between neighboring triangles, such as 180°, 90° and 45°. Because of the constraints imposed by small threshold distances $d_{\mathrm{J,th}}$ and $d_{\mathrm{GB,th}}$ during smoothing, neither folded nor inverted triangles are found on the grain boundaries of the smoothed mesh for the test microstructures. The last region where one might need to check the quality of the triangles is the triple junction. In figure 9, the dihedral angle distributions of the triple junctions of the smoothed mesh are given for (*a*) Microstructure I and (*b*) Microstructure II, respectively. The

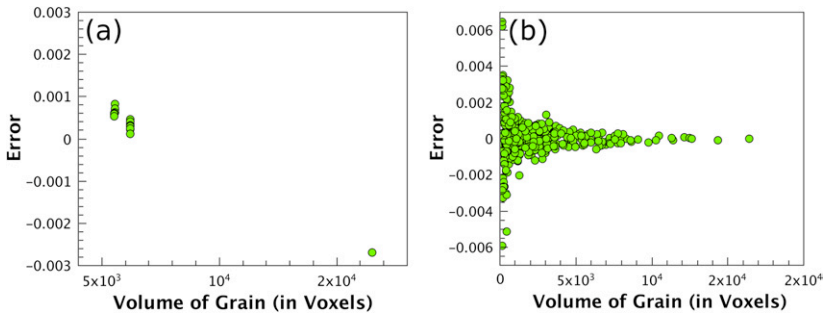

Figure 8. Volume of grains from initial segmentation versus error in volume change of each grain after smoothing (*a*) Microstructure I and (*b*) Microstructure II. The error term is define by $(V_{\text{mcubes}}-V_{\text{smoothed}})/V_{\text{mcubes}}$. Note that most of the grains have less than 0.2% volume changes during smoothing procedure, and that smaller grains experience larger relative volume changes.

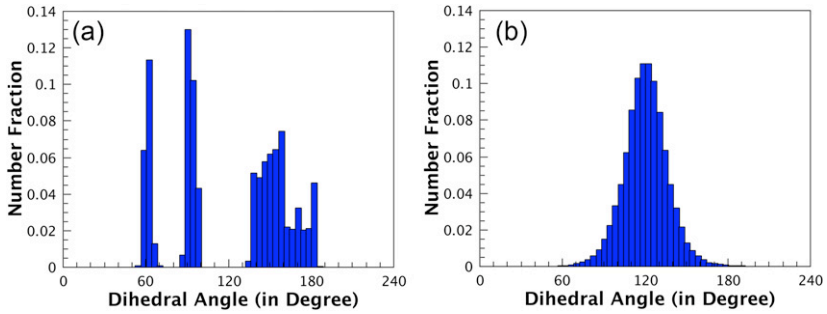

Figure 9. Dihedral angle distributions at the triple junctions of the smoothed mesh (*a*) Microstructure I and (*b*) Microstructure II. The dihedral angles are calculated using the winding sequence of neighboring triangle triplets, which share a common edge at the triple junctions. The binning size is 4°. For Microstructure I (*a*), a discrete distribution is found due to the regularity of the unit cell structure. For the simulated polycrystalline case ((*b*)), the distributions are well centered at 120° with a relatively small standard deviation of 15.95°, signifying that the proposed smoothing algorithm recovers the isotropic nature of the triple junctions in the simulated microstructure.

dihedral angles are calculated using the winding sequence of neighboring triangle triplets, all of which share a common edge at the triple junctions. For Microstructure I, the distribution is discrete due to the regularity of the position, shape and size of particles in the unit cell structure (figure 9(*a*)). For the simulated polycrystalline case, the distributions are centered at 120° with relatively small standard deviations of 15.95° (figure 9(*b*)), which suggests that the smoothing algorithm correctly recovers the isotropic nature of the triple junctions in the microstructures generated from the isotropic grain growth simulation.

4.5. Grain morphology

Figure 10(*a*) shows one of the large grains and (*b*) shows one of the small grains in Microstructure II after smoothing. In the figure, thick black lines are triple junctions while thick white lines are quadruple junctions bounding the grains. Each grain boundary is coloured by its neighboring grain ID, as prescribed in the voxellated microstructure before

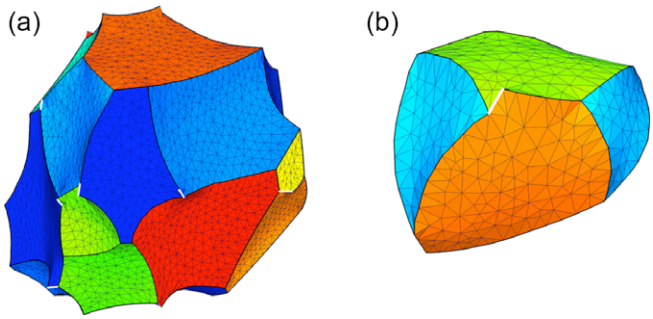

Figure 10. (*a*) One of the large grains and (*b*) one of the small grains in Microstructure II after smoothing. Thick black lines are triple junctions and thick white lines are quadruple junctions bounding the grains. Each grain boundary is coloured according to neighboring grain ID using a random colour scheme. Note that the large grain has a concave shape and more neighboring grains whereas the small grain is surrounded by convex boundaries and less neighboring grains, signifying that, from the microstructural images, the proposed algorithm recovers the first order characteristic of the isotropically grown polycrystals.

the segmentation and smoothing processes. As is obvious in the figure, the quadruple junction lines are sporadically located on the grain boundary edges, which are unrealistic features of the ideal, stable polycrystalline structures because of the finite resolution of the original voxellated microstructural image. One can clean out these features in the original voxellated microstructures before segmentation if needed. Note that the large grain has a concave shape and more neighboring grains whereas the small grain is surrounded by convex boundaries and less neighboring grains, signifying that the proposed algorithm recovers the first order characteristic of the grains in the isotropically grown polycrystals by using only the digital images.

5. Discussion

5.1. Effect of the cubic simulation lattice on the dihedral angle distribution obtained from the proposed algorithm

One might argue that the observed values of the standard deviations in dihedral angles of Microstructure II are too large to permit a conclusion that the smoothing algorithm works well because the microstructures from the isotropic grain growth simulations are expected to have a theoretical dihedral angle of 120° at the triple junctions. By contrast, Harker and Parker [44] analytically showed that the angles between the traces of two vertical grain boundaries at 120°, measured on the arbitrary intersection planes, form a Gaussian distribution with a peak at 120°. The reason for the angles not being exactly 120° is because they are measured from the two-dimensional planes at an arbitrary angle. From the dihedral angle measurements on the two-dimensional sections of the digital polycrystals obtained from the Monte Carlo Potts isotropic grain growth simulation, Chandross and Holm [45] also observed that wider dihedral angle distributions occur because of 'geometric mapping' [45] onto a cubic lattice and 'boundary anisotropy' [45] of the cubic lattice. Based on these reports, it is now of interest to compare the dihedral angle distribution from the smoothed image of Microstructure II, generated through the Monte Carlo Isotropic grain growth, to the previous

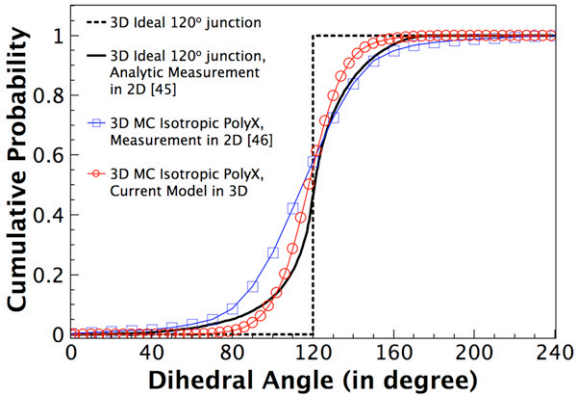

Figure 11. Cumulative distribution of the dihedral angles in Microstructure II after the smoothing, compared to those from previous reports [44, 45]. In the figure, the dashed line and the solid line represent the cases of an ideal polycrystal where all of the grain junctions meet at 120°, but the dihedral angles are measured in two dimensions [44] and three dimensions, respectively. Note that the proposed smoothing algorithm produces the narrowest measured distribution, while its deviation from the ideal 120° distribution (dashed line) can be ascribed to both the finite resolution and the anisotropy of the cubic simulation lattice used for this study as similarly proposed [45].

findings. In figure 11, the cumulative distribution of the dihedral angles in Microstructure II after smoothing is compared to those from previous reports. In the figure, the dashed line and the solid line represent the cases of an ideal polycrystal where all of the grain junctions meet at 120°, but the dihedral angles are measured in two dimensions and three dimensions, respectively. Note that the smoothing algorithm produces a narrower distribution than the two-dimensional measurements, as expected. However, even though the grain boundary network in Microstructure II is smoothed in three dimensions, there is no particular expectation that the dihedral angles have to be precisely 120° in an evolving microstructure using the Monte Carlo Potts model on a cubic lattice. Such deviations can be ascribed to both the finite resolution and the anisotropy of the cubic simulation lattice identified by the previous work [45]. This issue is explored in more detail as follows.

To examine the effect of the finite resolution, the type and the anisotropy of the simulation lattice on the measured dihedral angles, 18 full and partial hexagonal columns are mapped onto a $60 \times 70 \times 50$ cubic lattice (figure 12(a)). Also, a magnified portion of a cross section of the structure, perpendicular to the columnar direction (x–y planes in figure 13), is examined to compare the dihedral angles around triple junctions from the discrete image (figure 12(b)) to analytical values (the analytical dihedral angle around the triple junctions between hexagonal columns is 120°). Interestingly, the ratio of the distance m along the x direction to the vertical distance n along the y direction, measured from the digitized image (red arrows in figure 12(b)), is 1.667 (10 : 6 in voxel length ratio), whereas the corresponding analytical value of the ratio is $\sqrt{3} = 1.732$. In other words, the dimensional ratio of the hexagon mapped with a finite resolution is not commensurate with the analytical value, even though the discrepancy decreases as the resolution of the mapping grid increases. This dimensional discrepancy has an effect on the local triple junction geometry, such that the dihedral angles around the triple junctions are not 120°. In figure 12(b), two triple junctions are approximated with straight white and black arrows, respectively. The white triple junction is composed of vectors with components of (10, 5), (−10, 5) and (0, −5) in voxel length, and the black triple junction has vectors with

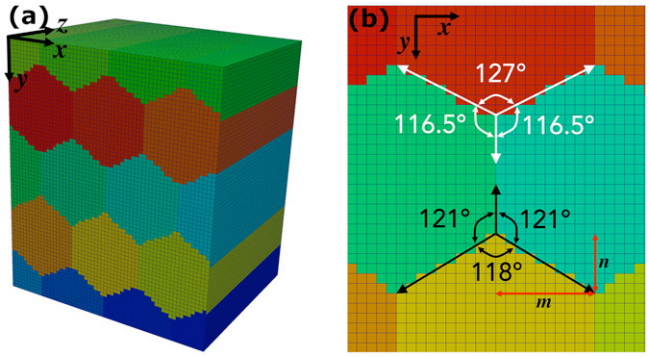

Figure 12. (*a*) Hexagonal columns mapped onto a $60 \times 70 \times 50$ cubic lattice, and (*b*) a magnified portion of a cross section of the structure (x–y plane) perpendicular to the columnar direction z. Note that the dimensional ratio of the mapped hexagon with this finite resolution is not commensurate with analytical value, which, in turn, creates triple junction geometries of which the dihedral angles are not 120° but range approximately from 114° to 130°.

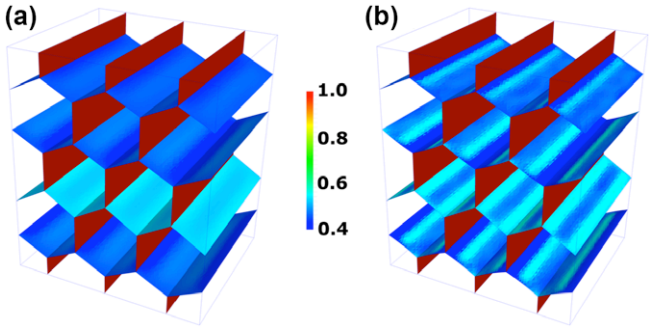

Figure 13. Effect of the proposed smoothing algorithm on the curvatures of the boundaries of the discrete hexagonal columns: (*a*) variation in direction cosine values of the boundaries with respect to x direction after smoothing without the volume correction step, and (*b*) variation in direction cosine values of the boundaries with respect to x direction after smoothing with the volume correction step. Note that, for case (*b*), long-range curvatures are introduced to individual boundaries, depending on their voxellated local curvatures, while for case (*a*), each boundary has a unique, relatively constant direction cosine value, close to the values determined by the voxellated local triple junction geometry.

components of $(-10, -6)$, $(10, -6)$ and $(0, 5)$. Obviously, two exemplary triple junctions are not identical, and, as labeled in the figure, the dihedral angles are different from 120°. In fact, there are found five different triple junction geometries in this structure, of which the dihedral angles range approximately from 114° to 130°. Therefore, there are inherent errors in dihedral angles measured from the discrete lattice image of the microstructure. Even more complicated situations arise if the hexagonal columns are mapped along an arbitrary direction other than x, y or z, and the measured dihedral angles will vary due to the anisotropy of the mapping grid.

More importantly, in figure 12(*b*), it should be noted that the truncated pixel areas with different colours across the straight boundaries are not equal to each other. In other words, the original volumes of the neighboring hexagonal columns are not maintained if flat boundaries

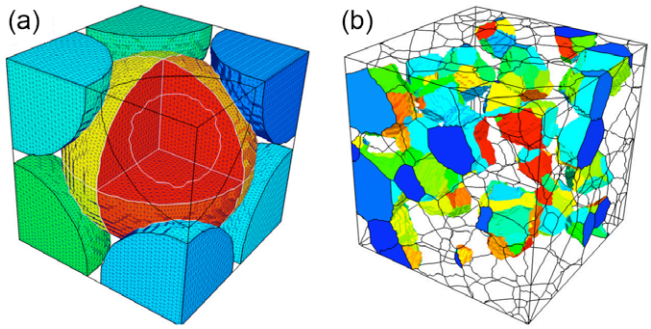

Figure 14. A portion of smoothed mesh of: (*a*) Microstructure I and (*b*) Microstructure II with $d_{\mathrm{J,th}} = 0.05$, $N_{\mathrm{J,th}} = 2$ and $\beta = 0.005$ for the CLS of the junctions, and $d_{\mathrm{GB,th}} = 0.033$, $N_{\mathrm{GB,th}} = 3$ and $\alpha = 0.45$ for the CLpS of the grain boundaries. Note that, during smoothing, such small parameters help maintaining the local features of the original segmentation, if necessary, compared to the results in figures 5 and 6.

are used for this structure. For such cases, after the proposed volume conservation step is applied, the boundaries between neighboring columns (i.e., prism planes) are not flat and, instead, show long-range curvature since the boundaries have to move to maintain the volumes of neighboring objects based on the original segmented image. In figure 14, the effect of the smoothing algorithm on the curvatures of the prism planes is presented using the absolute direction cosine values of the local boundary normal vectors with respect to x direction of the microstructure. In figure 13(*a*), the variation in direction cosines after smoothing without the volume correction step is presented. For this case, each prism plane has a unique, nearly constant direction cosine value, which is close to the values determined by the voxellated local triple junction geometry. For the volume conservation case that is shown in figure 13(*b*), long-range curvatures are introduced to individual boundaries, depending on their voxellated local curvatures. The standard deviation in the dihedral angles measured from the surface mesh in figure 13(*a*) is 5.80°, which is a reasonable value, considering that the dihedral angles from the voxellated triple junctions is found to range from 114° to 130°. This suggests that the smoothing without the volume conservation step generates flat boundaries very close to straight lines on the triple junction geometries in figure 12(*b*). By contrast, the standard deviation in the dihedral angles measured from the surface mesh in figure 13(*b*) is 14.52°; signifying that, for this structure, the volume conservation scheme broadens the dihedral angle distribution, leading to greater uncertainty, which needs to be avoided when approximating boundaries. In other cases, depending on which characteristics of the microstructure are being measured, the volume conservation step is sometimes obviously necessary (for example, 'Microstructure I' in figure 5). Therefore, the dihedral angles measured (figure 9(*b*)) in the microstructures obtained from the isotropic grain growth simulations using the MCP (figure 6(*c*)) include errors from the finite resolution and the anisotropy of the cubic simulation lattice. Since most of the grain boundaries in polycrystals result from microstructural evolution through curvature-driven motion, it seems to be reasonable to include the volume conservation step during smoothing the grain boundary network in polycrystals in order to realize the curvatures of the boundaries.

5.2. Effect of the parameter values on mesh smoothing

During the CLS of the junctions and the CLpS of the grain boundaries, the threshold distances, $d_{\mathrm{J,th}}$ and $d_{\mathrm{GB,th}}$, determine the roughness of the smoothed junctions and grain boundaries with a given smoothing iteration number, $N_{\mathrm{J,th}}$ and $N_{\mathrm{J,th}}$, respectively. Larger threshold distances will generate flatter junctions and grain boundaries during the smoothing process in a given iteration, whereas smaller threshold distances produce wavier junctions and grain boundaries. Conversely, one can also control the roughness of the smoothed mesh by using small iteration numbers with a given threshold distance value. Figure 14 shows a portion of mesh of (*a*) Microstructure I and (*b*) Microstructure II after smoothing using $d_{\mathrm{J,th}} = 0.05$, $N_{\mathrm{J,th}} = 2$ and $\beta = 0.005$ for the CLS of the junctions, and $d_{\mathrm{GB,th}} = 0.033$, $N_{\mathrm{GB,th}} = 3$ and $\alpha = 0.45$ for the CLpS of the grain boundaries. Compared to the results given in figures 5 and 6, obtained from $d_{\mathrm{J,th}} = 0.1$, $N_{\mathrm{J,th}} = 6$ and $\beta = 0.005$ for the CLS, $d_{\mathrm{GB,th}} = 0.1$, $N_{\mathrm{GB,th}} = 6$ and $\alpha = 0.45$ for the CLpS, it is clear that such small parameter values help to maintain the local features of the original segmentation. Even though the resultant meshes in figure 14 only seem to have artifacts resulting from insufficient smoothing of the corresponding microstructures, one can preserve local nonuniformities in the curvature, if any are physically significant (for example, protrusions or cusps), by using combinations of small parameter values during the smoothing process.

6. Summary

In this paper, a segmentation and smoothing algorithm for approximating the boundary networks in the digital images of polycrystalline microstructures is proposed. The main goal of the proposed method is to extract the realistic grain boundary network from digital images of polycrystalline microstructures, possibly obtained from either simulations or experiments, while eliminating any artificial, aliased features from segmentation while at the same time maintaining their characteristic features. Specifically, the algorithm performs a simple segmentation of the grain boundary network using the multi-materials marching cubes algorithm and allows the grain boundaries to get smoothed while maintaining the segmented volumes of grains. The stair-stepped, aliased grain junctions, grain boundary traces and grain boundaries of the originally segmented microstructures are successfully eliminated, resulting in smooth lines of edges and patches of triangles with the long-range curvatures based on the originally segmented images. The curvatures of these lines and boundaries are realized based on the original voxellated and segmented microstructural images, not by the application of any assumed physical processes. Moreover, the local features of the originally segmented mesh can be maintained by assigning small values to the selected parameters during smoothing. The smoothing algorithm produces smooth grain boundary networks while at the same time it positions them near the original segmentation well within one unit voxel length and conserves the grain volumes from the original segmentation. The dihedral angle distribution in the Monte Carlo Potts model generated isotropic polycrystalline image after segmentation and smoothing is verified to be very reasonable when compared to the previously reported analytic solutions and measurements. However, the dihedral angle distributions in the digitized microstructures are found to have errors, depending on (1) the characteristics of microstructures under examination, and (2) the type and the anisotropy of the mapping lattice for digital microstructural images. From the individual grain morphologies, it is found that the proposed algorithm successfully recovers the expected variation of curvature with grain size. Given these findings, one can apply the proposed method to the real three-dimensional, experimentally reconstructed single-phase polycrystals and directly characterize the grain boundary networks

in terms of grain size distribution, number of neighboring grains versus grain size, mean curvature and width of grains, growth rate of individual grains, the GBCD and the GBED. For a FEM analysis, additional procedures (such as triangle quality enhancement, control of numbers of triangular elements and volumetric element meshing from the enhanced surface mesh) are required. These issues will be explored in the future.

Acknowledgments

This work was supported in part by the MRSEC at CMU under NSF Award No DMR-0520425. S-B L acknowledges support for the analysis and modeling provided in part by the 2013 Research Fund of the UNIST (Ulsan National Institute of Science and Technology). The authors are grateful to Dr M Uchic at Air Force Research Laboratory (AFRL) for supplying serial section orientation data of IN100.

References

[1] Smith C 1948 *Trans. Am. Inst. Min. Metall. Eng.* **175** 15
[2] Williams W and Smith C 1952 *Trans. Am. Inst. Min. Metall. Eng.* **194** 755
[3] Smith C 1953 *Acta Metall.* **1** 295
[4] von Neumann J 1952 *Metal interfaces* ed C Herring (Cleveland, OH: American Society or Metals) 108
[5] Mullins W 1956 *J. Appl. Phys.* **27** 900
[6] MacPherson R D and Srolovitz D J 2007 *Nature* **446** 1053
[7] Underwood E 1970 *Quantitative Stereology* (New York: Addison-Wesley)
[8] Saylor D M, El-Dasher B S, Rollett A D and Rohrer G S 2004 *Acta Mater.* **52** 3649
[9] Rohrer G S, Saylor D M, El-Dasher B S, Adams B L, Rollett A D and Wynblatt P 2004 *Z. Metallkd.* **95** 197
[10] Kim C-S, Rollett A D and Rohrer G S 2006 *Scr. Mater.* **54** 1005
[11] Rohrer G S, Randle V, Kim C-S and Hu Y 2006 *Acta Mater.* **54** 4489
[12] Bozzolo N, Sawina G, Gerspach F, Sztwiertnia K, Rollett A D and Wagner F 2007 *Mater. Sci. Forum* **863** 558–9
[13] Rohrer G S 2007 *JOM* **59** 38
[14] Papillon F, Rohrer G S and Wynblatt P 2009 *J. Am. Ceram. Soc.* **92** 3044
[15] Wang W G, Zhou B X, Rohrer G S, Guo H and Cai Z 2010 *Mater. Sci. Eng.* A **527** 3695
[16] Dillon S J and Rohrer G S 2009 *J. Am. Ceram. Soc.* **92** 1580
[17] Rohrer G S, Li J, Lee S-B, Rollett A D, Groeber M and Uchic M D 2010 *Mater. Sci. Technol.* **26** 661
[18] Li J, Dillon S J and Rohrer G S 2009 *Acta Mater.* **57** 4304
[19] Rowenhorst D, Lewis A C and Spanos G 2010 *Acta Mater.* **58** 5511
[20] Wakai F, Enomoto N and Ogawa H 2000 *Acta Mater.* **48** 1297
[21] Quey R, Dawson P R and Barbe F 2011 *Comput. Methods Appl. Mech. Eng.* **200** 1729
[22] Barbe F and Quey R 2007 18 ème Congrès Français de Mécanique, Grenoble, août 2007 27–31
[23] Könke C, Eckardt S, Häfner S, Luther T and Unger J 2010 *Int. J. Multi. Comput. Eng.* **8** 17
[24] Bhandari Y, Sarkar S, Groeber M, Uchic M, Dimiduk D M and Ghosh S 2007 *Comput. Mater. Sci.* **41** 222
[25] Ghosh S, Bhandari Y and Groeber M 2008 *Comput.-Aided Des.* **40** 293
[26] Miranda A C O, Martha L F, Wawrzynek P A and Ingraffea A R 2009 *Eng. Comput.* **25** 207
[27] Dillard S E, Bingert J F, Tohma D and Hamann B 2007 *IEEE Trans. Vis. Comput. Graphics* **13** 1528
[28] Brakke K A 1992 *Exp. Math.* **1** 141
[29] Kuprat A, Khamayseh A, Geroge D and Larkey L 2001 *J. Comput. Phys.* **172** 99
[30] Moore R H, Rohrer G S and Saigal S 2009 *Eng. Comput.* **25** 221
[31] Gilbert E 1962 *Ann. Math. Stat.* **33** 958

[32] Mahin K W, Hanson K and Morris J W Jr 1980 *Acta Metall.* **28** 443
[33] Rickman J M, Tong W S and Barmak K 1997 *Acta Mater.* **45** 1153
[34] Wu Z and Sullivan J M Jr 2003 *Int. J. Numer. Methods Eng.* **58** 189
[35] Lorensen W E and Cline H E 1987 *Comput. Graphics* **21** 163
[36] Anderson M P, Srolovitz D J, Grest G S and Shani P S 1984 *Acta Metall.* **32** 783
[37] Anderson M P, Grest G S and Srolovitz D J 1985 *Scr. Metall.* **19** 225
[38] Anderson M P, Grest G S and Srolovitz D J 1989 *Phil. Mag.* B **59** 293
[39] Holm E A, Glazier J A, Srolovitz D J and Grest G S 1991 *Phys. Rev.* A **43** 2662
[40] Wright S I and Larsen R J 2002 *J. Microsc.* **205** 245
[41] Lee S-B, LeDonne J E, Lim C V S, Beyerlein I J and Rollett A D 2012 *Acta Mater.* **60** 1747
[42] Sintay S D and Rollett A D 2012 *Modelling Simul. Mater. Sci. Eng.* **20** 075005
[43] Rohrer G S 2011 *J. Mater. Sci.* **46** 5881
[44] Harker D and Parker E R 1945 *Trans. Am. Soc. Metals* **34** 156
[45] Chandross M and Holm E A 2010 *Metall. Mater. Trans.* A **41A** 3018