



Graph Coloring Approach to Mesh Generation in Multiphase Media with Smooth Boundaries

Siddhartha Srivastava* and Veera Sundararaghavan†
University of Michigan, Ann Arbor, Michigan 48109-2140

DOI: 10.2514/1.J058357

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

Nomenclature

$D_i(f)$	=	data cost contribution from i th vertex with labeling f
f	=	array representation of labeling
$G(V, E)$	=	undirected graph with vertices V and edges E
$g(\mathbf{n})$	=	distance measure in direction \mathbf{n}
$H(f)$	=	energy/cost function for labeling f
$I(v_i)$	=	pixel/voxel-based coloring of i th vertex
$\text{int}(\cdot)$	=	interior of a set (\cdot)
L	=	set of labels
M	=	mode of the distribution of area/volume
\mathcal{P}_h	=	partition of domain
P_i	=	indexed polygon
$\text{Vol}(\cdot)$	=	area/volume of (\cdot)
$w_{i,j}(\cdot)$	=	connection weight for edge (v_i, v_j)
α	=	scalar parameter
Δs	=	length/area of the edge/face (of element)
$\{\phi\}$	=	null set

I. Introduction

STRUCTURAL metallic materials used in the aerospace industry (e.g., Aluminum 2000 and 7000 series, advanced titanium alloys) are composed of aggregates of crystals. With the emergence of the paradigm of integrated computational materials engineering (ICME) [1], multiscale design/optimization approaches for tailoring engineering properties of materials through controlled microstructure [2,3] are of great interest. A multiscale approach for the design of turbine blades is presented in Fig. 1 to illustrate stress variation in the macroscopic and microscopic scale. Such a simulation involves solving for microstructure-dependent properties for macroscale analysis, which in turn requires meshing at the microstructural level that captures the grain size and shape features [4]. Meshing of three-dimensional (3D) microstructures to conserve

such grain features is of immense interest as these play an important role in processes such as localization [5] and fracture [6].

Computational tools, like finite element (FE) methods, are almost ubiquitously present in the form of both commercial and private software. In the realm of polycrystalline materials, numerical procedures generally require the knowledge of the microstructure, which is often provided through experimental methods such as Electron Back Scatter Diffraction (EBSD). EBSD provides the spatial distribution of crystallographic orientations of different grains and can be used to estimate quantities like average misorientation. A complete review of experimental and computational EBSD techniques is provided in [7]. From the computational perspective, these experimental images offer information about the spatial distribution of phases in a pixelated format for two-dimensional (2D) or voxelated format for 3D. These data also provide a convenient form of a mesh that consists of uniform quadrilateral/hexahedral elements. An open source package is developed in [8] for generation of voxelated representations of serial sectioned EBSD maps. However, voxelation leads to a stepped, block-like representation of otherwise smooth boundaries. This affects the quality of computational simulation through the introduction of spurious stresses [9]. There are other drawbacks in using structured meshes of quadrilateral/hexahedral elements for 2D/3D elements as well; for example, the boundary length/area does not converge to the correct value with refinement.

Unstructured grids where element surfaces conform with the phase boundaries offer a better approximation of the spatial morphology. An unstructured meshing scheme, which is widely employed, generates objects in the lowest dimension and then re-uses these objects as seeds for generating higher-order objects. More precisely, nodes (0-dimensional objects) are generated and are connected using lines (1-dimensional objects), where more points are sampled. Faces (2D objects) are created by joining these lines and are meshed using triangles with edge points as seeds. Then, these faces are connected to create a grain structure (3D objects) and are meshed with tetrahedrons using face triangles as seeds. This procedure allows for a smoother phase boundary and improves the estimation of quantities like phase boundary energy. This procedure is also implemented for mesh generation in polycrystalline materials in an open source software package developed in Ref. [10]. However, this procedure can be computationally expensive as it requires re-meshing for every new experimental image.

A new technique for optimal partitioning of an arbitrary unstructured mesh into different phases is proposed in this paper. In practice, partitioning the mesh means assigning an index to each

Presented as Paper 2019-0966 at the AIAA Science and Technology Forum and Exposition 2019, San Diego, CA, 7–11 January 2019; received 5 February 2019; revision received 25 July 2019; accepted for publication 5 August 2019; published online 30 August 2019. Copyright © 2019 by Veera Sundararaghavan. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-385X to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Ph.D. Candidate, Aerospace Engineering.

†Associate Professor, Aerospace Engineering.

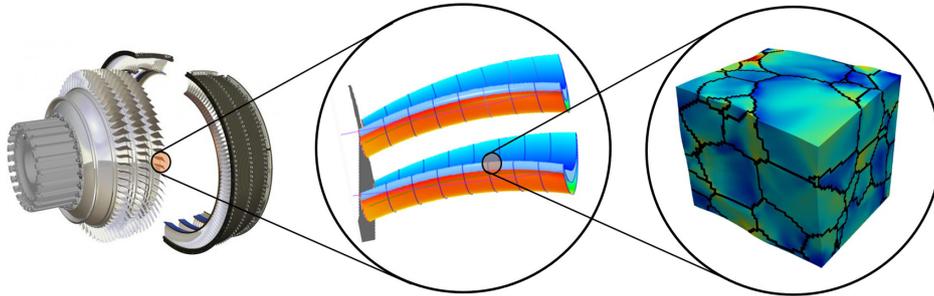


Fig. 1 Multiscale design of turbine blades for aircraft engines.

element where elements with similar indexes form a single phase. Thus, any experimental image can be superimposed on a generic unstructured FE mesh by effectively partitioning the mesh based on voxelated data. Two conditions are expected to be satisfied. First, the phase index assignment to the FE mesh should be as close as possible to the experimental voxel data. Second, the phase index assignments should be such that the boundaries are as smooth as possible. We solve this problem using graph partitioning (GP) theory, and this is the novel contribution of this paper.

It has been shown that GP is an NP-hard problem, thus achieving an efficient exact solution is difficult unless $P = NP$ [11]. Seminal contributions in the development of the GP algorithm are made in [12–14]. Spectral methods (for instance, see [15]) are often used. Recent advances in this field have been reviewed in [16]. Improved results are achieved with global methods, such as multilevel approaches, which operate in stages with refinement. METIS and KaHIP are good open source graph partitioners that use multilevel methods and are developed in [17,18], respectively. Because GP is often studied as a combinatorial optimization problem, combinatorial optimization methods, in particular, metaheuristic methods such as simulated annealing, are also commonly used. One of the most widely used algorithms specialized for the Potts energy model was developed in [19]. A comparative study of some of these algorithms is provided in [20].

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

II. Problem Formulation

Some definitions are presented for ease of understanding and to facilitate a common vocabulary. The formulation for both 2D and 3D mesh generation is presented in this section.

Definition 1 (FE mesh): Given a closed bounded polyhedral domain $\Omega \in \mathbb{R}^3$, we can associate a finite partition \mathcal{P}_h of Ω into polyhedrons P_i (indexed with i) such that

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

such that

- 1) $\text{int}(P_i) \neq \emptyset \forall P_i \in \mathcal{P}_h$
- 2) $\text{int}(P_i) \cap \text{int}(P_j) = \emptyset \forall P_i, P_j \in \mathcal{P}_h$ s.t. $P_i \neq P_j$
- 3) if $F = P_i \cap P_j \neq \emptyset$ for some $P_i, P_j \in \mathcal{P}_h$ and $P_i \neq P_j$, then F is either a whole face (polygon), a whole edge (line segment), or a node (point) of the polygons P_i and P_j

- 4) $h = \max_{P_i \in \mathcal{P}_h} h_i$, where h_i denotes the longest Euclidean distance between two points of P_i .

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

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

Definition 2 (graph and undirected graph): A graph (G) is a pair of sets (V, E) , where V is the set of vertices and E is the set of edges/connections. For each element $e \in E$ there is an associated order pair (x, y) for some $x, y \in V$, that is, $E \subseteq V \times V$. The graph $G = (V, E)$ is undirected if $(x, y) \in E \Rightarrow (y, x) \in E \quad \forall x, y \in V$.

Remark: To avoid confusion with the edges of a mesh, from here on, the term *connection* is preferred for graphs.

The mesh for multiphase materials is defined by assigning a phase to each element of the partition. This assignment or labeling, as it will be referred to in the rest of the paper, is done using an undirected graph (G) . The graph G is embedded in the mesh, \mathcal{P}_h , by placing a single vertex, v_i , in the interior of each P_i . Without loss of generality, the location of v_i can be determined as the centroid of the polygon P_i . It ensures that $v_i \in \text{int}(P_i)$ when P_i is convex (generally true for all elements of FE meshes). A connection is introduced between two vertices v_i and v_j if and only if P_i and P_j share a common face. In case of 2D meshes, this connection is introduced if and only if P_i and P_j share a common edge. Furthermore, each connection is endowed with a weight determined by the area and the unit normal of the shared face. Consequently in 2D meshes, the weight is determined by the length and the unit normal of the shared edge. In particular, if \mathbf{n} and Δs are the unit normal and the area (length) of the face (edge), respectively, then weight w_i of the connection e_i is given as $w_i = g(\mathbf{n})\Delta s$. The function g is restricted to be strictly positive and symmetric, that is, $g > 0$ and $g(\mathbf{n}) = g(-\mathbf{n})$. This construction is illustrated in Fig. 2, where a connection between 2 representative elements, **A** and **B**, is introduced with a weight determined by the unit normal \mathbf{n} and length of the edge Δs .

Definition 3 (multiway k -cut problem [21]): Given an undirected graph, $G = (V, E)$, edge(connections) weights, $w: e_i \rightarrow \mathbb{R}^+ \forall e_i \in E$, and a set of terminals $S = \{s_1, s_2, \dots, s_k\} \subset V$, a multiway cut is a set of edges(connections) E' that leaves each of the terminals in a separate component. The cost of the multiway cut is defined as the sum of the weights of the edges(connections) in E' . The goal of the multiway cut problem is to minimize this cost.

Remarks:

- 1) For $k > 3$, the multiway cut problem is NP-hard

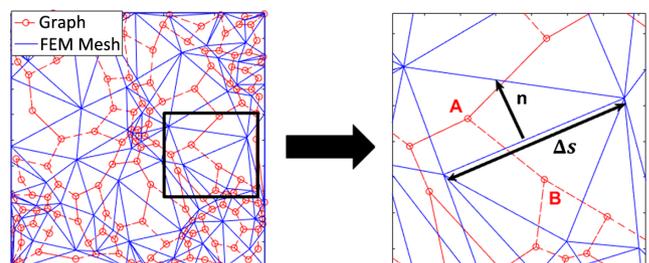


Fig. 2 Construction of the graph $G = (V, E)$ from a FE mesh.

2) This problem can equivalently and rather more intuitively be presented as a labeling problem where each vertex in V is assigned a label $f_i \in L$, where $|L| = k$. The multiway cut is defined as a set of edges (connections) between vertices with different labels.

In this work, labels are identified as different phases. For a particular choice of labels f (i.e., each v_i is assigned the label f_i), and the corresponding multiway cut E' , a Potts-like energy H of is defined using Eq. (1). The first term in this form is referred to as the data term, and the second term is referred to as the smooth term.

$$H(f) = \sum_{v_i \in V} D_i(f) + \sum_{(v_i, v_j) \in E'} w_{i,j}(f) \quad (1)$$

The image information is encoded in the data term using a phenomenological form. Let I be a function that evaluates the label (phase information) from the image data (pixel/voxel). For instance, $I(v_i) = r$ means v_i belongs to the pixel region with phase index r . Additionally, let the function $\text{Vol}(P_i)$ evaluate the area/volume of the i th element in case of 2D/3D meshes, with M representing mode of the distribution. As an ansatz, D_i assumes the form of Eq. (2), where α is a positive real number and is treated as a control parameter. It is worth observing that the minimization of this term leads to preservation of phase structure of the input data.

$$D_i(f) = \begin{cases} 0 & \text{if } I(v_i) = f_i \\ \alpha(1 - e^{-\text{Vol}(P_i)/M}) & \text{if } I(v_i) \neq f_i \end{cases} \quad (2)$$

The smooth term, as the name suggests, penalizes any roughness of phase boundaries. In this work, it is specialized to the form presented in Eq. (3), where \mathbf{n} represents the normal to the shared edge between i th and j th element. The action of the second term can be better understood by taking $g(\mathbf{n}) = 1$ for all \mathbf{n} . In this special case, the second term reduces to the total length of the boundary between each component of the k -way partition, that is, length of the phase boundary. Minimization of this term leads to the minimization of the phase boundary length and renders a smoothing effect. The total energy is minimized using the alpha expansion method discussed in [19] by solving the equivalent multiway k -cut problem. The data structure and the pseudo-code for implementing this method are presented in the Appendix.

$$w_{i,j}(f) = \begin{cases} 0 & \text{if } f_i = f_j \\ g(\mathbf{n})\Delta s & \text{if } f_i \neq f_j \end{cases} \quad (3)$$

Simultaneous minimization of the two energy terms gives rise to Pareto optimal solutions, that is, solutions where both terms cannot be simultaneously decreased any further. This means that there exists a labeling such that any change in it either leads to deviation from the input image or introduces roughness in grain boundary. The solution is selected based on the choice of α in Eq. (2). This phenomenon is illustrated in the next section by means of an example.

III. Results and Discussion

The salient features of the algorithm are illustrated using two examples. In the first example, polycrystalline materials are used with the grains representing the different phases of the material. Next, a case study of woven fibers composite is considered where the fiber

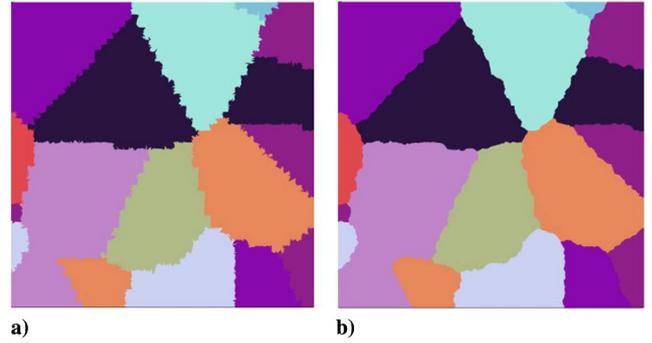


Fig. 4 Grain identification in a mesh using a 50×50 pixel image: a) Pixel-based labeling, b) Multiway k -cut-based labeling.

and the matrix are identified as the two distinct phases. In polycrystalline materials, the smooth boundary is an artifact of microstructure evolution, which is governed by surface energy minimization. According to the theory discussed in the previous section, the naive labeling based on the pixel location results in rough grain boundaries. The energy minimization problem is not required to estimate this labeling. In fact, it can be uniquely determined by the relation, $f_i = I(v_i)$, which was treated as the basis for formulating the data cost. This grain boundary is smoothed by minimization of energy form given in Eq. (1). This process is illustrated in Fig. 3 for a microstructure image with 200×200 pixels. A triangular mesh with 30,625 nodes and 60,547 elements is, first, labeled based on pixel positions. The boundaries are then smoothed by solving the multiway k -cut problem on the embedded graph containing 90,470 connections with the data parameter chosen as $\alpha = 0.002$ and the metric function chosen as $g(\mathbf{n}) = 1$. The same procedure can be carried out for low-resolution images where pixel-based labeling may result in a very rough grain boundary. Grain identification for mesh elements using pixel-based and multiway k -cut based is presented for a low-resolution image of a pixel size of 50×50 in Fig. 4. It is observed that the smoothing effect is less prominent (in comparison to the high-resolution image) due to the pixelated nature of the base image.

The dependence of partitioning on the data term is shown in Fig. 5. The grain boundary length ratio (i.e., the nondimensional length of the grain boundary with respect to the unit cell) is calculated with respect to labelings for different data cost. Three regions are observed based on the value of the α defined in Eq. (2). The first region corresponds to the case when higher weight is given to the minimization of length (low value of α). In this case, the addition of data cost is preferred over the addition of surface energy. On the contrary, labeling is governed by data cost in the third region. Therefore, this region is plagued with rough boundaries and an overall higher estimate of grain boundary length/area. Practical meshes with smooth boundaries and a minimal loss of grain structure are offered in the second region.

A comparison of the performance of this mesh with that of a uniform mesh with pixel-based labeling and an exact mesh (conformal mesh) is presented. A bicrystal with an incident angle of 60° between isotropic components is studied in uniaxial loading. For simplicity, the FE simulations are presented for linear-elastic deformation assuming plane strain. The distribution of Von-mises stress is presented for the different meshes in Fig. 6. The grain

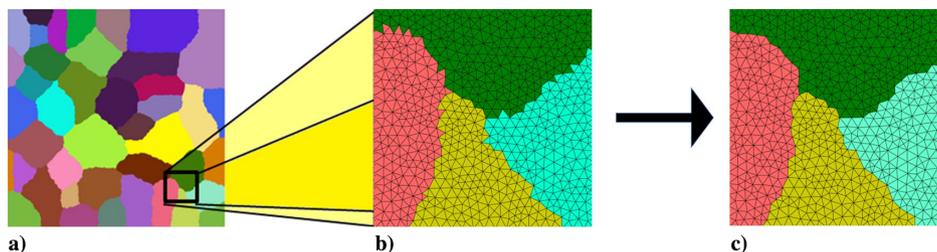


Fig. 3 An illustration of labeling procedure: a) Microstructure image, b) Pixel-based labeling, c) Smoothed mesh using multiway k -cut-based labeling.

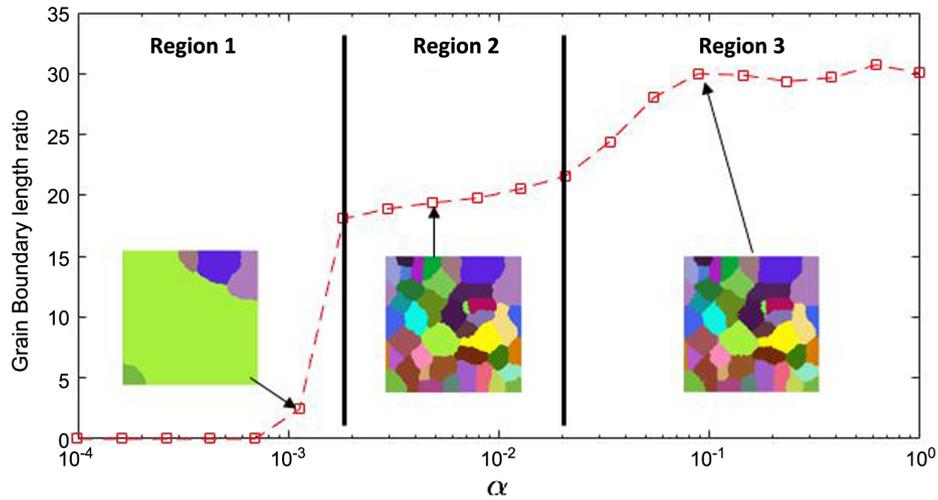


Fig. 5 Dependence of graph partitioning on the data term in energy.

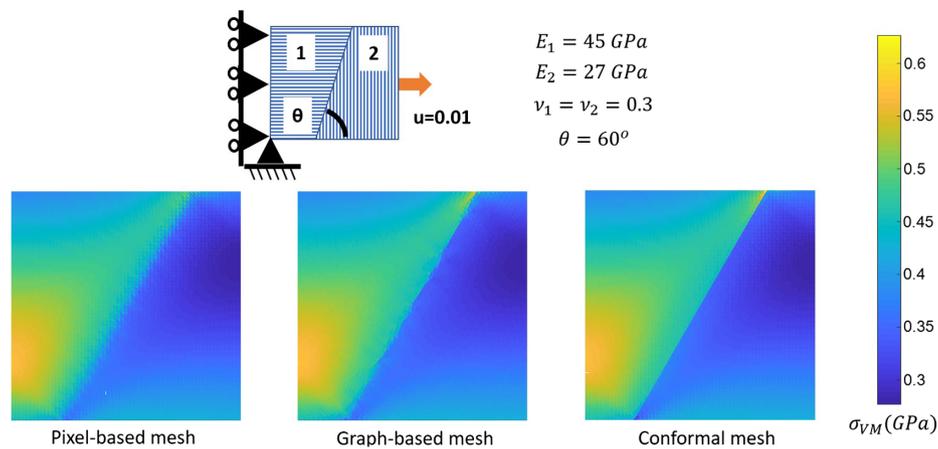


Fig. 6 Comparative study of FE simulation results for a bicrystal in plane strain (linear elastic) using pixel-based (quad mesh), graph-based (triangle mesh with smoothing), and conformal mesh.

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

Additive manufacturing techniques, like selective laser melting (SLM), electron beam melting (EBM), and shaped metal deposition (SMD), often result in microstructure with elongated grains [22–24]. These structures can be better captured using the characteristic alignment of the grains. Motivated by the elliptical shape of grains, a non-Euclidean distance measure can be used to favor or penalize certain directions. A useful class of such distance measures is a Riemannian norm. For this purpose, the function g can be specialized as Eq. (4).

$$g(v) = \sqrt{v^T \cdot D \cdot v} \quad (4)$$

where D is a positive definite matrix. In general, polycrystalline materials with nonequiaxed morphology show smoother grain boundaries in the normal/transverse direction (ND/TD). This directional smoothing is achieved by penalizing the length of a line element in an appropriate direction using the metric presented in Eq. (4). An example of elongated microstructures is presented in Fig. 7. It is observed that a metric with $D_{11} = 1$, $D_{22} = 10$, and $D_{12} = D_{21} = 0$ minimizes the kinks in the normal direction. This procedure can be specialized to many other kinds of morphologies using the matrix, D . The iso-surfaces for the distance function $g(\mathbf{n})$

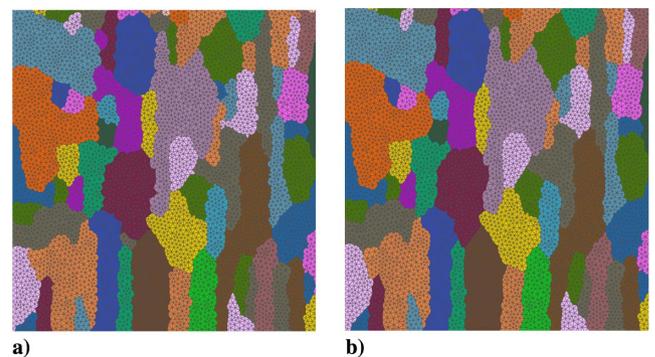


Fig. 7 Grain identification for elongated polycrystals using a) Euclidean metric and b) using Riemannian metric.

for different choices of matrix D are presented as an illustration in Fig. 8. Based on the theory of Wulff construction, the iso-surface represents the grain shape ([25]).

As suggested in Sec. III, this procedure can be easily extended to 3D meshes. For the purpose of illustration, a $64 \times 64 \times 64$ voxelated image of a microstructure is superimposed on an FE mesh with tetrahedral elements. The sample solution is presented in Fig. 9 for a mesh with approximately 3×10^5 elements. This smoothing procedure is also illustrated for a woven fiber composite system. A voxelated mesh and a tetrahedral (smooth) mesh with approximately 10^6 nodes are presented in Fig. 10. A comparative study of convergence in volume and surface area is also presented in Fig. 10c.

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

Fig. 8 Illustration of various mythologies based on the metric tensor, D .

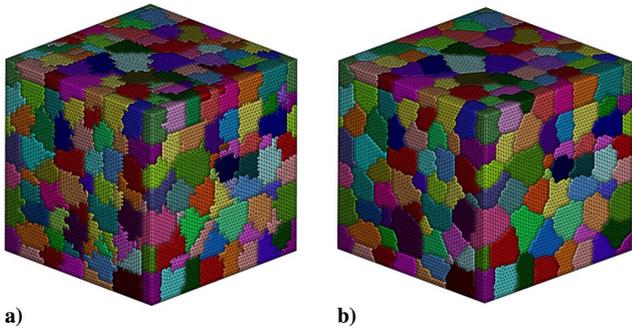


Fig. 9 Grain identification in 3D meshes using a $64 \times 64 \times 64$ voxel base image: a) Vertex-position-based labeling. b) Multiway k -cut-based labeling.

Different voxelated meshes are used with “Vox $< n >$ ” representing a mesh with $(n + 1)^3$ nodes. The fiber composite with Vox100 mesh is shown in Fig. 10a with the tetrahedral mesh (Tet100) in Fig. 10b. The normalization is done with respect to the analytical values of respective quantities. It is observed that the estimate of fiber volume is almost consistent in all the meshes, but the estimate of fiber surface area is improved in the tetrahedral mesh.

The time complexity of the algorithm is experimentally studied for 2D and 3D test cases. Ten instances of meshes with a similar number of elements are labeled using GP. The total average time for GP using alpha expansion is calculated for the 2D triangle and 3D tetrahedral elements. An almost linear trend is observed in Fig. A1 (log-log scale is shown but note that the simulation time scales linearly with number of elements). This is faster in comparison to the experimental results presented in [26]. Faster heuristics are seen for the 2D mesh in

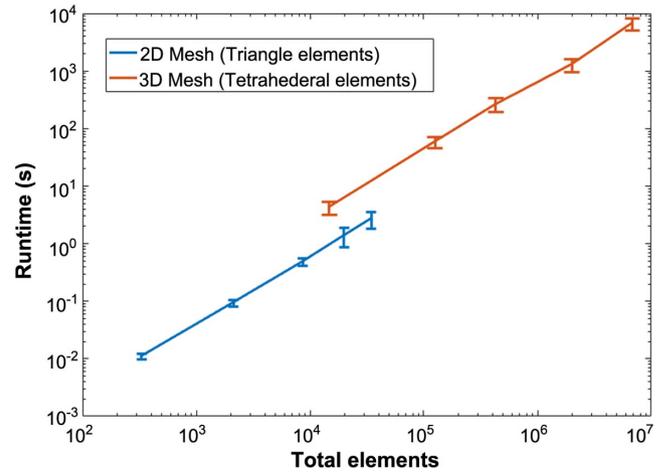


Fig. A1 Time complexity analysis of the mesh generation procedure.

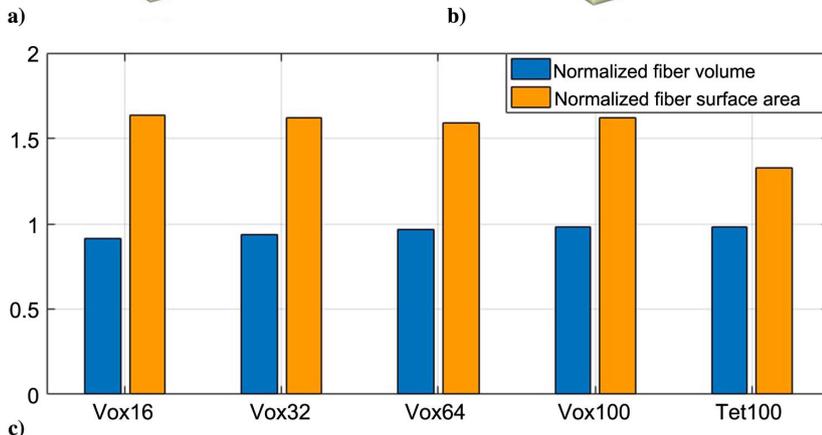
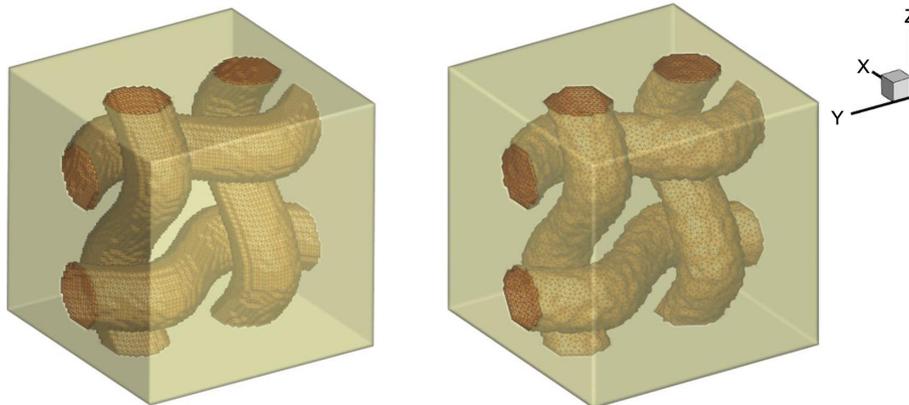


Fig. 10 Mesh generation for woven fiber composite. a) Voxelated mesh with $10^2 \times 10^2 \times 10^2$ elements. b) Tetrahedral mesh with multiway k -cut-based labeling. c) Comparison of fiber volume and surface area of various voxelated meshes and a tetrahedral (smoothened) mesh.

comparison to 3D for the same number of elements. This is because the embedded graph in the 2D case has a planar topology and graph-cut algorithm is known to be more efficient in these cases.

IV. Conclusions

Based on the integrated computational materials engineering paradigm, there has been increasing interest in the use of microstructure modeling tools for optimization of aerospace materials manufacturing processes. Experimental evaluation of the spatial distribution of multiphase materials provides images in a voxelated/pixelated format. However, such a format does not accurately capture features like grain surface area and curvature that are important for modeling fracture. In addition, FE simulation on voxelated microstructures may lead to the prediction of spurious stresses due to the stepped, block-like representation of otherwise smooth boundaries. In this paper, a theory has been developed for conversion of as-measured voxelated microstructure to an unstructured grid with a smoother representation of boundaries. This problem is solved using the graph partitioning theory, and this is the novel contribution of this paper. This procedure smoothens the grain boundaries using a Potts energy model while optimally preserving the grain label information of the initial experimental data. The trade-off between smoothing and data preservation is controlled using a user-controlled parameter. The performance is tested for both equiaxed and nonequiaxed morphologies using Riemannian metric measures. It is observed that smoother grain boundaries in the normal/transverse direction (ND/TD) can be realized using this approach. The methodology has an efficient runtime and is easily extendable to 3D structures like woven fiber composites and 3D polycrystalline material as demonstrated in this work.

Acknowledgment

V.S. would like to acknowledge the Air Force Office of Scientific Research, contract FA9550-18-1-0091, for financial support.

Appendix: Pseudo-Code and Data Structures

Here, we present the data structure and the pseudo-code for the mesh labeling algorithm. There are two kinds of data that are required in this algorithm: 1) FE mesh data and 2) image data. It is recommended to calculate and save all the variables as defined below at the time of mesh generation and be passed on to mesh labeling algorithm. The FE mesh data structure is preprocessed to include the following variables:

- 1) *Dimension*: Value is 2 for 2D and 3 for 3D.
- 2) *MeshSize*: The x -dimension of mesh is in the range $[0, MeshSize(1)]$ and the y -dimension is in the range $[0, MeshSize(2)]$. For 3D mesh, the z -direction is in the range $[0, MeshSize(3)]$.
- 3) *Coordinate*: Location of each node of the mesh
- 4) *Nnode*: Number of nodes
- 5) *Connectivity*: Contains tuples with nodes of each element
- 6) *Nelement*: Number of elements
- 7) *Neighbor*: In 3D meshes this data contains each pair of elements that share a face. In case of 2D, it contains each pair of elements that share an edge.
- 8) *Npair*: Number of *Neighbors*
- 9) *ElementVolume*: Defined only for 3D meshes and contains the volume of each element
- 10) *Face*: Defined only for 3D meshes and contains tuples with nodes of the shared faces (indexed as the *Neighbor* data)
- 11) *FaceArea*: Area of face defined in *Face* data
- 12) *FaceNormal*: Unit normal of face defined in *Face* data

Algorithm 1: Set-up Data cost (2D/3D)

```

1:  $M \leftarrow$  Mesh data ▷ Import mesh data
2:  $I \leftarrow$  Image data ▷ Import Experimental image data
3:  $Unary \leftarrow$  zeros( $M.Nlabel, M.Nelement$ ) ▷ Initialize Data cost variable as an matrix of zeros
4: if  $M.Dimension == 3$  then ▷ 3D Case
5:    $temp \leftarrow$  mode( $M.ElementVolume$ ) ▷ Evaluate mode of the distribution of element volume
6:    $Factor \leftarrow \alpha(1 - \exp(-M.ElementVolume/temp))$  ▷ Evaluate Eq. (2)
7:   for  $index1 \leftarrow 1$  to  $M.Nelement$  do ▷ Loop over each element
8:      $xpos \leftarrow$  mean( $M.Coordinate(M.Connectivity(index1, :), 1)$ ) ▷ Coordinates of center of the element
9:      $ypos \leftarrow$  mean( $M.Coordinate(M.Connectivity(index1, :), 2)$ )
10:     $zpos \leftarrow$  mean( $M.Coordinate(M.Connectivity(index1, :), 3)$ )
11:     $nx =$  floor( $xpos \times I.ImageSize(1) / M.MeshSize(1)$ ) + 1 ▷ Location of center with respect to image grid data
12:     $ny =$  floor( $ypos \times I.ImageSize(2) / M.MeshSize(2)$ ) + 1
13:     $nz =$  floor( $zpos \times I.ImageSize(2) / M.MeshSize(3)$ ) + 1
14:    for  $index2 \leftarrow 1$  to  $I.Nlabel$  do ▷ Loop over each Label value
15:      if  $I.LabelValues(index2) == I.LabelGridData(nx, ny, nz)$  then
16:         $Unary(index2, index1) \leftarrow 0$  ▷ No cost if the Label matches the image data
17:      else
18:         $Unary(index2, index1) \leftarrow Factor(index1)$  ▷ Add cost determined by Eq. (2)
19:      end if
20:    end for
21:  end for
22: else
23:    $temp \leftarrow$  mode( $M.ElementArea$ ) ▷ Evaluate mode of the distribution of element area
24:    $Factor \leftarrow \alpha(1 - \exp(-M.ElementArea/temp))$  ▷ Evaluate Eq. (2)
25:   for  $index1 \leftarrow 1$  to  $M.Nelement$  do ▷ Loop over each element
26:      $xpos \leftarrow$  mean( $M.Coordinate(M.Connectivity(index1, :), 1)$ ) ▷ Coordinates of center of the element
27:      $ypos \leftarrow$  mean( $M.Coordinate(M.Connectivity(index1, :), 2)$ )
28:      $nx =$  floor( $xpos \times I.ImageSize(1) / M.MeshSize(1)$ ) + 1 ▷ Location of center with respect to image grid data
29:      $ny =$  floor( $ypos \times I.ImageSize(2) / M.MeshSize(2)$ ) + 1
30:     for  $index2 \leftarrow 1$  to  $I.Nlabel$  do ▷ Loop over each Label value
31:       if  $I.LabelValues(index2) == I.LabelGridData(nx, ny, nz)$  then
32:          $Unary(index2, index1) \leftarrow 0$  ▷ No cost if the Label matches the image data
33:       else
34:          $Unary(index2, index1) \leftarrow Factor(index1)$  ▷ Add cost determined by Eq. (2)
35:       end if
36:     end for
37:   end for
38: end if
39: return  $Unary$ 

```

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

```

1:  $M \leftarrow$  Mesh data ▷ Import mesh data
2:  $I \leftarrow$  Image data ▷ Import Experimental image data
3:  $Pairwise \leftarrow$  zeros( $N_{element}$ ) ▷ Initialize Smooth cost variable as a square matrix of zeros
4: if  $M.Dimension == 3$  then ▷ 3D Case
5:   for  $index \leftarrow 1$  to  $M.N_{pair}$  do
6:      $paircost \leftarrow g(M.FaceNormal(index)) \times M.FaceArea(index)$  ▷ Evaluate Eq. (3)
7:      $Pairwise(M.Neighbor(index, 1), M.Neighbor(index, 2)) \leftarrow paircost$  ▷ Update  $Pairwise$  variable
8:   end for
9: else ▷ 2D Case
10:  for  $index \leftarrow 1$  to  $M.N_{pair}$  do
11:     $paircost \leftarrow g(M.EdgeNormal(index)) \times M.EdgeArea(index)$  ▷ Evaluate Eq. (3)
12:     $Pairwise(M.Neighbor(index, 1), M.Neighbor(index, 2)) \leftarrow paircost$  ▷ Update  $Pairwise$  variable
13:  end for
14: end if
15: return  $Pairwise$ 

```

Algorithm 3: Using GCO Library for estimating the labeling

```

1: Input:  $Unary, Pairwise, LabelValues, NLabel, Nelement$ 
2:  $h \leftarrow$  GCO_Create( $NElement, NLabel$ )
3: GCO_SetDataCost( $h, Unary$ );
4:  $Labelcost \leftarrow$  ones( $NLabel$ ) - Id( $NLabel$ )
5: GCO_SetSmoothCost( $h, Labelcost$ );
6: GCO_SetNeighbors( $h, Pairwise$ );
7: GCO_Expansion( $h$ );
8:  $Label \leftarrow$  LabelValues(GCO_GetLabeling( $h$ ));
9: return  $Label$ 
10: end

```

13) *ElementArea*: Defined only for 2D meshes and contains the area of each element

14) *Edge*: Defined only for 2D meshes and contains tuples with nodes of shared edges (indexed as the *Neighbor* data)

15) *EdgeLength*: Length of each edge defined in *Edge* data

16) *EdgeNormal*: Unit normal of each edge defined in *Edge* data

The image data structure includes the following variables:

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

2) *XGridData*: Contains the x -coordinate value of the voxelated/pixelated image with size determined by *ImageSize*.

3) *YGridData*: Contains the y -coordinate value of the voxelated/pixelated image with size determined by *ImageSize*.

4) *ZGridData*: Defined only for 3D image data. Contains the z -coordinate value of the voxelated/pixelated image with size determined by *ImageSize*.

5) *LabelGridData*: Label values at each grid point.

6) *NLabel*: Number of labels

7) *LabelValues*: Contains the label values of enumerated from 1 to *NLabels*

As discussed in the paper, this method uses an energy minimization approach. Pseudo-code for estimation of data cost and smooth cost is presented in Algorithms 1 and 2, respectively. The energy minimization is carried out using the GCO library developed in [19]. The pseudo-code for using this library is presented in Algorithm 3:

References

- [1] Sangid, M., Matlik, J. F., Keskin, A., Thacker, B. H., Bichon, B. J., Ball, D. L., Engelstad, S. P., Ward, C., Venkatesh, V., Kim, H. A., et al., "Integrating ICME Practices into Design Systems and Structural Analysis," *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0874, 2017. doi:10.2514/6.2017-0874
- [2] Sundararaghavan, V., and Zabarav, N., "A Statistical Learning Approach for the Design of Polycrystalline Materials," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, Vol. 1, No. 5, 2009, pp. 306–321. doi:10.1002/sam.v1:5
- [3] Acar, P., and Sundararaghavan, V., "Utilization of a Linear Solver for Multiscale Design and Optimization of Microstructures," *AIAA Journal*, Vol. 54, No. 5, 2016, pp. 1751–1759. doi:10.2514/1.J054822
- [4] Sun, S., and Sundararaghavan, V., "A Probabilistic Crystal Plasticity Model for Modeling Grain Shape Effects Based on Slip Geometry," *Acta Materialia*, Vol. 60, Nos. 13–14, 2012, pp. 5233–5244. doi:10.1016/j.actamat.2012.05.039
- [5] Sun, S., and Sundararaghavan, V., "A Peridynamic Implementation of Crystal Plasticity," *International Journal of Solids and Structures*, Vol. 51, No. 19, 2014, pp. 3350–3360. doi:10.1016/j.ijsolstr.2014.05.027
- [6] Sundararaghavan, V., and Srivastava, S., "MicroFract: An Image Based Code for Microstructural Crack Path Prediction," *SoftwareX*, Vol. 6, 2017, pp. 94–97. doi:10.1016/j.softx.2017.04.002
- [7] Schwartz, A. J., Kumar, M., Adams, B. L., and Field, D. P., *Electron Backscatter Diffraction in Materials Science*, Springer, New York, 2014, pp. 19–30. doi:10.1007/978-1-4757-3205-4_2
- [8] Groeber, M. A., and Jackson, M. A., "DREAM. 3D: A Digital Representation Environment for the Analysis of Microstructure in 3D," *Integrating Materials and Manufacturing Innovation*, Vol. 3, No. 1, 2014, pp. 56–72. doi:10.1186/2193-9772-3-5
- [9] Fang, G., Said, B. E., Ivanov, D., and Hallett, S. R., "Smoothing Artificial Stress Concentrations in Voxel-Based Models of Textile Composites," *Composites Part A: Applied Science and Manufacturing*, Vol. 80, Jan. 2016, pp. 270–284. doi:10.1016/j.compositesa.2015.10.025
- [10] Quey, R., Dawson, P., and Barbe, F., "Large-Scale 3D Random Polycrystals for the Finite Element Method: Generation, Meshing and Remeshing," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, Nos. 17–20, 2011, pp. 1729–1745. doi:10.1016/j.cma.2011.01.002
- [11] Garey, M. R., Johnson, D. S., and Stockmeyer, L., "Some Simplified NP-Complete Graph Problems," *Theoretical Computer Science*, Vol. 1, No. 3, 1976, pp. 237–267. doi:10.1016/0304-3975(76)90059-1
- [12] Kernighan, B. W., and Lin, S., "An Efficient Heuristic Procedure for Partitioning Graphs," *The Bell System Technical Journal*, Vol. 49, No. 2, 1970, pp. 291–307. doi:10.1002/bltj.1970.49.issue-2
- [13] Fiduccia, C. M., and Mattheyses, R. M., "A Linear-Time Heuristic for Improving Network Partitions," *Papers on Twenty-Five Years of Electronic Design Automation*, IEEE Publ., Piscataway, NJ, 1988, pp. 241–247. doi:10.1109/dac.1982.1585498
- [14] Sanders, P., and Schulz, C., "High Quality Graph Partitioning," *Graph Partitioning and Graph Clustering*, Vol. 588, No. 1, 2012, pp. 1–17. doi:10.1090/conm/588/11700
- [15] Newman, M. E., "Spectral Methods for Community Detection and Graph Partitioning," *Physical Review E*, Vol. 88, No. 4, 2013, Paper 042822. doi:10.1103/PhysRevE.88.042822
- [16] Buluç, A., Meyerhenke, H., Saffro, I., Sanders, P., and Schulz, C., "Recent Advances in Graph Partitioning," *Algorithm Engineering*,

- Springer, New York, 2016, pp. 117–158.
doi:10.1007/978-3-319-49487-6_4
- [17] Karypis, G., and Kumar, V., “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, 1998, pp. 359–392.
doi:10.1137/S1064827595287997
- [18] Sanders, P., and Schulz, C., “Think Locally, Act Globally: Highly Balanced Graph Partitioning,” *International Symposium on Experimental Algorithms*, Springer, New York, 2013, pp. 164–175.
doi:10.1007/978-3-642-38527-8_16
- [19] Boykov, Y., Veksler, O., and Zabih, R., “Fast Approximate Energy Minimization via Graph Cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, 2001, pp. 1222–1239.
doi:10.1109/34.969114
- [20] Kolmogorov, V., and Rother, C., “Comparison of Energy Minimization Algorithms for Highly Connected Graphs,” *European Conference on Computer Vision*, Springer, New York, 2006, pp. 1–15.
doi:10.1007/11744047_1
- [21] Vazirani, V. V., *Multiway Cut and k-Cut*, Springer, Berlin, 2003, pp. 38–46.
doi:10.1007/978-3-662-04565-7_4
- [22] Murr, L. E., Gaytan, S. M., Ramirez, D. A., Martinez, E., Hernandez, J., Amato, K. N., Shindo, P. W., Medina, F. R., and Wicker, R. B., “Metal Fabrication by Additive Manufacturing Using Laser and Electron Beam Melting Technologies,” *Journal of Materials Science & Technology*, Vol. 28, No. 1, 2012, pp. 1–14.
doi:10.1016/S1005-0302(12)60016-4
- [23] Herzog, D., Seyda, V., Wycisk, E., and Emmelmann, C., “Additive Manufacturing of Metals,” *Acta Materialia*, Vol. 117, Sept. 2016, pp. 371–392.
doi:10.1016/j.actamat.2016.07.019
- [24] Baufeld, B., Van der Biest, O., and Gault, R., “Additive Manufacturing of Ti–6Al–4V Components by Shaped Metal Deposition: Microstructure and Mechanical Properties,” *Materials & Design*, Vol. 31, June 2010, pp. S106–S111.
doi:10.1016/j.matdes.2009.11.032
- [25] Dobrushin, R. L., Kotecký, R., and Shlosman, S., *Wulff Construction: A Global Shape from Local Interaction*, Vol. 104, American Mathematical Soc., Providence, RI, 1992, pp. 1–22.
doi:10.1090/mmono/104/03
- [26] Schmidt, F. R., Töppe, E., and Cremers, D., “Efficient Planar Graph Cuts with Applications in Computer Vision,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Publ., Piscataway, NJ, 2009, pp. 351–356.
doi:10.1109/cvprw.2009.5206863

R. K. Kapania
Associate Editor