

# 3D Superalloy Grain Segmentation Using a Multichannel Edge-Weighted Centroidal Voronoi Tessellation Algorithm

Yu Cao, *Student Member, IEEE*, Lili Ju, Youjie Zhou, *Student Member, IEEE*,  
and Song Wang, *Senior Member, IEEE*

**Abstract**—Accurate grain segmentation on 3D superalloy images is very important in materials science and engineering. From grain segmentation, we can derive the underlying superalloy grains' micro-structures, based on how many important physical, mechanical, and chemical properties of the superalloy samples can be evaluated. Grain segmentation is, however, usually a very challenging problem because: 1) even a small 3D superalloy sample may contain hundreds of grains; 2) carbides and noises may degrade the imaging quality; and 3) the intensity within a grain may not be homogeneous. In addition, the same grain may present different appearances, e.g., different intensities, under different microscope settings. In practice, a 3D superalloy image may contain multichannel information where each channel corresponds to a specific microscope setting. In this paper, we develop a multichannel edge-weighted centroidal Voronoi tessellation (MCEWCVT) algorithm to effectively and robustly segment the superalloy grains from 3D multichannel superalloy images. MCEWCVT performs segmentation by minimizing an energy function, which encodes both the multichannel voxel-intensity similarity within each cluster in the intensity domain and the smoothness of segmentation boundaries in the 3D image domain. In the experiment, we first quantitatively evaluate the proposed MCEWCVT algorithm on a four-channel Ni-based 3D superalloy data set (IN100) against the manually annotated ground-truth segmentation. We further evaluate the MCEWCVT algorithm on two synthesized four-channel superalloy data sets. The qualitative and quantitative comparisons of 18 existing image segmentation algorithms demonstrate the effectiveness and robustness of the proposed MCEWCVT algorithm.

**Index Terms**—3D image segmentation, grain segmentation, centroidal Voronoi tessellation, multichannel imaging.

## I. INTRODUCTION

**I**N material industry, it usually takes scientists and engineers several years to design and develop a new kind of superalloy with desired properties. This process usually

Manuscript received January 13, 2013; revised May 11, 2013; accepted June 6, 2013. Date of publication June 19, 2013; date of current version September 5, 2013. This work was supported in part by AFOSR FA9550-11-1-0327, NSF DMS-0913491, NSF IIS-0951754, NSF IIS-1017199, and the DARPA Mind's Eye program W911NF-10-2-0060. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sina Farsiu.

Y. Cao, Y. Zhou, and S. Wang are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: cao@cec.sc.edu; zhou42@email.sc.edu; songwang@cec.sc.edu).

L. Ju is with the Department of Mathematics, University of South Carolina, Columbia, SC 29208 USA (e-mail: ju@math.sc.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2013.2270113

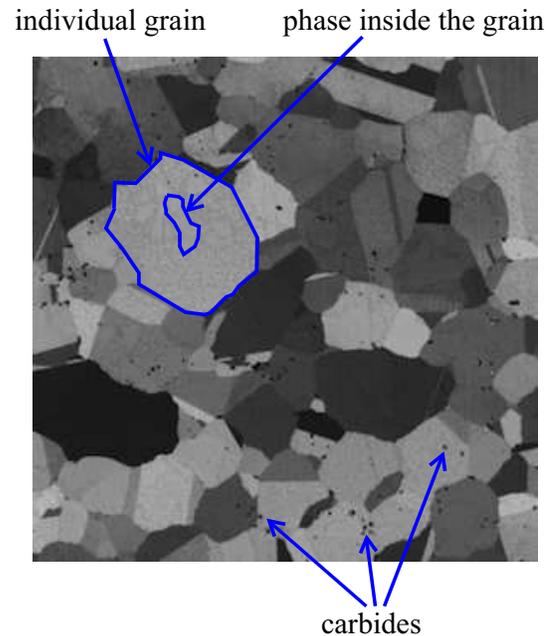


Fig. 1. Superalloy sample image on one serial-sectioned slice.

involves multiple rounds of test sample generation and the microscopic structure analysis. Like many other materials, superalloy is composed of a large number of grains, of which the size, shape and neighboring relations usually determine the physical, mechanical and chemical properties of the sample, e.g., lightness, hardness, stiffness, electrical conductivity, fluid permeability and thermostability [1], [2]. These grains can be imaged by a microscope on serial-sectioned slices. Figure 1 shows one of the slices.

In the materials science community, a common practice of extracting all the grains in a superalloy sample is to manually segment each 2D image slice and then reconstruct the 3D grains by corresponding segments obtained from 2D slices. Given the large number of grains and slices, manual segmentation is very tedious, time consuming and prone to error. In addition, many superalloy images contain multichannel information – each channel corresponds to a specific microscopy setting. Multichannel imaging provides more information to identify the boundary between adjacent grains since two adjacent grains may show similar intensities in one channel but different intensities in another. Figure 2

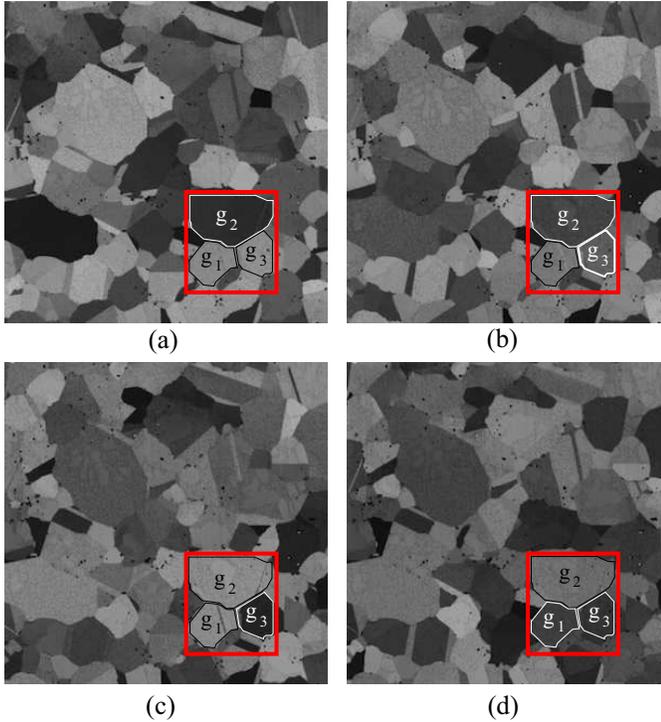


Fig. 2. One slice of a superalloy sample with four image channels (4 different electronic microscope settings). (a) 4000\_Series. (b) 5000\_Series. (c) 6000\_Series. (d) 7000\_Series.

shows a 4-channel image of the same superalloy slice. We can see that adjacent grains  $g_1$  and  $g_2$  can be better separated in channels (a, b, d) than in channel (c). However, adjacent grains  $g_1$  and  $g_3$  can be better separated in (c) than in (a, b, d). This increases the burden of human segmentation since a human annotator has to examine multiple image channels in segmenting one 2D slice. For example, for the 170-slice 4-channel Ni-based image used in our later experiments, three human annotators have to work full time for about two weeks to construct the manual segmentation. Therefore, automatic algorithms are highly desirable for segmenting 3D superalloy images. However, grain segmentation on superalloy images is usually a very challenging problem because 1) the grains to be segmented are of large quantity – even a small 3D superalloy sample may contain hundreds of grains; 2) carbides and noises may degrade the imaging quality; and 3) the intensity within one grain may not be homogeneous, as shown in Figure 1. Additionally, the developed algorithms should be able to leverage the multichannel information for more accurate grain segmentation.

In principle, many general 2D image segmentation methods can be used to automatically segment 2D slices [3]–[13] or semi-automatically [14], [15]. For example, in [16] a stochastic segmentation algorithm following the ‘expectation-maximization’/‘maximization of the posterior marginals’ (EM/MPM) principle is developed for segmenting 2D Ni-based grain images. However, these 2D segmentation methods do not consider the grain-structure continuity between neighboring slices. This not only affects the segmentation accuracy, but also brings difficulties in reconstructing the underlying 3D

grain structures. 3D volume segmentation methods can address this limitation. However, some 3D segmentation methods, such as N-D graph cuts [17], are of high computational and space complexity. Some 3D methods, such as 3D random walks [15], require manually specified seeds for grains as initialization. Some other 3D methods, including 3D level set methods [18], [19], [20], [10] and isosurfaces methods [21], usually handle the segmentation of a small number of disjoint structures very well. Statistical clustering methods, such as K-means, have difficulties in considering the spatial information of the voxels and may produce many undesirable fragments. These limitations prevent their applications to grain segmentation on 3D superalloy images. Additionally, many of these 2D and 3D methods have difficulties in leveraging the multichannel information to improve segmentation accuracy.

In this paper, we propose a new voxel clustering algorithm, dubbed *Multichannel Edge-Weighted Centroidal Voronoi Tessellation* (MCEWCVT) for 3D multichannel superalloy image segmentation. In this algorithm, grain segmentation is achieved by minimizing an energy function that consists of a multichannel clustering energy term (defined in voxel intensity space) and an edge smoothing energy term (defined in 3D image space). As a result, the proposed MCEWCVT can fully take advantage of the multichannel information and produce compact segments with smooth segmentation boundaries. In the experiment, we first quantitatively evaluate the proposed MCEWCVT algorithm on a 4-channel Ni-based 3D superalloy dataset (IN100) against the manually annotated ground-truth. We further evaluate the MCEWCVT algorithm on two synthesized 4-channel superalloy datasets using the optimal parameters found in segmenting the IN100 data. The proposed MCEWCVT algorithm is also compared with 18 existing 2D and 3D image segmentation algorithms on the IN100 dataset.

The remainder of this paper is organized as follows. In Section II, we briefly review the Centroidal Voronoi Tessellation (CVT) model, derive a new clustering energy suitable for multichannel grain image segmentation, and introduce the MCEWCVT algorithm. In Section III, we report the experimental results, together with qualitative and quantitative comparisons against 18 existing segmentation algorithms on the IN100 dataset. In Section IV, we report the experimental results on two synthesized datasets. Section V concludes the paper.

## II. MULTICHANNEL EDGE-WEIGHTED CENTROIDAL VORONOI TESSELLATION ALGORITHM

A 3D superalloy image can be denoted as a function  $u$  defined on a domain  $\Omega \subseteq \mathbb{R}^3$  where the values of  $u$  represent the gray intensities of the voxels. Since the voxels can be indexed by integer triplets,  $u$  is a discrete function defined over a set of points with integer coordinates, i.e. the point  $(x, y, z) = (i, j, k)$ , where  $(i, j, k)$  is an integer triplet that ranges over the volume domain. Thus, the domain of  $u$  for a superalloy volume is an index set  $D = \{(i, j, k) \mid i = 1, \dots, I, j = 1, \dots, J, k = 1, \dots, K\}$  for some positive integers  $I, J$  and  $K$ .

### A. Basic Centroidal Voronoi Tessellation Model

Let  $\mathbb{U} = \{u(i, j, k)\}_{(i,j,k) \in D}$  denote the set of intensity values of the 3D superalloy image and  $\mathcal{W} = \{w_l\}_{l=1}^L$  a set of typical intensity levels. The Voronoi region  $V_l$  ( $l = 1, 2, \dots, L$ ) in  $\mathbb{U}$  corresponding to  $w_l$  is defined by

$$V_l = \{u(i, j, k) \in \mathbb{U} \mid |u(i, j, k) - w_l| \leq |u(i, j, k) - w_m|, m = 1, 2, \dots, L, l \neq m\}, \quad (1)$$

where  $|\cdot|$  could be pre-defined metric measures such as the Euclidean distance. The set  $\mathcal{V} = \{V_l\}_{l=1}^L$  is called a *Voronoi tessellation* or *Voronoi clustering* [22]–[24] of the set  $\mathbb{U}$ . The set of chosen intensities  $\mathcal{W} = \{w_l\}_{l=1}^L$  are referred as the *Voronoi generators*. Since we have  $V_p \cap V_q = \emptyset$  if  $p \neq q$  and  $\mathbb{U} = \bigcup_{l=1}^L V_l$ , the Voronoi tessellation  $\mathcal{V}$  can be viewed as a special partition of  $\mathbb{U}$  in the physical space.

Let  $\rho$  be a pre-defined density function defined on  $D$ . Given a partition of  $\mathbb{U}$ , denoted by  $\{U_l\}_{l=1}^L$ , the *centroid* (i.e. *center of mass* or *cluster mean*) of each cell  $U_l$ , with respect to the density  $\rho$ , is defined to be the intensity  $w_l^* \in U_l$  which minimizes

$$\min_{w \in U_l} \sum_{u(i,j,k) \in U_l} \rho(i, j, k) |u(i, j, k) - w|^2. \quad (2)$$

Note that this centroid definition is slightly different from that proposed in [25] with the newly introduced density function. For an arbitrary Voronoi tessellation ( $\{w_l\}_{l=1}^L; \{V_l\}_{l=1}^L$ ) of  $\mathbb{U}$ , it is often not the case that  $w_l = w_l^*$  for  $l = 1, 2, \dots, L$ , where  $\{w_l^*\}_{l=1}^L$  are the corresponding centroids of  $\{V_l\}_{l=1}^L$ . If the generators of the Voronoi regions  $\{V_l\}_{l=1}^L$  of  $\mathbb{U}$  coincide with their corresponding centroids, i.e.

$$w_l = w_l^*, \quad \text{for } l = 1, 2, \dots, L,$$

then we call the Voronoi tessellation  $\{V_l\}_{l=1}^L$  a *centroidal Voronoi tessellation* (CVT) [23] of  $\mathbb{U}$  and refer to  $\{w_l\}_{l=1}^L$  as the corresponding CVT generators.

The construction of CVTs can be achieved by an “energy” minimization process [23]. Generally, for any set of generators  $\mathcal{W} = \{w_l\}_{l=1}^L$  and any partition  $\mathcal{U} = \{U_l\}_{l=1}^L$  of  $\mathbb{U}$ , the classic *clustering energy* of  $(\mathcal{W}; \mathcal{U})$  can be defined as:

$$E_C(\mathcal{W}; \mathcal{U}) = \sum_{l=1}^L \sum_{u(i,j,k) \in U_l} \rho(i, j, k) |u(i, j, k) - w_l|^2. \quad (3)$$

Suppose we have determined the clusters  $\{U_l\}_{l=1}^L$  for a given 3D superalloy image represented by  $u(i, j, k)$  for  $(i, j, k) \in D$ . Then a segmentation in the physical space of the image can be naturally produced as  $\mathcal{D} = \{D_l\}_{l=1}^L$ , where

$$D_l = \{(i, j, k) \mid u(i, j, k) \in U_l\}.$$

Note that  $L$  is not the number of grains in the underlying 3D superalloy image – each  $D_l$  may contain multiple disjoint 3D segments which may represent different grains. In another word, non-adjacent grains may have identical or similar intensities and be clustered into the same  $D_l$ . In practice, we usually choose  $L$  to be smaller than the expected number of grains in the image when using Voronoi tessellation algorithms.

Consequently, the classic clustering energy (3) can be rewritten in physical segmentation terminology as

$$E_C(\mathcal{W}; \mathcal{D}) = \sum_{l=1}^L \sum_{(i,j,k) \in D_l} \rho(i, j, k) |u(i, j, k) - w_l|^2. \quad (4)$$

It is well known from [23] that  $(\mathcal{W}; \mathcal{D})$  is a minimizer of  $E_C(\mathcal{W}; \mathcal{D})$  only if  $(\mathcal{W}; \mathcal{D})$  forms a CVT of  $D$ . Several algorithms of minimizing the above energy function can be found in [26]–[28].

### B. Multichannel Clustering Energy

Let  $N$  denote the number of channels, i.e., we have  $N$  images,  $u^1, u^2, \dots, u^N$ , of the same superalloy sample photographed under different microscope settings. Then we can rewrite  $\mathbb{U}$  and  $\mathcal{W}$  in the vector form

$$\mathbb{U} = \{\vec{u}(i, j, k) = (u^1, u^2, \dots, u^N)^T \in \mathbb{R}^N\}_{(i,j,k) \in D}$$

and

$$\mathcal{W} = \{\vec{w}_l = (w_l^1, w_l^2, \dots, w_l^N)^T \in \mathbb{R}^N\}_{l=1}^L,$$

respectively.

Note that one grain may be visually separable in one channel but has the close intensity value with its adjacent grains in other channels. In order to capture the grains with intensities that are distinct with its adjacent grains only in some (but not all) of the channels, we need to choose a proper measurement of the distance from  $\vec{u}(i, j, k)$  to  $\vec{w}_l$ . In this paper, we take the  $L_\infty$  norm as the distance measurement, which is defined as

$$\|\vec{x}\|_\infty = \max(|x^1|, |x^2|, \dots, |x^N|)$$

where  $\vec{x} = (x^1, x^2, \dots, x^N) \in \mathbb{R}^N$ .

This way, we can rewrite the classic clustering energy (4) as

$$E_C(\mathcal{W}; \mathcal{D}) = \sum_{l=1}^L \sum_{(i,j,k) \in D_l} \rho(i, j, k) \|\vec{u}(i, j, k) - \vec{w}_l\|_\infty^2. \quad (5)$$

Note that (5) will reduce to (4) when  $N = 1$ .

### C. Edge Energy

Besides the multichannel clustering energy term, we would like to further utilize an edge related energy term to enforces the continuity and smoothness on the boundaries of the 3D superalloy grains/segments. In this paper, we define the edge related energy term for a given clustering  $\mathcal{D}$  of the physical space similar to that proposed in [25], [29].

For each voxel  $(i, j, k) \in D$ , denote  $\mathbb{N}_\omega(i, j, k)$  a local neighborhood, which could be a  $2\omega \times 2\omega \times 2\omega$  cube centered at  $(i, j, k)$  or a sphere centered at  $(i, j, k)$  with radius  $\omega$ . We then define a local characteristic function  $\chi_{(i,j,k)} : \mathbb{N}_\omega(i, j, k) \rightarrow \{0, 1\}$  as

$$\chi_{(i,j,k)}(i', j', k') = \begin{cases} 1 & \text{if } \pi_u(i', j', k') \neq \pi_u(i, j, k), \\ 0 & \text{otherwise,} \end{cases}$$

where  $\pi_u(i, j, k) : D \rightarrow \{1, \dots, L\}$  indicates the cluster that  $\vec{u}(i, j, k)$  belongs to. The edge energy term can be defined as

$$E_L(\mathcal{D}) = \sum_{(i,j,k) \in D} \sum_{(i',j',k') \in \mathbb{N}_\omega(i,j,k)} \chi_{(i,j,k)}(i', j', k'). \quad (6)$$

Generalizing the analysis for 2D cases in [25], it is not too hard to demonstrate that  $E_L$  is proportional to  $\omega^4 A$  in the asymptotic sense where  $A$  is the area of the boundaries (surfaces in the 3D space) between the 3D segments. Note that the edge energy has nothing to do with the generators  $\mathcal{W}$ .

#### D. Total Energy

In order to enforce the clear detection of grain boundaries, we take the density function  $\rho$  as

$$\rho = 1 + |\nabla \bar{u}|. \quad (7)$$

By combining (5), (6) and (7), we can define the total energy for our model, i.e., the *multichannel edge-weighted clustering energy*, as:

$$\begin{aligned} E(\mathcal{W}; \mathcal{D}) &= E_C(\mathcal{W}; \mathcal{D}) + \lambda E_L(\mathcal{D}) \\ &= \sum_{l=1}^L \sum_{(i,j,k) \in D_l} (1 + |\nabla \bar{u}(i, j, k)|) \|\bar{u}(i, j, k) - \bar{w}_l\|_\infty^2 \\ &\quad + \lambda \sum_{(i,j,k) \in D} \sum_{(i',j',k') \in \mathbb{N}_w(i,j,k)} \chi_{(i,j,k)}(i', j', k') \end{aligned} \quad (8)$$

where  $\lambda$  is a weighting function controlling the balance between  $E_C$  and  $E_L$ .

#### E. Multichannel Edge-Weighted Distance

Basically, the total multichannel edge-weighted energy minimization is achieved through iteratively transferring voxels from one cluster to another. Specifically, in each iteration, we transfer a voxel to a certain cluster which decreases the total energy the most.

Let us rewrite equation (8) as

$$\begin{aligned} E(\mathcal{W}; \mathcal{D}) &= \\ &\sum_{(i',j',k') \in D \setminus (i,j,k)} \rho(i', j', k') \|\bar{u}(i', j', k') - \bar{w}_{\pi_u(i',j',k')}\|_\infty^2 \\ &\quad + \rho(i, j, k) \|\bar{u}(i, j, k) - \bar{w}_{\pi_u(i,j,k)}\|_\infty^2 \\ &\quad + \sum_{(i',j',k') \in D \setminus (i,j,k)} \epsilon_{\mathcal{L}}(i', j', k') \\ &\quad + \epsilon_{\mathcal{L}}(i, j, k) \end{aligned} \quad (9)$$

where

$$\epsilon_{\mathcal{L}}(i, j, k) = \lambda \sum_{(i',j',k') \in \mathbb{N}_w(i,j,k)} \chi_{(i,j,k)}(i', j', k').$$

Now let's consider the variation of the total energy when transferring a voxel  $(i, j, k)$  from its current cluster  $D_l$  to another cluster  $D_m$ . In equation (9), the first term on the right side has no change. The change of the second term is

$$\rho(i, j, k) (\|\bar{u}(i, j, k) - \bar{w}_m\|_\infty^2 - \|\bar{u}(i, j, k) - \bar{w}_l\|_\infty^2). \quad (10)$$

Denote  $n_k(i, j, k)$  the number of voxels within  $(D_k \cap \mathbb{N}_w(i, j, k)) \setminus (i, j, k)$ . Based on the analysis from [25], we know that the changes in the third and fourth term after transferring are both equal to

$$\lambda n_l(i, j, k) - \lambda n_m(i, j, k) = \lambda [n_l(i, j, k) - n_m(i, j, k)]. \quad (11)$$

By combining (10) and (11), we have the overall variation of the total energy by transferring voxel  $(i, j, k)$  from cluster  $D_l$  to  $D_m$  as

$$\begin{aligned} &\rho(i, j, k) (\|\bar{u}(i, j, k) - \bar{w}_m\|_\infty^2 - \|\bar{u}(i, j, k) - \bar{w}_l\|_\infty^2) \\ &\quad + 2\lambda (n_l(i, j, k) - n_m(i, j, k)), \end{aligned}$$

which can be rewritten as

$$\begin{aligned} &[\rho(i, j, k) \|\bar{u}(i, j, k) - \bar{w}_m\|_\infty^2 - 2\lambda n_m(i, j, k)] \\ &\quad - [\rho(i, j, k) \|\bar{u}(i, j, k) - \bar{w}_l\|_\infty^2 - 2\lambda n_l(i, j, k)]. \end{aligned} \quad (12)$$

Thus we define the *multichannel edge-weighted distance* from a voxel  $(i, j, k)$  to generator  $\bar{w}_l$  to be

$$\begin{aligned} &dist((i, j, k), \bar{w}_l) \\ &= \sqrt{\rho(i, j, k) \|\bar{u}(i, j, k) - \bar{w}_l\|_\infty^2 + 2\lambda \tilde{n}_l(i, j, k)} \\ &= \sqrt{(1 + |\nabla \bar{u}(i, j, k)|) \|\bar{u}(i, j, k) - \bar{w}_l\|_\infty^2 + 2\lambda \tilde{n}_l(i, j, k)} \end{aligned} \quad (13)$$

where  $\tilde{n}_l(i, j, k) = |\mathbb{N}_w(i, j, k)| - n_l(i, j, k) - 1$  is the number of voxels in  $\mathbb{N}_w(i, j, k) \setminus (D_l \cup (i, j, k))$ .

In conclusion, moving a voxel to the cluster of a generator to which it has the shortest edge-weighted distance defined by (13) will decrease the total clustering energy  $E(\mathcal{W}; \mathcal{D})$  the most.

#### F. Multichannel Edge-Weighted Voronoi Regions

Given a set of multichannel generators  $\mathcal{W} = \{\bar{w}_l\}_{l=1}^L$ , we can define the multichannel edge-weighted Voronoi region  $\tilde{\mathcal{D}} = \{\tilde{D}_l\}_{l=1}^L$  in the physical volume space  $D$  as

$$\begin{aligned} \tilde{D}_l &= \{(i, j, k) \in D \mid dist((i, j, k), \bar{w}_l) \leq \\ &\quad dist((i, j, k), \bar{w}_m), m = 1, \dots, L, l \neq m)\}. \end{aligned} \quad (14)$$

From equations (12) and (13), it is easy to find that when  $\mathcal{W}$  is fixed, the multichannel edge-weighted Voronoi tessellation  $\tilde{\mathcal{D}} = \{\tilde{D}_l\}_{l=1}^L$  associated with  $\mathcal{W}$  corresponds to the minimizer of the multichannel edge-weighted energy  $E(\mathcal{W}; \mathcal{D})$ , i.e.,

$$\tilde{\mathcal{D}} = \arg \min_{\mathcal{D}} E(\mathcal{W}; \mathcal{D}).$$

Then we define the *multichannel edge-weighted Voronoi tessellation energy* for a given set of generators  $\mathcal{W} = \{\bar{w}_l\}_{l=1}^L$  to be

$$E_{MCEWVT}(\mathcal{W}) = E(\mathcal{W}; \tilde{\mathcal{D}}). \quad (15)$$

Algorithm 1 can be used to efficiently construct the multichannel edge-weighted Voronoi regions for a given set of generators.

#### G. The MCEWCVT Model and Its Implementation

In order to define the MCEWCVT model, we need to further determine the centroids of a given set of partition  $\tilde{\mathcal{D}} = \{\tilde{D}_l\}_{l=1}^L$  of  $D$ , i.e., find  $\mathcal{W} = \{\bar{w}_l^*\}_{l=1}^L$  such that

$$\bar{w}_l^* = \arg \min_{\bar{w}} \sum_{(i,j,k) \in \tilde{D}_l} \rho(i, j, k) \|\bar{u}(i, j, k) - \bar{w}\|_\infty^2 \quad (16)$$

for  $l = 1, 2, \dots, L$ . Since we use the  $L_\infty$ -norm, it is hard to find an analytical solution for  $\bar{w}_l$ . Usually,  $\bar{w}_l$  defined

---

**Algorithm 1:**  $\{\tilde{D}_l\}_{l=1}^L = \text{MCEWVT}(\vec{u}, \{\vec{w}_l\}_{l=1}^L, \{D_l\}_{l=1}^L)$

---

- 1: INPUT: A set of  $N$  images determined by  $\vec{u}$ , a set of generators  $\{\vec{w}_l\}_{l=1}^L$  and an initial partition  $\{D_l\}_{l=1}^L$  of the physical space  $D$  (can be arbitrarily chosen).
  - 2: START:
  - 3: **for all** voxels  $(i, j, k) \in D$  **do**
  - 4:   a) calculate the multichannel edge-weighted distances from the voxel  $(i, j, k)$  to all generators  $\{\vec{w}_l\}_{l=1}^L$ .
  - 5:   b) transfer the voxel  $(i, j, k)$  from its current cluster to the cluster whose generator has the shortest multichannel edge-weighted distance to it.
  - 6: **end for**
  - 7: If no voxel in the loop is transferred, return  $\{\tilde{D}_l\}_{l=1}^L = \{D_l\}_{l=1}^L$  and exit; otherwise, go to step 3.
- 

---

**Algorithm 2:**  $(\{\tilde{w}_l\}_{l=1}^L, \{\tilde{D}_l\}_{l=1}^L) = \text{MCEWCVT}(\vec{u}, L)$

---

- 1: INPUT: A set of  $N$  images determined by  $\vec{u}$  and an integer  $L$ .
  - 2: START:
  - 3: Arbitrarily choose an initial partition  $\{\tilde{D}_l\}_{l=1}^L$  of the physical space  $D$  or by some specific methods.
  - 4: For each cluster  $\tilde{D}_l, l = 1, \dots, L$ , calculate its cluster centroid  $\vec{w}_l^*$ .
  - 5: Take  $\{\vec{w}_l^*\}_{l=1}^L$  as the generators, calculate an updated partitioning  $\{\tilde{D}'_l\}_{l=1}^L$  by using Algorithm 1.
  - 6: If  $\{\tilde{D}'_l\}_{l=1}^L$  and  $\{\tilde{D}_l\}_{l=1}^L$  are the same, return  $(\{\vec{w}_l\}_{l=1}^L; \{D_l\}_{l=1}^L)$  and exit; otherwise, set  $\tilde{D}_l = \tilde{D}'_l$  for  $l = 1, \dots, L$  and go to step 4.
- 

through the above minimization process could be solved numerically. For example, the Powell method [30] could be used to numerically calculate  $\vec{w}_l$  although there is no derivative information available.

**Definition (MCEWCVT):** For a given multichannel edge-weighted Voronoi tessellation  $(\{\vec{w}_l\}_{l=1}^L; \{\tilde{D}_l\}_{l=1}^L)$  of  $D$ , we call it a multichannel edge-weighted centroidal Voronoi tessellation (MCEWCVT) of  $D$  if the generators  $\{\vec{w}_l\}_{l=1}^L$  are also the corresponding centroids of the associated multichannel edge-weighted Voronoi regions  $\{\tilde{D}_l\}_{l=1}^L$ , i.e.,

$$\vec{w}_l = \vec{w}_l^*, \quad l = 1, 2, \dots, L.$$

Based on the CVT principle (see Section 3 of [23]), we know that  $(\mathcal{W}; \tilde{\mathcal{D}})$  is a minimizer of  $E(\mathcal{W}; \tilde{\mathcal{D}})$  only if  $(\mathcal{W}; \tilde{\mathcal{D}})$  forms a MCEWCVT of  $D$ . We propose Algorithm 2 for constructing the MCEWCVTs. As discussed in [25] for the EWCVT construction algorithms, some improvements of Algorithm 2 can be obtained by using a narrow-banded implementation and better initialization processes. We also note that the energy  $E_{\text{MCEWVT}}(\mathcal{W})$  keeps decreasing along the iterations in this algorithm.

#### H. Comparing MCEWCVT with the Lloyd Algorithm

The Lloyd algorithm provides a popular implementation to CVT/K-means. The proposed MCEWCVT algorithm and

the Lloyd algorithm share similar algorithmic structures. Specifically, both of them are iterative optimization algorithms where each iteration consists of two steps: 1) assignment step that assigns each data point to the cluster with the closest center; and 2) updating step that updates cluster centers to be the centroids of the data points in each cluster. In addition, both algorithms share a similar convergence property. According to [31], [32], as the clustering energy decreases monotonically, a clustering algorithm must be weakly convergent. Thus, the proposed MCEWCVT algorithm is weakly convergent given its minimization process. In Section III, we will show that the clustering energy of MCEWCVT decreases in the optimization when segmenting superalloy data.

However, there is a major difference between the proposed MCEWCVT algorithm and the Lloyd algorithm. The Lloyd algorithm handles the energy where the  $L_2$  norm is used to measure the distance from a data point to a cluster center. This way, the Lloyd algorithm updates the cluster centers using mean values of the data in each cluster. In MCEWCVT, we include an edge-smoothness term in the clustering energy and use a multichannel edge-weighted distance based on  $L_\infty$  norm. As a result, the MCEWCVT algorithm theoretically needs to use Powell method rather than mean values to update new cluster centers.

### III. EXPERIMENTS ON IN100 DATASET

#### A. Testing Dataset

We first test the proposed algorithm on an *IN100* (Ni-based) multichannel superalloy dataset. This dataset consists of 4 channels of superalloy slice images taken under different electronic microscope parameter settings. Each slice was photographed as new facets appearing by keeping abrading the up-front facet of the superalloy sample. The size of each 2D slice is  $671 \times 671$  and the number of slices in each channel is 170. The resolution within a slice is  $0.2 \mu\text{m}$  and the resolution between slices is  $1 \mu\text{m}$ . One can notice that the resolution within 2D slices are 5 times of that between adjacent image slices. This fact may not affect 2D segmentation/edge-detection results, but will affect 3D segmentation results, especially for the results produced by the proposed algorithm, since the measurement of boundary smoothness (i.e., the edge related energy term (6)) requires identical/close image resolutions along all three directions. To address this issue, we linearly interpolate the 3D superalloy image with 4 additional slices between each pair of consecutive slices in the original data. This way, the interpolated data contain  $169 \times 5 + 1 = 846$  slices. The interpolated data volume contains roughly 2,500 grains. The testing dataset also comes with the ground-truth segmentation created by manually annotating each 2D slice. The human annotators consider grain intensity variations from all 4 channels and extract unified grain boundaries in each slice using annotation tools provided by the Berkeley Segmentation Benchmark [12].

#### B. Experiment Design

In the experiments, we quantitatively evaluate the segmentation results by examining the coincidence between the detected

boundaries and the ground-truth grain boundaries. Specifically, we calculate the  $F_1$ -measure (using the tool provided in [12]) which is the harmonic mean of precision and recall, i.e.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

In the following, we evaluate the proposed algorithm under different parameter settings and compare the performance with 18 well known 2D/3D segmentation/edge-detection methods quantitatively and qualitatively. For 3D methods, in order to obtain considerable good results within a reasonable amount of time, we downsize the original volume size to  $336 \times 336 \times 508$ . We apply 3D methods on the downsized image volume and rescale the segmentation results on each 2D slice to the original size, i.e.  $671 \times 671$ . In addition, for fair comparison, we apply 2D methods on the downsized non-interpolated slices (with size  $336 \times 336$ ), followed by rescaling the segmentation results back to the original size. The final segmentation results of 2D/3D methods on the 170 non-interpolated slices are used to conduct quantitative evaluation against the ground-truth. We also provide  $p$ -values of the results produced by the proposed MCEWCVT and the comparison methods.

### C. Results of MCEWCVT

Theoretically, MCEWCVT (Algorithm 2) stops when the energy function (8) stays unchanged and this may require a very long running time. A practical stop condition could be

$$\frac{|E_{i+1} - E_i|}{E_i} < \epsilon. \quad (17)$$

We can also set a maximum allowed iterations  $maxIter$  to stop the algorithm. In our experiments, MCEWCVT stops when any one of these two stop conditions is triggered. In addition, we use CVT/K-means clustering results as the initial partitions for the MCEWCVT algorithm (step 3 in Algorithm 2).

This way, there are totally six parameters/factors that can be tuned in MCEWCVT:

- $L$ , the number of clusters;
- $\lambda$ , the weighting parameter which balances the intensity clustering energy term  $E_C$  and the edge smoothing energy term  $E_L$ ;
- $\omega$  and  $\mathbb{N}_\omega(i, j, k)$ , which define the size and shape of a 3D local neighborhood region (centered at each voxel  $(i, j, k)$ );
- $\epsilon$  and  $maxIter$ , the pre-defined threshold and maximum allowed iterations as the stop conditions of MCEWCVT.

Enumerating all possible values and combinations of these six parameters/factors would be to some extent infeasible, thus we fix three parameters/factors ( $\mathbb{N}_\omega(i, j, k)$ ,  $\epsilon$  and  $maxIter$ ), and test different values for the other three parameters ( $L$ ,  $\lambda$  and  $\omega$ ). Specifically, we set  $\epsilon = 10^{-4}$ ,  $maxIter = 140$ , and the shape of  $\mathbb{N}_\omega(i, j, k)$  to be a sphere with radius  $\omega$ . The different values tested for  $L$ ,  $\lambda$  and  $\omega$  are  $L = \{15, 25, 35, 55, 100, 150, 200, 300, 500\}$ ,  $\lambda = \{100, 200, 300, 400, 500\}$ ,  $\omega = \{4, 6, 8\}$ , respectively. Table I summarizes the segmentation accuracy ( $F_1$ -measure) under different  $\{L, \lambda, \omega\}$ . The best result is shaded. From the

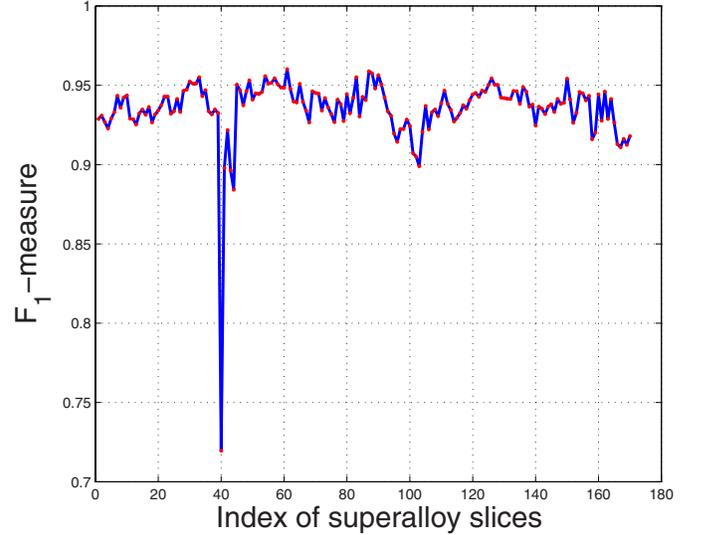


Fig. 3. Segmentation accuracy ( $F_1$ -measure) on each superalloy image slice when using parameters  $\{L = 300, \lambda = 300, \omega = 6\}$ .

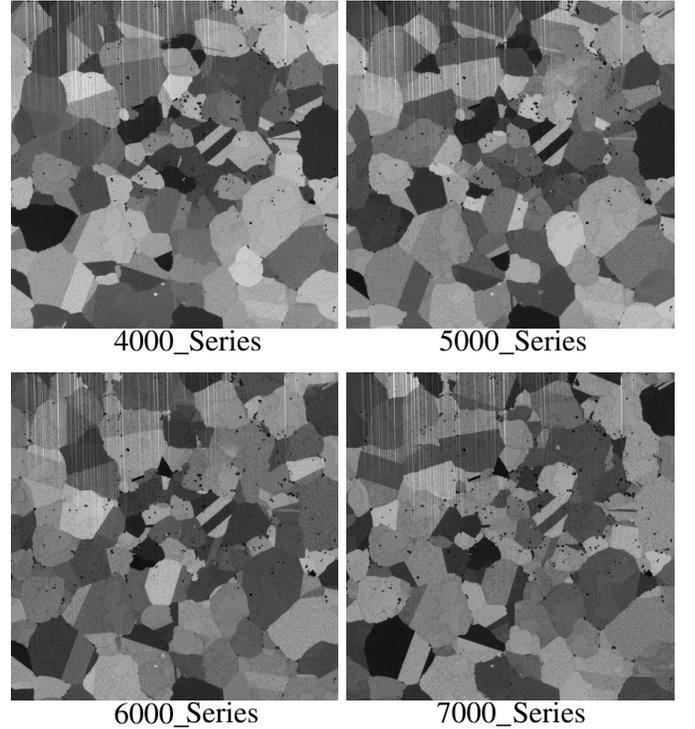


Fig. 4. A superalloy image slice with strong imaging noises (the 40-th slice).

table, we can find that  $L = 300$ ,  $\lambda = 300$  and  $\omega = 6$  lead to the highest segmentation accuracy among all tested values for  $L$ ,  $\lambda$  and  $\omega$ .

Figure 3 further shows  $F_1$ -measures on each 2D image slice (when using parameters  $\{L = 300, \lambda = 300, \omega = 6\}$ ). The low segmentation accuracy on the 40-th slice is caused by strong image noises as shown in Fig. 4.

Figure 5 shows a 3D visualization of the MCEWCVT segmentation using parameters  $\{L = 300, \lambda = 300, \omega = 6\}$  which correspond to the best result shaded in Table I.

Figure 6 shows that the energy decreases along the iterations of the MCEWCVT algorithm. This supports the statement in

TABLE I  
SEGMENTATION ACCURACY ( $F_1$ -MEASURE) OF MCEWCVT UNDER DIFFERENT VALUES OF  $L$ ,  $\lambda$ , AND  $\omega$ . THE BEST RESULT IS SHADED.

	$(L, \omega) = (15, 4)$	$(L, \omega) = (15, 6)$	$(L, \omega) = (15, 8)$	$(L, \omega) = (25, 4)$	$(L, \omega) = (25, 6)$	$(L, \omega) = (25, 8)$	$(L, \omega) = (35, 4)$	$(L, \omega) = (35, 6)$	$(L, \omega) = (35, 8)$
$\lambda = 100$	74.96%	86.81%	89.06%	75.90%	88.20%	90.46%	77.24%	89.27%	91.10%
$\lambda = 200$	82.82%	89.34%	87.48%	84.27%	90.63%	89.32%	85.50%	91.30%	90.16%
$\lambda = 300$	85.99%	89.19%	84.33%	87.29%	90.72%	86.86%	88.31%	91.34%	87.81%
$\lambda = 400$	87.48%	88.63%	80.73%	88.75%	90.31%	83.73%	89.53%	90.98%	84.86%
$\lambda = 500$	88.42%	87.83%	76.96%	89.55%	89.61%	80.34%	90.21%	90.44%	81.78%
	$(L, \omega) = (55, 4)$	$(L, \omega) = (55, 6)$	$(L, \omega) = (55, 8)$	$(L, \omega) = (100, 4)$	$(L, \omega) = (100, 6)$	$(L, \omega) = (100, 8)$	$(L, \omega) = (150, 4)$	$(L, \omega) = (150, 6)$	$(L, \omega) = (150, 8)$
$\lambda = 100$	77.33%	90.18%	92.05%	80.17%	91.28%	92.70%	81.63%	91.78%	92.95%
$\lambda = 200$	86.22%	92.20%	91.22%	87.71%	92.75%	92.02%	88.59%	92.90%	92.22%
$\lambda = 300$	89.03%	92.40%	89.09%	90.05%	92.97%	90.06%	90.64%	93.15%	90.37%
$\lambda = 400$	90.32%	92.14%	86.35%	91.18%	92.79%	87.69%	91.61%	93.01%	88.05%
$\lambda = 500$	91.10%	91.62%	83.56%	91.82%	92.40%	85.08%	92.15%	92.55%	85.72%
	$(L, \omega) = (200, 4)$	$(L, \omega) = (200, 6)$	$(L, \omega) = (200, 8)$	$(L, \omega) = (300, 4)$	$(L, \omega) = (300, 6)$	$(L, \omega) = (300, 8)$	$(L, \omega) = (500, 4)$	$(L, \omega) = (500, 6)$	$(L, \omega) = (500, 8)$
$\lambda = 100$	82.66%	92.02%	92.99%	84.06%	92.44%	93.42%	85.46%	92.46%	93.46%
$\lambda = 200$	88.90%	93.08%	92.47%	89.58%	93.41%	92.74%	90.03%	93.34%	93.02%
$\lambda = 300$	90.76%	93.26%	90.66%	91.13%	<b>93.57%</b>	90.79%	91.29%	93.53%	90.98%
$\lambda = 400$	91.64%	93.13%	88.46%	91.88%	93.51%	88.66%	91.84%	93.44%	88.85%
$\lambda = 500$	92.21%	92.73%	86.02%	92.23%	93.14%	86.47%	92.24%	93.15%	86.53%

Section II-H that the MCEWCVT is weakly convergent.

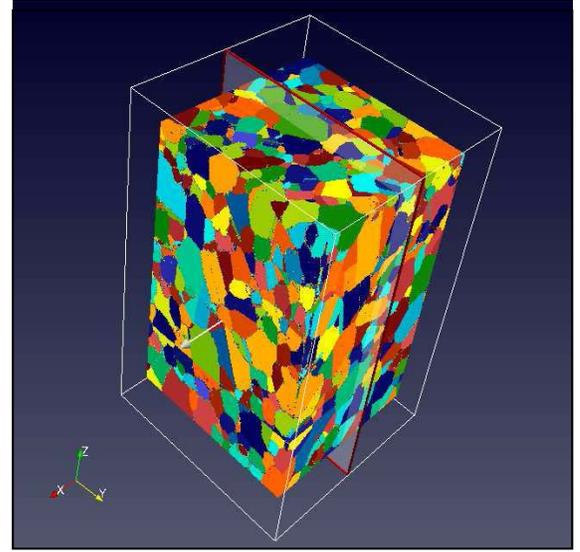
#### D. Robustness of MCEWCVT to Random CVT Initializations

As mentioned above, we use CVT to initialize the clusters for the proposed MCEWCVT algorithm. Since CVT randomly chooses initial cluster centers, we would like to further investigate the robustness of the MCEWCVT to random CVT initializations. In this section, we choose parameters which lead to the best segmentation accuracy, i.e.  $\{L = 300, \lambda = 300, \omega = 6\}$ , to evaluate MCEWCVT algorithm based on 10 CVT initializations (all using  $L = 300$ ). The segmentation accuracies with respect to these 10 CVT initializations are presented in Table II, from which we can see that the MCEWCVT algorithm is very robust to random CVT initializations.

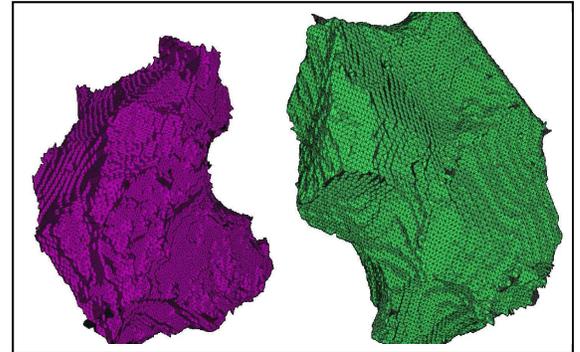
#### E. Comparing with Existing 2D/3D Segmentation Methods

In order to further validate the proposed MCEWCVT algorithm, we compare it with other 18 2D/3D image segmentation methods. Largely based on the code's availability, we choose ten 2D segmentation methods, including 2D level set [10], mean shift [3], watershed [4], statistical region merging (srm) [5], normalized cuts [6], efficient-graph based segmentation [7], topological watersheds [8], EM/MPM [16], power watersheds [14], random walks [15]; five 2D soft edge-detection methods, including globalized probability of boundary (gpb) [11], Berkeley brightness/color/texture gradient detectors (pbCanny, pbCBTG and pbBGTG) [12], ultrametric contour maps (ucm) [13]; three 3D segmentation methods, including 3D watershed [10], 3D level set [4] and CVT/K-means (directly applied on 3D images as in the proposed method). Specifically, for 2D methods, we only apply them onto 170 non-interpolated 2D slices. For 3D methods, we apply them to the interpolated 3D superalloy image and then project 3D segmentation boundaries onto 170 initial slices for quantitative performance evaluation. Note that gpb, pbCanny, pbCBTG and pbBGTB are soft edge detection algorithms – they only detect disjoint edges and may not produce complete boundaries to partition an image into separate grains.

Note that most of these comparison methods were not developed for multichannel imaging. While some of them can incorporate multi-dimensional color information, they usually



(a)



(b)

Fig. 5. (a) Visualization of the MCEWCVT segmentation result using Paraview [33]. Clusters are shown in different colors. (b) Two segmented grains, visualized using MeshLab [34].

use  $L_2$  norm for defining the distance in the color space. For superalloy images, we need to use  $L_\infty$  norm to combine the multichannel information, as justified in Section II-B. To thoroughly evaluate the performance of these comparison methods, we take the 4 image channels  $u_1, u_2, u_3$  and  $u_4$  and

TABLE II

SEGMENTATION ACCURACY ( $F_1$ -MEASURE) WITH RESPECT TO 10 CVT RANDOM INITIALIZATIONS, USING PARAMETERS  $\{L = 300, \lambda = 300, \omega = 6\}$ .

Initialization	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
$F_1$ -measure	93.5444%	93.5437%	93.5443%	93.5441%	93.5425%	93.5413%	93.5425%	93.542%	93.5463%	93.5447%

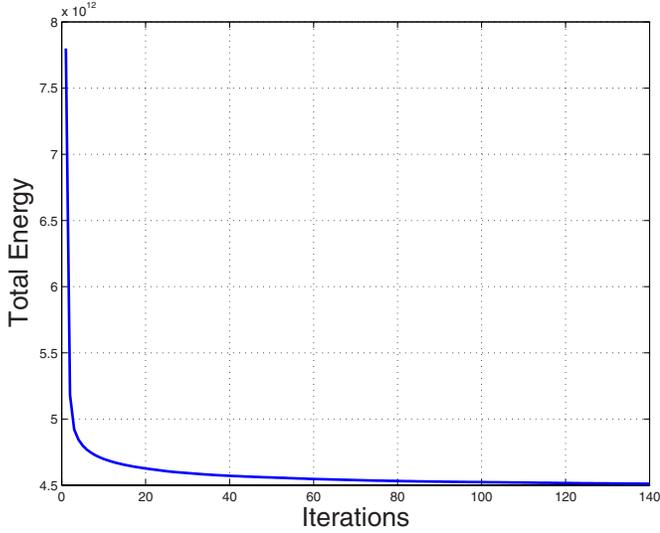


Fig. 6. Energy decreasing as the MCEWCVT algorithm performs iterative minimization.

further create two synthesized channels as

$$u_5 = \frac{u_1 + u_2 + u_3 + u_4}{4}$$

and

$$u_6 = \sqrt{\frac{u_1^2 + u_2^2 + u_3^2 + u_4^2}{4}}.$$

In our experiment, for all the comparison methods except for CVT, we apply them to segment each of these six channels independently. This way, for such a comparison method, we obtain six different segmentation results on the test 3D superalloy image. They are evaluated against the ground-truth segmentation independently and the result with the best  $F_1$ -measure (obtained from one of these six channels) is finally selected as the segmentation result. We use this best  $F_1$ -measure as the segmentation accuracy for this comparison method. First of all, we evaluate the performance of the classical CVT/K-means algorithm since the proposed MCEWCVT can be treated as an extension of the CVT/K-means algorithm. The classical CVT algorithm does not consider the edge related energy term. In our experiment, we simply use it to do clustering in the multichannel intensity space with Euclidean distance. We try the cluster number of  $L$  in  $\{2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, 60, 100, 150, 200, 300, 500, 800\}$ . From Table III, we can see that as the cluster number increases, the segmentation accuracy first increases and then decreases. The best segmentation accuracy appears at  $L = 3$  (shaded). In general, without the boundary-smoothness term, CVT is sensitive to image noise and this gets more severe when the cluster number is large.

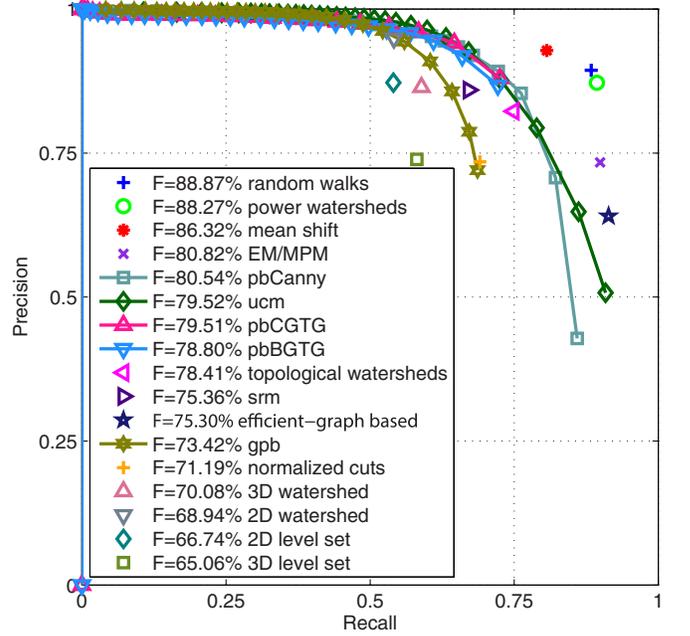


Fig. 7. Precision-recall curves or values of the comparison methods except for CVT.

For the other 17 comparison methods, their parameter settings are summarized in Table IV. For the parameters that are not discussed in this table, we simply use their default settings provided in the corresponding software packages. For the soft-edge detectors, i.e. gpb, pbCanny, pbCBTG, pbBGTB, and ucm, we vary the threshold to the edge probability map to derive a precision-recall curve. For the other methods, we obtain a precision value and a recall value from the segmentation result. The precision-recall curves or values of these 17 comparison methods are shown in Fig. 7.

We compile the  $F_1$ -measures of the proposed algorithm and comparison methods into Table V. Specifically, for the proposed MCEWCVT, we select  $\{L = 300, \lambda = 300, \omega = 6\}$ . For CVT, we select  $L = 3$ . For gpb, pbCanny, pbCBTG and pbBGTB, we select the best  $F_1$ -measure from the precision-recall curve. For all the comparison methods other than CVT, we use the parameters and settings shown in Table IV. For random walks, power watersheds and the 2D level set, they need seeds as input. We give these three comparison methods additional favors by providing seeds (for each 2D grain segment) at the center of each human annotated grain. For the 3D level set, since we don't have 3D seeds, we put 6, 144 seeds uniformly distributed within the 3D image volume. From Table V, we can see that the proposed MCEWCVT algorithm significantly outperforms all other 2D/3D comparison methods. Note that the second and third best performed methods are random walks and power watersheds, which require initial seeds. The fourth best performed method, mean

TABLE III  
SEGMENTATION ACCURACY ( $F_1$ -MEASURE) OF THE CVT ALGORITHM USING DIFFERENT CLUSTER NUMBER  $L$ . THE BEST RESULT IS SHADED.

$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 10$	$L = 15$	$L = 20$	$L = 25$	$L = 30$
60.36%	66.99%	65.62%	60.38%	60.25%	56.4%	53.74%	53.53%	52.96%
$L = 40$	$L = 50$	$L = 60$	$L = 100$	$L = 150$	$L = 200$	$L = 300$	$L = 500$	$L = 800$
57.24%	56.02%	53.96%	54.54%	51.48%	49.04%	46.69%	42.28%	38.12%

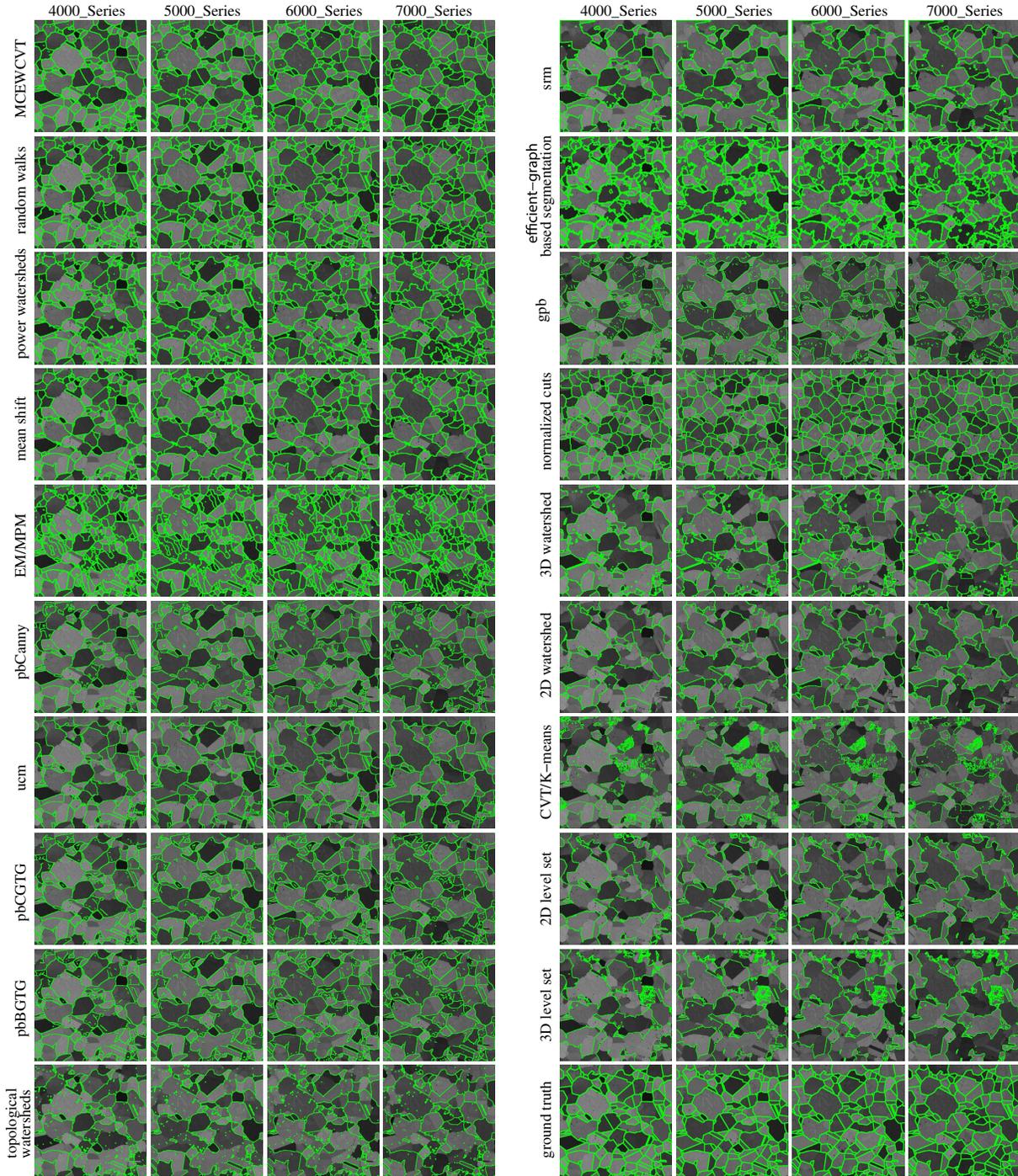


Fig. 8. Visual comparisons of the segmentation results of the proposed MCEWCVT, other comparison methods and ground-truth, on one slice (4 channels).

shift, achieves an accuracy at 86.32%, which is about 7% lower than the proposed MCEWCVT algorithm. The soft-edge detection methods (e.g., pbCanny, ucm) achieves a reasonable

performance, if a proper threshold to the edge probability map can be selected. The EM/MPM is developed for 2D single-channel superalloy image segmentation and its performance

TABLE IV  
PARAMETER SETTINGS FOR THE 2D/3D COMPARISON METHODS OTHER THAN CVT

Methods	Parameter settings
random walks	2D seeds: center of the ground-truth grains; weighting parameter: 90
power watersheds	2D seeds: center of the ground-truth grains
mean shift	spatial bandwidth: 10; range bandwidth: 7; minimum region size: 30
EM/MPM	class: 15 (the maximum allowed in its GUI implementation); EM loops: 20; MPM loops: 20; Beta: 2; Min. variance: 20
pbCanny	derivative computation scale: 2; pb resolution: 100; multiplier for lower hysteresis: 0.33
ucm	oriented probability of boundary: calculated by gpb; output format: 'imageSize'
pbCGTG	radius: [0.01 0.02 0.02 0.02]; number of orientations: 8
pbBGTG	radius: [0.01 0.02]; number of orientations: 8
topological watersheds	filtration: 25
srm	segmentation number control parameter: 128
efficient-graph based	sigma: 0.5; K: 100; min: 20
gpb	resizing factor: 1.0
normalized cuts	number of segments: number of grains in the ground-truth
3D watershed	using 26-connectivity
2D watershed	using 8-connectivity
2D level set	2D seeds: center of the ground-truth grains; PDE: 'minimal variance'; initial level set circle radius: 5; iterations: 30
3D level set	6, 144 seeds evenly distributed in 3D volume; PDE: 'minimal variance'; initial level set circle radius: 5; iterations: 30

TABLE V  
SEGMENTATION ACCURACY OF THE PROPOSED MCEWCVT ALGORITHM AND THE COMPARISON METHODS

Methods	MCEWCVT	random walks	power watersheds	mean shift	EM/MPM
$F_1$ -measure	93.57%	88.87%	88.27%	86.32%	80.82%
Methods	pbCanny	ucm	pbCGTG	pbBGTG	topological watersheds
$F_1$ -measure	80.54%	79.52%	79.51%	78.8%	78.41%
Methods	srm	efficient-graph based	gpb	normalized cuts	3D watershed
$F_1$ -measure	76.36%	75.30%	73.42%	71.19%	70.08%
Methods	2D watershed	CVT/K-means	2D level set	3D level set	
$F_1$ -measure	68.94%	66.99%	66.74%	65.06%	

is much lower than the proposed MCEWCVT algorithm. Table VI shows the  $p$ -values for all the methods. Clearly, all the  $p$ -values are low which indicates the statistical significance of the results shown in Table V. Figure 8 shows a visual comparison between the segmentation results of the MCEWCVT and the comparison methods on a selected image slice. We visualize the segmentation results on all four channels.

As for the running time, the proposed MCEWCVT algorithm takes about 72 hours to segment all 170 slices in a workstation with an AMD Opteron(TM) Processor 6234 at 2.4GHz. While this running time is long, it is still much more efficient than manual segmentation (2D segmentation on each slice followed by corresponding 2D segments across slices to construct a 3D segmentation), which usually takes weeks. In practice, we expect that the running time of the proposed MCEWCVT algorithm can be significantly reduced by using parallel implementation, because in each iteration, the transfers of different voxels into the nearest clusters are independent.

#### IV. EXPERIMENTS ON SYNTHESIZED DATASETS

In the previous section, Table I provides the segmentation accuracy when using different  $L$ ,  $\lambda$  and  $\omega$ . In practice, without the ground-truth segmentation, an important issue is to select appropriate values for  $L$ ,  $\lambda$  and  $\omega$  to segment a superalloy data. Generally, the cluster number  $L$  is related to the number of grains, and the edge neighborhood  $\omega$  is related to the grain size. In this section, we use synthesized superalloy data to justify that the optimal  $L$ ,  $\lambda$  and  $\omega$  values found from one

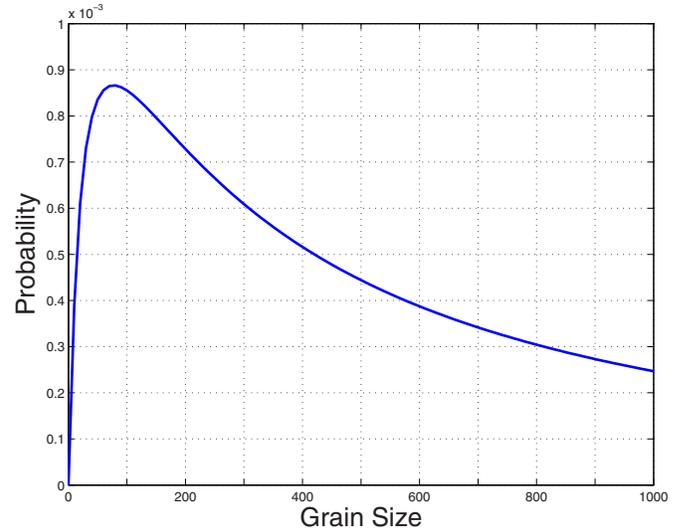


Fig. 9. The Log-Norm distribution of the grain size derived from IN100 dataset.

dataset can be applied to other datasets with similar grain numbers and sizes.

Given the high cost of collecting real data and the intensive labor in annotating the ground-truth segmentation in real data, synthesized data have been widely used in materials science for structural analysis. Recently, DREAM.3D [35] has become a popular tool for constructing synthesized superalloy data. In this paper, we use DREAM.3D to construct two 3D multichannel superalloy datasets: *Synthesized-I* and

TABLE VI  
*p*-VALUES OF THE PROPOSED MCEWCVT ALGORITHM AND THE COMPARISON METHODS

Methods	MCEWCVT	random walks	power watersheds	mean shift	EM/MPM
<i>p</i> -value	0.25%	0.34%	0.43%	0.2%	1.05%
Methods	pbCanny	ucm	pbCGTG	pbBGTG	topological watersheds
<i>p</i> -value	0.43%	0.42%	0.33%	0.36%	0.53%
Methods	srm	efficient-graph based	gpb	normalized cuts	3D watershed
<i>p</i> -value	0.36%	1.65%	0.35%	0.81%	0.3%
Methods	2D watershed	CVT/K-means	2D level set	3D level set	
<i>p</i> -value	0.1%	1.59%	0.26%	0.67%	

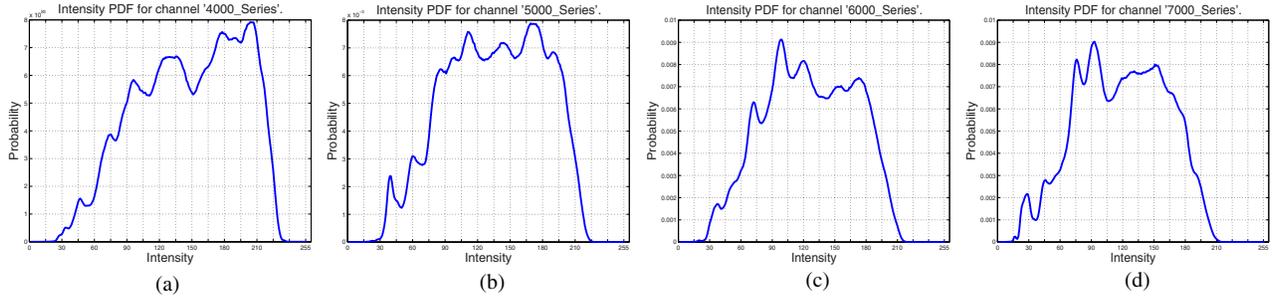


Fig. 10. Intensity distribution in four channels: (a) 4000\_Series; (b) 5000\_Series; (c) 6000\_Series; and (d) 7000\_Series.

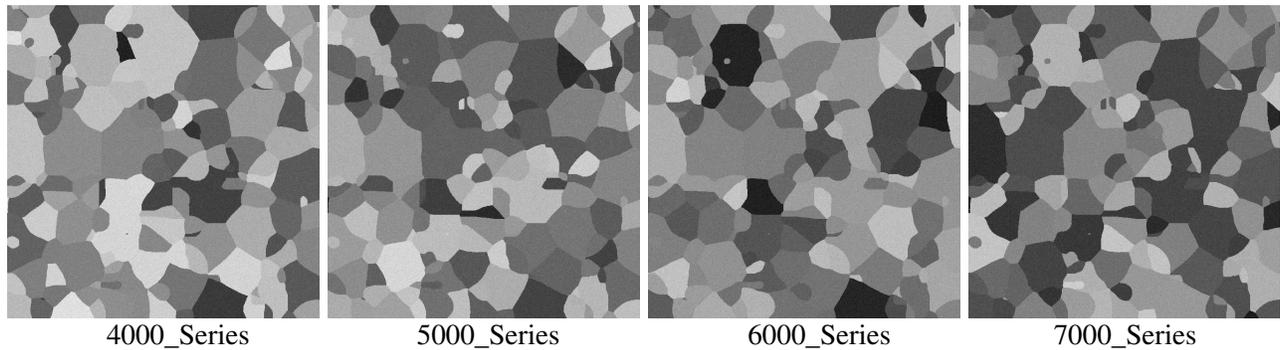


Fig. 11. An illustration of synthesized multichannel superalloy images. Four channels are named as ‘4000\_Series’, ‘5000\_Series’, ‘6000\_Series’ and ‘7000\_Series’, respectively.

TABLE VII  
 SEGMENTATION ACCURACY (*F*<sub>1</sub>-MEASURE) OF MCEWCVT ON SYNTHESIZED-I. THE BEST RESULT IS SHADED.

	$(L, \omega) = (200, 4)$	$(L, \omega) = (200, 6)$	$(L, \omega) = (200, 8)$
$\lambda = 200$	88.68%	98.81%	98.32%
$\lambda = 300$	96.16%	98.9%	97.65%
$\lambda = 400$	98.27%	98.68%	96.96%
	$(L, \omega) = (300, 4)$	$(L, \omega) = (300, 6)$	$(L, \omega) = (300, 8)$
$\lambda = 200$	91.21%	98.91%	98.46%
$\lambda = 300$	97.06%	99%	97.8%
$\lambda = 400$	98.65%	98.8%	97.13%
	$(L, \omega) = (500, 4)$	$(L, \omega) = (500, 6)$	$(L, \omega) = (500, 8)$
$\lambda = 200$	93.84%	98.8%	98.6%
$\lambda = 300$	97.93%	98.67%	97.95%
$\lambda = 400$	98.3%	98.3%	97.1%

TABLE VIII  
 SEGMENTATION ACCURACY (*F*<sub>1</sub>-MEASURE) OF MCEWCVT ON SYNTHESIZED-II. THE BEST RESULT IS SHADED.

	$(L, \omega) = (200, 4)$	$(L, \omega) = (200, 6)$	$(L, \omega) = (200, 8)$
$\lambda = 200$	88.7%	98.9%	98.3%
$\lambda = 300$	96.2%	98.7%	97.7%
$\lambda = 400$	98.2%	98.6%	97.1%
	$(L, \omega) = (300, 4)$	$(L, \omega) = (300, 6)$	$(L, \omega) = (300, 8)$
$\lambda = 200$	91.6%	98.8%	98.3%
$\lambda = 300$	97.3%	99.1%	97.9%
$\lambda = 400$	98.8%	98.7%	97.4%
	$(L, \omega) = (500, 4)$	$(L, \omega) = (500, 6)$	$(L, \omega) = (500, 8)$
$\lambda = 200$	94.2%	98.9%	98.5%
$\lambda = 300$	98.1%	98.5%	97.6%
$\lambda = 400$	98.8%	98.2%	97.3%

*Synthesized-II.* Each synthesized dataset is a  $671 \times 671 \times 671$  volume with around 2,000 grains and the grain sizes are sampled from a Log-Norm distribution derived from the grain sizes in IN100 dataset. We use the Log-Norm distribution as suggested in DREAM.3D [35] because the grain sizes in a superalloy sample can usually be described by such a distribution. Figure 9 shows the Log-Norm distribution

(probability distribution function or PDF) with parameters  $\{\mu = 6.96, \sigma = 1.62\}$  derived from IN100 data. Furthermore, we set an intensity value for each 3D grain in each channel by independently sampling from the intensity distribution in the corresponding channel in IN100 data. Figure 10 shows the intensity distribution in each channel. Finally, we add zero mean, Gaussian white noise of random local variances

(random numbers from a uniform distribution in  $[0,1]$ ) to each pixel independently. In the experiment, this is achieved by directly using the MATLAB function ‘imnoise’. Figure 11 shows one slice of the synthesized data in four channels. Similar as in the IN100 testing, we downsize the image volume size to  $336 \times 336 \times 336$ . The segmentation results on each 2D slices are re-scaled back to the original size (i.e.  $671 \times 671$ ) which are further evaluated against the ground-truth.

We perform the proposed MCEWCVT algorithm on Synthesized-I and Synthesized-II datasets with parameters around  $\{L = 300, \lambda = 300, \omega = 6\}$ , which are the optimal parameters in segmenting IN100 data. Table VII and Table VIII show the segmentation accuracy ( $F_1$ -measure) on Synthesized-I and Synthesized-II, respectively. From the results, we can see that the optimal parameters  $\{L, \lambda, \omega\}$  for segmenting these two synthesized datasets are similar to those for segmenting IN100 data. This suggests that, in practice, we may use the optimal parameters derived from one dataset to segment other superalloy datasets with similar grain number and grain size. Note that, the segmentation accuracies on the synthesized data are higher than those on IN100 data. This is because the synthesized data do not contain complex imaging noise and sub-grain structures as shown in Fig. 4.

## V. CONCLUSION

In this paper, we developed a multichannel edge-weighted centroidal Voronoi tessellation (MCEWCVT) algorithm for 3D superalloy image segmentation. Compared to existing 2D/3D segmentation methods, the proposed algorithm considers both the multichannel intensity similarity and segmentation-boundary smoothness. The effectiveness and robustness of the proposed algorithm is fully justified by experiments on a real Ni-based 4-channel superalloy dataset (IN100) and two synthesized datasets and thorough comparisons to 18 2D/3D existing image segmentation methods.

In the future research on this topic, we expect to develop new extensions to incorporate more specific prior knowledge, such as intensity homogeneity definition, grain geometry and grain layout, to further improve the segmentation accuracy. A preliminary attempt on this direction has been made in [36], which leverages the manual segmentation on a small number of slices to improve the segmentation accuracy on the entire 3D volume.

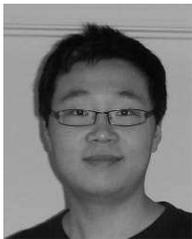
## ACKNOWLEDGMENT

The authors would like to thank Prof. M. De Graef from the Carnegie Mellon University for providing the superalloy images and Q. Zou and C. Qu for helping constructing the manual segmentation of the test images.

## REFERENCES

- [1] R. Reed, *The Superalloys Fundamentals and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2001.
- [2] G. Voort, F. Warmuth, and S. Purdy, *Metallography: Past, Present, and Future (75th Anniversary Volume)*. Philadelphia, PA, USA: ASTM, 1993.
- [3] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [4] F. Meyer, “Topographic distance and watershed lines,” *Signal Process.*, vol. 38, no. 1, pp. 113–125, Jul. 1994.
- [5] R. Nock and F. Nielsen, “Statistical region merging,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1425–1458, Nov. 2004.
- [6] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 1997.
- [7] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
- [8] G. Bertrand, “On topological watersheds,” *J. Math. Imaging Vis.*, vol. 22, nos. 2–3, pp. 217–230, 2005.
- [9] C. Li, C. Xu, C. Gui, and M. Fox, “Level set evolution without re-initialization: A new variational formulation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 430–436.
- [10] G. L  th  n, J. Jonasson, and M. Borga, “Phase based level set segmentation of blood vessels,” in *Proc. Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
- [11] P. Arbel  ez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [12] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Jul. 2001, pp. 416–423.
- [13] P. Arbel  ez, “Boundary extraction in natural images using ultrametric contour maps,” in *Proc. IEEE Comput. Soc. Workshop Perceptual Org. Comput. Vis.*, Jun. 2006, p. 182.
- [14] C. Couprie, L. Najman, L. Grady, and H. Talbot, “Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2009, pp. 731–738.
- [15] L. Grady, “Random walks for image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.
- [16] H. Chuang, L. Huffman, M. Comer, J. Simmons, and I. Pollak, “An automated segmentation for Nickel-based superalloy,” in *Proc. Int. Conf. Pattern Recognit.*, Oct. 2008, pp. 2280–2283.
- [17] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient N-D image segmentation,” *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, Nov. 2006.
- [18] R. Whitaker, “A level-set approach to 3D reconstruction from range data,” *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 203–231, Sep. 1998.
- [19] K. Museth, D. Breen, L. Zhukov, and R. Whitaker, “Level set segmentation from multiple non-uniform volume datasets,” in *Proc. IEEE Conf. Visualizat.*, Nov. 2002, pp. 179–186.
- [20] D. Magee, A. Bulpitt, and E. Berry, “Level set methods for the 3D segmentation of CT images of abdominal aortic aneurysms,” in *Proc. Med. Image Understand. Anal.*, 2001, pp. 141–144.
- [21] F. Vivodtzev, L. Linsen, G. Bonneau, B. Hamann, K. Joy, and B. Olshausen, “Hierarchical isosurface segmentation based on discrete curvature,” in *Proc. Symp. Data Visualizat.*, 2003, pp. 249–258.
- [22] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. New York, NY, USA: Wiley, 2000.
- [23] Q. Du, V. Faber, and M. Gunzburger, “Centroidal Voronoi tessellations: Applications and algorithms,” *Soc. Ind. Appl. Math. Rev.*, vol. 41, no. 4, pp. 637–676, 1999.
- [24] Q. Du, M. Gunzburger, L. Ju, and X. Wang, “Centroidal Voronoi tessellation algorithms for image compression, segmentation, and multi-channel restoration,” *J. Math. Imaging Vis.*, vol. 24, no. 2, pp. 102–119, Mar. 2006.
- [25] J. Wang, L. Ju, and X. Wang, “An edge-weighted centroidal Voronoi tessellation model for image segmentation,” *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1844–1858, Aug. 2009.
- [26] J. Hartigan and M. Wong, “A k-means clustering algorithm,” *J. R. Stat. Soc. Ser. C*, vol. 28, no. 1, pp. 100–108, 1979.
- [27] D. Sparks, “Euclidean cluster analysis,” *J. R. Stat. Soc. Ser. C, Appl. Stat.*, vol. 22, no. 1, pp. 126–130, 1973.
- [28] H. Sp  th, *Cluster Dissection and Analysis. Theory, FORTRAN Programs, Examples*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.
- [29] J. Wang, L. Ju, and X. Wang, “Image segmentation using local variation and edge-weighted centroidal Voronoi tessellations,” *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3242–3256, Nov. 2011.

- [30] M. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *Comput. J.*, vol. 7, no. 2, pp. 152–162, 1964.
- [31] Q. Du and M. Emelianenko, "Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations," *SIAM J. Numer. Anal.*, vol. 44, no. 1, pp. 102–119, 2006.
- [32] M. Emelianenko and A. Rand, "Nondegeneracy and weak global convergence of the Lloyd algorithm in  $R^d$ ," *SIAM J. Numer. Anal.*, vol. 46, no. 3, pp. 1423–1441, 2008.
- [33] (2012, Jul.). *Paraview: Parallel Visualization Application* [Online]. Available: <http://www.paraview.org/>
- [34] (2012, Aug.). *MeshLab: An Open Source, Portable, and Extensible System for the Processing and Editing Of Unstructured 3D Triangular Meshes* [Online]. Available: <http://meshlab.sourceforge.net/>
- [35] (2013, May). *DREAM.3D: Digital Representation Environment for Analyzing Microstructure in 3D* [Online]. Available: <http://dream3d.bluequartz.net/>
- [36] Y. Cao, L. Ju, and S. Wang, "Grain segmentation of 3D superalloy images using multichannel EWCVT under human annotation constraints," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 244–257.
- [37] Y. Cao, L. Ju, Q. Zou, C. Qu, and S. Wang, "A multichannel edge-weighted centroid Voronoi tessellation algorithm for 3D superalloy image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 17–24.



**Yu Cao** (S'11) received the B.S. degree in information and computation science and the M.S. degree in applied mathematics from Northeastern University, Shenyang, China, in 2003 and 2007, respectively. Currently, he is pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA. His current research interests include computer vision, machine learning, and pattern recognition.



**Lili Ju** received the B.S. degree in mathematics from Wuhan University, Wuhan, China, in 1995, the M.S. degree in computational mathematics from the Chinese Academy of Sciences, Beijing, China, in 1998, and the Ph.D. degree in applied mathematics from Iowa State University, Ames, IA, USA, in 2002. From 2002 to 2004, he was an Industrial Post-Doctoral Researcher with the Institute of Mathematics and its Applications, University of Minnesota, Minneapolis, MN, USA. He joined the University of South Carolina, Columbia, SC, USA, in 2004, where he is currently a Professor of mathematics. He has served an Associate Editor for the *SIAM Journal on Numerical Analysis* since 2012. His current research interests include numerical analysis, image processing, ice and ocean modeling, mesh generation and optimization, and parallel computing.



**Youjie Zhou** (S'13) is currently pursuing the Ph.D. degree in computer science and engineering with the University of South Carolina, Columbia, SC, USA, as a Research Assistant with the Computer Vision Laboratory. His current research interests include computer vision, machine learning, and large-scale multimedia analysis. He received the B.S. degree in software engineering from East China Normal University (ECNU), Shanghai, China, in 2010. From 2007 to 2010, he was a Research Assistant with the Institute of Massive Computing, ECNU, where he worked on multimedia news exploration and retrieval.



**Song Wang** (M'02–SM'13) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, in 2002. From 1998 to 2002, he was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently an Associate Professor. His current research interests include computer vision, medical image processing, and machine learning. He is currently serving as the Publicity/Web Portal Chair of the Technical Committee of Pattern Analysis and Machine Intelligence, the IEEE Computer Society, and an Associate Editor of *PATTERN RECOGNITION LETTERS*. He is a member of the IEEE Computer Society.