2015

Propagated image Segmentation Using Edge-Weighted Centroidal Voronoi Tessellation based Methods

Youjie Zhou
University of South Carolina

Propagated Image Segmentation Using Edge-Weighted Centroidal Voronoi Tessellation Based Methods

by

Youjie Zhou

Bachelor of Engineering
East China Normal University 2010

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2015

Accepted by:

Song Wang, Major Professor

Lili Ju, Co-Advisor and External Examiner

Michael N. Huhns, Committee Member

Srihari Nelakuditi, Committee Member

Yan Tong, Committee Member

Lacy Ford, Senior Vice Provost and Dean of Graduate Studies

Dedication

To my family.

Acknowledgments

There are many people who have helped me in the journey towards a PhD. First and foremost, I would like to thank, Dr. Song Wang and Dr. Lili Ju, as my supervisor and co-advisor respectively, for guiding the right way of research for me. I would also like to express my thanks to my committee, Dr. Huhns, Dr. Nelakuditi and Dr. Tong, who have provided constructive feedback on this manuscript. Also to Dr. Jeff Simmons as the research collaborator who has spent a lot of efforts on discussing about this work.

Many thanks to colleagues in the lab and many friends who have made the graduate school life much easier and more fun, especially to Dr. Yu Cao, Dr. Jarrell Waggoner, Dr. Dhaval Salvi, Dr. Ping Liu, Yuewei Lin, Dazhou Guo, Jun Zhou, Xiaochuan Fan, Kang Zheng, Yang Mi, Hongkai Yu, Kareem Ezz El-Deen, Shizhong Han, Zibo Meng and James O'Reilly.

I would like to thank my previous mentors at early stages of this research adventure, including Dr. Hangzai Luo and Dr. Jianping Fan, without their careful training, there would be more detours along the journey.

The last but most importantly, I would like to thank my parents, my girlfriend, Ying Zhu, and the whole family for their constant love and countless support.

Abstract

Propagated image segmentation is the problem of utilizing the existing segmentation of an image for obtaining a new segmentation of, either a neighboring image in a sequence, or the same image but in different scales. We name these two cases as the inter-image propagation and the intra-image propagation respectively. The inter-image propagation is particularly important to material science, where efficient and accurate segmentation of a sequence of 2D serial-sectioned images of 3D material samples is an essential step to understand the underlying micro-structure and related physical properties. For natural images with objects in different scales, the intra-image propagation, where segmentations are propagated from the finest scale to coarser scales, is able to better capture object boundaries than single-shot segmentations on a fixed image scale.

In this work, we first propose an inter-image propagation method named *Edge-Weighted Centroid Voronoi Tessellation with Propagation of Consistency Constraint* (CCEWCVT) to effectively segment material images. CCEWCVT segments an image sequence by repeatedly propagating a 2D segmentation from one slice to another, and in each step of this propagation, we apply the proposed consistency constraint in the pixel clustering process such that stable structures identified from the previous slice can be well-preserved. We further propose a non-rigid transformation based association method to find the correspondence of propagated stable structures in the next slice when the inter-image distance becomes large. We justify the effectiveness of the proposed CCEWCVT method on 3D material image sequences, and we compare its performance against several state-of-the-art 2D, 3D, propagated segmen-

tation methods. Then for the intra-image propagation, we propose a superpixel construction method named *Hierarchical Edge-Weighted Centroidal Voronoi Tessellation* (HEWCVT) to accurately capture object boundaries in natural images. We model the problem as a multilevel clustering process: superpixels in one level are clustered to obtain larger size superpixels in the next level. The clustering energy involves both color similarities and the proposed boundary smoothness of superpixels. We further extend HEWCVT to obtain supervoxels on 3D images or videos. Both quantitative and qualitative evaluation results on several standard datasets show that the proposed HEWCVT method achieves superior or comparable performances to other state-of-the-art methods.

Table of Contents

Dedication iii

Acknowledgments iv

Abstract v

List of Tables ix

List of Figures x

Chapter 1 Introduction 1

1.1 Research Motivation 1

1.2 Research Contribution 4

1.3 Dissertation Outline 5

Chapter 2 Related Work 7

2.1 Image Segmentation Methods without Propagation 7

2.2 Related Inter-Image Propagation Methods 12

2.3 Related Intra-Image Propagation Methods 13

Chapter 3 Inter-Image Propagation 15

3.1 Stable Segment Structure and Its Propagation 16

3.2 Cluster Initialization Satisfying Consistency Constraint 18

3.3 Construction of EWCVT with Consistency Constraint 19

3.4 Segment Consistency under Long Distance Propagation 21
3.5 Experiments 31
3.6 Discussion 51

Chapter 4 Intra-Image Propagation 52
4.1 Hierarchical Edge-Weighted Centroidal Voronoi Tessellation 53
4.2 Simple-Connectivity Enforcement 58
4.3 Complexity and Convergence Analysis 58
4.4 Extension to Supervoxels 60
4.5 Experiments 61
4.6 Discussion 81

Chapter 5 Conclusion 82

Bibliography 83

Related Publications 88

List of Tables

Table 3.1 Quantitative comparison of 2D/3D/Propagated segmentation methods on the IN100 dataset. 33

Table 3.2 Comparison on running times of 2D/3D/Propagated segmentation methods on the IN100 dataset. 34

Table 3.3 Quantitative comparison of 2D/3D/Propagated segmentation methods on the IN100-300 dataset. 44

Table 3.4 Quantitative comparison of 2D/3D/Propagated segmentation methods on the IN100-900 dataset. 44

Table 4.1 Average running time of different superpixel/supervoxel algorithms on several image/video datasets. 64

Table 4.2 Parameter settings (ω, θ) that achieve the highest performance on each hierarchy level in the grid search on the validation sets. . . 65

Table 4.3 Class-average segmentation accuracies on the MSRC dataset and the PASCAL VOC2007 dataset using superpixels constructed by different algorithms. 72

List of Figures

Figure 1.1 Microscopic grain images and segmentation. From left to right: original image slices with ground-truth grain boundaries, segmentation using a 2D method, segmentation using a 3D method, and segmentation using the proposed algorithm, respectively. . . . 2

Figure 1.2 Sample results of the proposed HEWCVT method on 2D images: the original image (left column), superpixels constructed on the finest level of hierarchy (middle column) and superpixels constructed on the highest level of hierarchy (right column). Superpixels of objects are visualized by colorful patches. The hierarchical nested relations among superpixels can be easily verified. 3

Figure 2.1 Boundary smoothness measurement for a pixel P. Each pixel is visualized as a polygon and its shape stands for the pixel's current cluster assignment. 10

Figure 2.2 Overview process of the EWCVT construction algorithm based on the k-means type techniques. 11

Figure 2.3 Image segmentation boundaries can be determined as the boundaries among constructed clusters. 11

Figure 3.1 Overview of the proposed propagation method for constructing segmentations for an image sequence. 16

Figure 3.2 An illustration of an incorrect clustering initialization which may not be able to preserve the propagated segment structure. See text for detailed description. 18

Figure 3.3 An Illustration of segment association problem for a set of segments constructed on an image sequence. Our goal is to find the correspondence among segments constructed on neighboring image slices, and eventually reconstruct the underlying 3D object structure by associating all segments together. 22

Figure 3.4 An illustration of correspondence problem for long distance propagation. Given a large inter-slice distance, it's hard to find the correspondence between propagated stable segment structure from previous image slice I^{i-1} and the segment structure on image slice I^i. 22

Figure 3.5 An illustration of the Kalman filter for recursively modeling target movement using observed target locations and predicting missing target locations on an image sequence. 24

Figure 3.6 Divide-and-conquer approach to find associations between predicted targets and observed ones. On the left, we first divide predictions into compact groups, and then, on the right, for each group, the association is constructed through the non-rigid transformation based algorithm. Later, group-wise associations are combined together. See text for detailed description. 27

Figure 3.7 Sliding window search for constructing the observation association. Red window on the right indicates the set of associated observations that is able to achieve the smallest TPS energy. . . . 27

Figure 3.8 Illustrations of the reconstructed 3D structures. The structures are obtained by finding the correspondence among a set of 2D segmentations using proposed non-rigid transformation based association algorithm. 30

Figure 3.9 An illustration of quantitative comparison with 2D/3D/Propagated segmentation methods on the IN100 dataset. 33

Figure 3.10 Qualitative comparisons of the four 2D/Propagated image segmentation methods (Proposed, EWCVT, GraphBased and gPb) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods. 36

Figure 3.11 Qualitative comparisons of the four 2D/Propagated image segmentation methods (Proposed, MeanShift, SRM and NormalizedCuts) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods. . . 37

Figure 3.12 Qualitative comparisons of the three 3D/Propagated image segmentation methods (Proposed, MCEWCVT and 3D Levelset) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods. 38

Figure 3.13 Qualitative comparisons of the three 3D/Propagated image segmentation methods (Proposed, 3D Watershed and StreamGBH) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods. 39

Figure 3.14 The F-score on each slice (left) during propagation. The sharp drop is caused by a corrupted slice (right) in the original IN100 dataset. 40

Figure 3.15 Illustrations of constructed 2D image slices of serial sectioned 3D synthesized grains. 42

Figure 3.16 A set of image patches randomly sampled from the IN100 dataset. 43

Figure 3.17 2D image slices of serial sectioned 3D synthesized grains, after filling with sampled patches. 43

Figure 3.18 An illustration of quantitative comparison with 2D/3D/Propagated segmentation methods on the IN100-300 dataset. 45

Figure 3.19 An illustration of quantitative comparison with 2D/3D/Propagated segmentation methods on the IN100-900 dataset. 46

Figure 3.20 Illustrations of the fiber dataset. For each image slice, we also visualize provided fiber detections right next to each image, highlighted by red ellipses. 47

Figure 3.21 Quantitative comparison of different multi-target tracking methods for long distance segment structure propagation. MOTA is shown in left, and average running time (after log transformation) is shown in right. 50

Figure 3.22 Quantitative comparison of different multi-target tracking methods for long distance segment structure propagation on a few individual tiles. 50

Figure 4.1 Overview of the proposed HEWCVT method for constructing superpixels/supervoxels in multiple scales. 52

Figure 4.2 Boundary smoothness measurement illustrations. Dash lines are cluster boundaries. Pink curve indicates the local neighborhood area for smoothness measurement. 53

Figure 4.3 An illustration of energy convergence of the proposed HEWCVT, with $\theta = 1.5$, when constructing superpixels on a sample image. . 59

Figure 4.4 Sample images and their human annotated boundaries for superpixel and supervoxel evaluations. 62

Figure 4.5 Illustrations of three superpixel/supervoxel evaluation metrics. . . 63

Figure 4.6 Superpixel performance, in terms of boundary recall, undersegmentation error and segmentation accuracy, of the proposed HEWCVT under different values of ω and θ in one hierarchy level on the image validation set. Better view in color. 67

Figure 4.7 Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the BSDS300 dataset. 69

Figure 4.8 Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the W1 dataset. 70

Figure 4.9 Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the W2 dataset. 70

Figure 4.10 Qualitative comparisons of the four superpixel methods (HEWCVT, VCells, GraphCut, SLIC, LRW) on two images from the BSDS300 dataset. The numbers at the left indicate the desired number of superpixels. Better view in color. 71

Figure 4.11 Qualitative comparisons of the four superpixel methods (HEWCVT, VCells, GraphCut, SLIC, LRW) on two images from the W1 and W2 datasets. The numbers at the top indicate the desired number of superpixels. Better view in color. 73

Figure 4.12 Sample images, human annotated semantic pixel labels, and statistics for the MSRC dataset and the PASCAL VOC2007 dataset. 74

Figure 4.13 Illustrations of quantitative evaluation on the semantic image segmentation task using superpixels constructed by different algorithms. Results for the MSRC dataset are shown in the left, and results for the PASCAL VOC2007 dataset are shown in the right. 75

Figure 4.14 Supervoxel evaluation (w/ connectivity enforcement) of GBH, SWA, and HEWCVT on the Xiph.org dataset. 76

Figure 4.15 Supervoxel evaluation (w/o connectivity enforcement) of GBH, SWA, and HEWCVT on the Xiph.org dataset. 76

Figure 4.16 Qualitative comparisons of the three supervoxel methods (HEWCVT, GBH, SWA) on four videos. On each video, the number of supervoxels generated by these three methods are similar for fairer comparison. Neighboring supervoxels are shown in different color. 78

Figure 4.17 An illustration of the dis-connectivity issue in GBH. Constructed supervoxels in two adjacent video frames are visualized with specific colors. Each supervoxel from GBH actually contains many disjoint fragments. Highlighted by black bounding boxes. Better view in color. 79

Chapter 1

Introduction

1.1 Research Motivation

Given high resolution and complex 2D/3D images with millions and millions pixels/voxels, segmentation methods that directly obtain single segmentation on the whole image are not as efficient as being applied on small images. Due to its efficiency, propagated image segmentation methods have attracted much attention recently. Efficient segmentation on a new image is achieved by utilizing existing segmentations from reference images. Based on the relationship between the new image and the reference images, we divide propagated segmentation problem into two categories: 1) Inter-image propagation where reference images and new images are from a same sequence of correlated images. Existing segmentations are propagated and utilized along the sequence. The final goal is to obtain segmentations for each image in the sequence. 2) Intra-image propagation where reference images and new images are the same but in different representations, e.x., different scales. In this work we focus on intra-image propagation among different scales. For each category, we analyze their importance in real applications below respectively.

Since 3D images can be serial-sectioned and represented by 2D image sequences, inter-image propagation is particularly important to 3D image segmentation because of its efficiency. In this work we choose 3D superalloy material image segmentation as an application and address the importance of inter-image propagated segmentation on it.

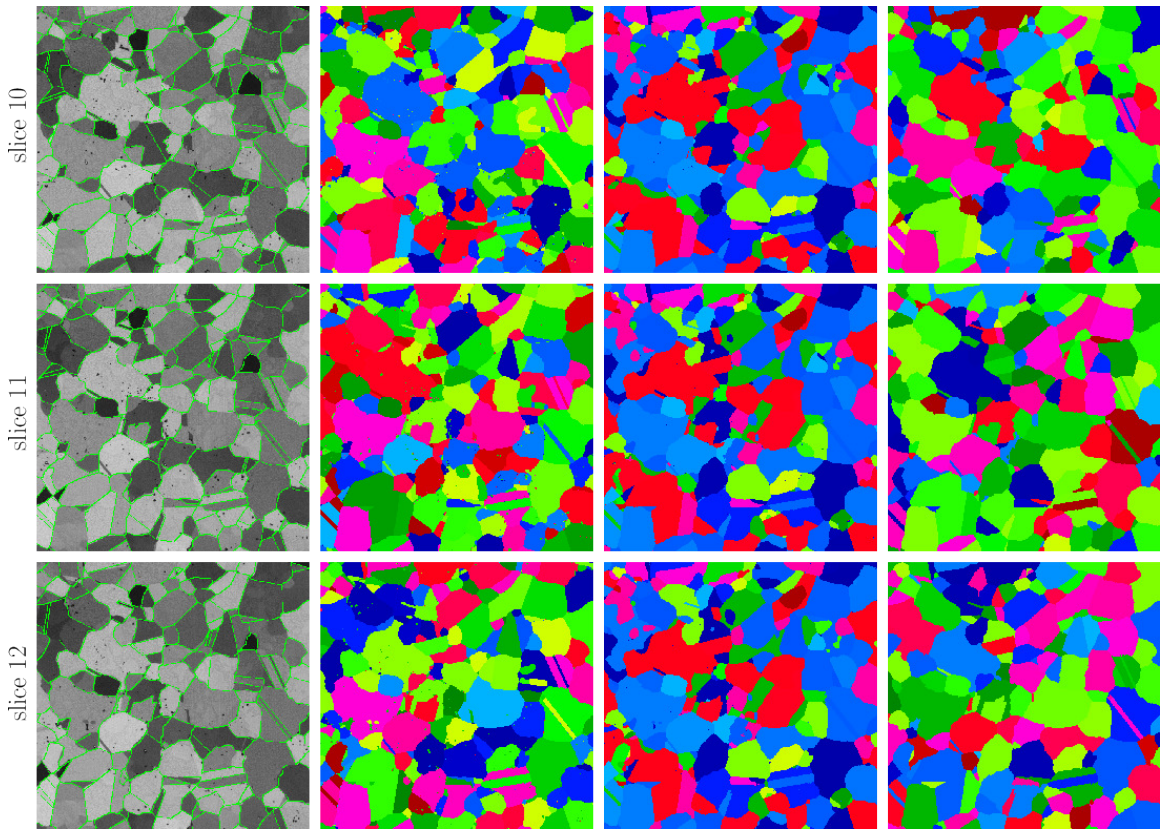


Figure 1.1: Microscopic grain images and segmentation. From left to right: original image slices with ground-truth grain boundaries, segmentation using a 2D method, segmentation using a 3D method, and segmentation using the proposed algorithm, respectively.

In material science, superalloy materials have been widely used in both commercial and military applications [35, 40] because of their excellent tensile strength and resistance to creep under high temperatures [49]. Such physical properties are mainly determined by the underlying micro-structures of superalloy samples, which are usually in the form of set of grains [35]. These grains are too small to be visible to human vision. In practice, high-performance electron microscopy is usually used to get the 2D surface image of the material sample [35]. To better identify the micro-structures, various chemicals, like acids, may be applied to the material surface to highlight the grain boundaries. In addition, to achieve the underlying 3D grain structures, a serial-sectioning technique is usually applied to unveil the internal structure of the material to the microscopy [49]. Example image slices from a 2D serial sec-

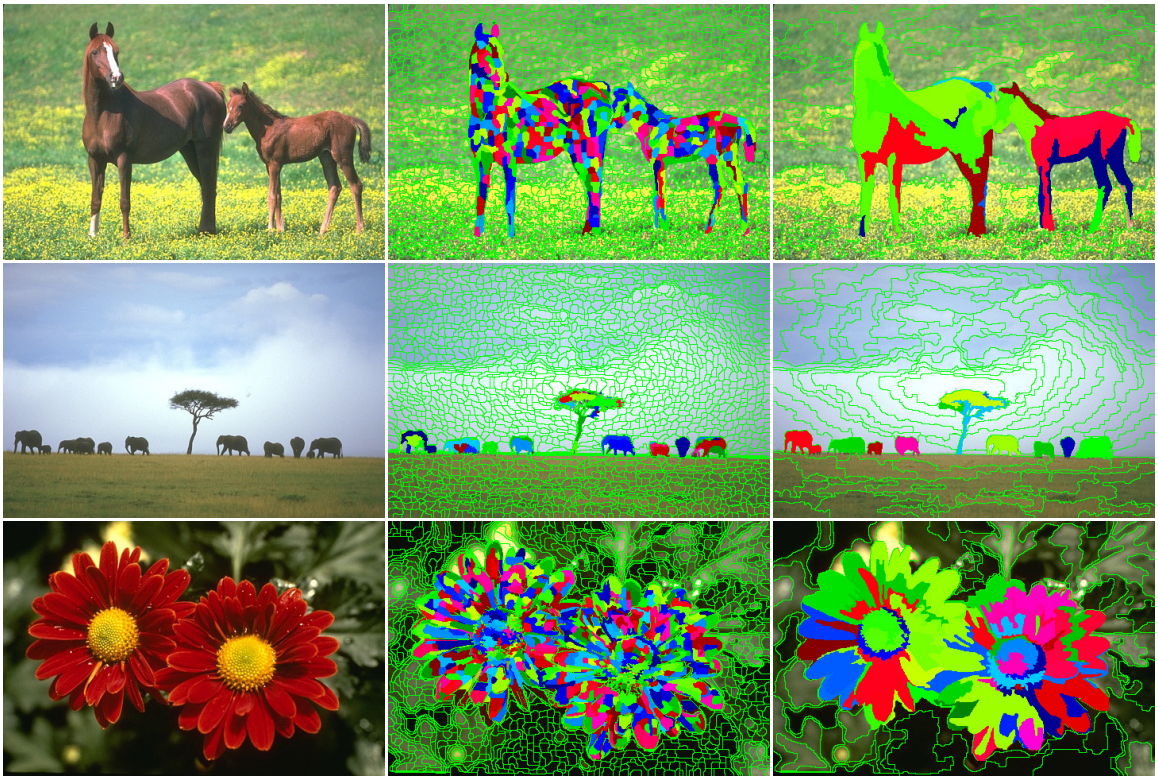


Figure 1.2: Sample results of the proposed HEWCVT method on 2D images: the original image (left column), superpixels constructed on the finest level of hierarchy (middle column) and superpixels constructed on the highest level of hierarchy (right column). Superpixels of objects are visualized by colorful patches. The hierarchical nested relations among superpixels can be easily verified.

tion of a 3D superalloy material sample are shown in the first column of Figure 1.1 where each cell is a grain. In order to reveal grain structures from such 2D image slices, material scientists must manually annotate the grain boundaries on each of the 2D slices, and then correspond 2D grains across all the slices to reconstruct the 3D grain structure. This manual annotation process is tedious, time-consuming, and often prone to error, given a large number of grains and serial-sectioned slices in a high-resolution 3D superalloy image. This calls for efficient and effective automatic grain segmentation, which not only captures the grain boundaries accurately, but also completes quickly. We address this problem by proposing an inter-image propagated segmentation method on the sequence of 2D slices.

For the intra-image propagation in different image scales, the advantage is that it

allows us to capture and discover object boundaries under different scales. Further, we can represent the correspondence among segmentations under different scales as a nested hierarchy structure. In this work, we model the intra-image propagated segmentation as constructing multiscale superpixels. Many vision applications require the use of multiscale superpixels with different coarse levels to better infer the high-level structural information [20, 25, 28, 41, 52, 46, 23]. Multiscale superpixels can usually be obtained by varying certain configurations, such as the number of or the average size of superpixels. However, simply varying these configurations may not generate multiscale superpixels with boundary consistency, i.e., the boundaries in a coarser level may not be drawn from the boundaries in a finer level. This way, the superpixels in different scales may not show a hierarchical nested relations, which is important for inferring high-level structural information [25, 11, 41, 20, 28]. Thus, we would like to address this problem by using intra-image propagated method for constructing multiscale superpixels.

1.2 Research Contribution

This work focuses on developing propagated methods for efficient image segmentation based on the Edge-Weighted Centroidal Voronoi Tessellation model.

We have proposed an inter-image propagated method for 3D grain images. 3D grain images are represented by a sequence of serial-sectioned 2D images and an initial 2D segmentation is propagated along this sequence. This method propagates inter-slice consistency constraints for accurate and fast segmentation of 3D grains in superalloy images. Specifically, the proposed method performs a 2D-constrained EWCVT segmentation on each image slice using the segmentation of the previous slice as the initialization, and during the clustering process the stable grain structure of the previous slice is also preserved. On the first image slice, we use the segmentation result of the EWCVT algorithm as the initialization. This way, the proposed method

obtains a segmentation on the new image slice while simultaneously preserving the segment correspondence with the previous image slice.

We have also proposed an intra-image propagated method for natural images/ videos. We construct multiscale superpixels/supervoxels to capture object boundaries. In this method, superpixels/supervoxels in a finer level is clustered to achieve superpixels/supervoxels in a new coarser level. In the finest level, all the image pixels/voxels are taken as the entities for HEWCVT clustering. This iterative clustering process guarantees the hierarchical nested relations across different levels. In HEWCVT method, the clustering energy consists of not only a term that measures the color/feature between superpixels/supervoxels, but also a proposed edge term that measures the boundary smoothness of the obtained superpixels/supervoxels. With this edge term, the proposed HEWCVT method is able to produce superpixel/supervoxel boundaries better aligned with the underlying structural boundaries in each level. Examples of superpixels on 2D images are shown in Figure 1.2. In the experiments, we justify the proposed method by qualitatively and quantitatively comparing its performance with the performance of several other state-of-the-art superpixels/supervoxels methods on three standard image/video datasets.

1.3 Dissertation Outline

This dissertation is organized as follows:

Chapter 2 introduces the general approach for propagated image segmentation, i.e., incorporating propagation properties into existing image segmentation methods. Related image segmentation methods have been reviewed, including methods without propagation, inter-image propagation methods and intra-image propagation methods. As the proposed methods are based on the Edge-Weighted Centroid Voronoi Tessellation method, thus we also revisit its algorithms and 3D extensions.

Chapter 3 introduces the proposed inter-image propagated segmentation method

for 3D grain images. We define the consistency constraint as preserving the stable segment structure during the propagation and discuss proposed cluster initialization satisfying such constraint. Algorithms to construct EWCVT with consistency constraint are developed. In the experiments, we apply the proposed method on a serial-sectioned 3D grain image and compare its performance, in terms of both segmentation accuracy and time efficiency, with the performance of several state-of-the-art 2D, 3D and propagation methods.

Chapter 4 introduces the proposed intra-image propagated segmentation method for constructing multiscale superpixels for natural images. The definition of proposed edge smoothness energy for superpixel is discussed and iterative algorithms to construct superpixels under different scales are developed. We also describe how to extend proposed method to multiscale supervoxel construction for videos. In the experiments, we justify the proposed method by qualitatively and quantitatively comparing its performance with the performance of several other state-of-the-art superpixels/supervoxels methods on three standard image/video datasets.

Finally, the summary of our achievements and potential future work are discussed in Chapter 5.

Chapter 2

Related Work

Propagated image segmentation methods usually extend existing image segmentation methods with considering propagation related properties. For example, for the inter-image propagation, the consistency among segmentations on neighboring images can be considered as the propagation constraint and may be incorporated into existing segmentation methods. Similar to the intra-image propagation, segmentation obtained in one specific scale may be directly propagated for obtaining segmentations on other scales using existing segmentation methods.

Therefore in this chapter, we first revisit several existing 2D and 3D image segmentation methods without propagation in Section 2.1, including the Edge-Weighted Centroid Voronoi Tessellation method that are utilized by the proposed methods. Then we introduce a few state-of-the-art inter-image and intra-image propagation methods in Section 2.2 and Section 2.3 respectively.

2.1 Image Segmentation Methods without Propagation

Graph-Based Image Segmentation (GB) In [18], an efficient graph-based image segmentation method was proposed. The graph is initialized over the entire image, where each pixel being its own unique region. Then a greedy algorithm iteratively traverses the region edges in a sorted order by increasing edge weights (measured the dissimilarity between two regions), and merge two regions if their edge weight is smaller than the internal variation of both regions. Once two regions are merged, their internal variation is also updated. The internal variation of a region is defined

as the largest weight in the minimum spanning tree (MST) of the contained components. Since the internal variation of a region with a single node is zero, then only singleton regions can cause an initial merge. To alleviate that, the relaxed internal variation $\mathrm{RInt}(R)$ is defined as

$$\mathrm{RInt}(R) = \mathrm{Int}(R) + \delta(R), \text{ with } \delta(R) = \frac{\tau}{|R|} \tag{2.1}$$

where τ is a constant parameter and $|R|$ is the number of pixels in region R. The τ controls the granularity of the final segmentation, where a larger τ usually results in larger regions but with higher probabilities of containing incorrect segmentation boundaries.

Since the simplicity and efficiency of the proposed graph-based framework, GB has been widely applied for 2D image segmentation [18], and recently several inter-image propagated segmentation methods extend it for obtaining 3D and video segmentations [51, 50, 22]. Later in this section we will introduce its related extensions.

Global Probability of Boundary (gPb) A simple but effective boundary detector named global probability of boundary was proposed in [3]. gPb estimates the posterior probability of boundaries passing through the center of local patch based features. Several color and gradient features have been extracted on the input image. The boundary estimation is achieved by a simple logistic regression classifier.

Edge-Weighted Centroid Voronoi Tessellation (EWCVT) [44] proposed an edge energy for the k-means clustering based segmentation method. By utilizing the proposed edge energy, their method is able to directly measure the smoothness of obtained boundaries, and thus achieves better results than the original k-means method. In this work, two proposed propagated segmentation methods are based on the EWCVT, thus we quickly revisit the EWCVT method below.

Let $\mathbb{U} = \{\vec{u}(i,j) \mid (i,j) \in \mathbb{I}\}$ denote the set of color or feature vectors of a 2D image $\mathbb{I}$ (extension to 3D images will be discussed in Section 4.4), where $\vec{u}$ is the color/feature function associated with $\mathbb{I}$. In the experiments, we use the `Lab` color feature. For L arbitrary color vectors $\mathcal{W} = \{\vec{w}_l\}_{l=1}^{L}$ (called *generators*), the corresponding *Voronoi tessellation* of $\mathbb{U}$ is defined as $\mathcal{V} = \{V_l\}_{l=1}^{L}$ such that $V_l = \{\vec{u}(i,j) \in \mathbb{U} \mid \|\vec{u}(i,j) - \vec{w}_l\| < \|\vec{u}(i,j) - \vec{w}_m\|,\ m = 1, \ldots, L \text{ and } m \neq l\}$, where $\|\cdot\|$ is a distance function defined on $\mathbb{U}$. Given a weight or density function ρ defined on each pixel of $\mathbb{I}$, we can further define the centroid (i.e., the center of mass) of each Voronoi region V_l as $\vec{w}_l^*$ such that $\vec{w}_l^* = \min_{\vec{w} \in V_l} \sum_{\vec{u}(i,j) \in V_l} \rho(i,j) \|\vec{u}(i,j) - \vec{w}\|^2$.

If the generators $\{\vec{w}_l\}_{l=1}^{L}$ of the Voronoi regions $\{V_l\}_{l=1}^{L}$ of $\mathbb{U}$ are the same as their corresponding centroids, i.e.,

$$\vec{w}_l = \vec{w}_l^*,\ l = 1, \ldots L,$$

then we call the Voronoi tessellation $\{V_l\}_{l=1}^{L}$ a *centroidal Voronoi tessellation* (CVT) of $\mathbb{U}$. Since each Voronoi region V_l stands for a cluster in the color space we can easily construct a corresponding partition of the 2D image $\mathbb{I}$ using the correspondence between pixel indices and color vectors through $\vec{u}$. Let $\mathcal{C} = \{C_l\}_{l=1}^{L}$ denote a clustering of the physical space of the image $\mathbb{I}$, then the CVT clustering energy can be defined as

$$E_{cvt}(\mathcal{C}, \mathcal{W}) = \sum_{l=1}^{L} \sum_{(i,j) \in C_l} \rho(i,j) \|\vec{u}(i,j) - \vec{w}_l\|^2. \tag{2.2}$$

The construction of CVTs often can be viewed as a clustering energy minimization problem, i.e., solving $\min_{(\mathcal{C},\mathcal{W})} E_{cvt}(\mathcal{C}, \mathcal{W})$. The Lloyd method [14, 15] (equivalent to the weighted k-means) has been widely used to compute CVTs, which is basically iterations between constructing Voronoi regions and centroids. Assume that the Euclidean distance is used for the color space, then we simply have the centroid of the cluster C_l as $\vec{w}_l^* = \sum_{(i,j) \in C_l} \rho(i,j) \vec{u}(i,j) / \sum_{(i,j) \in C_l} \rho(i,j)$.

In order to enforce the smoothness of segment boundaries, a special edge energy was proposed and added into the clustering energy [44, 45]. Specifically, let us define

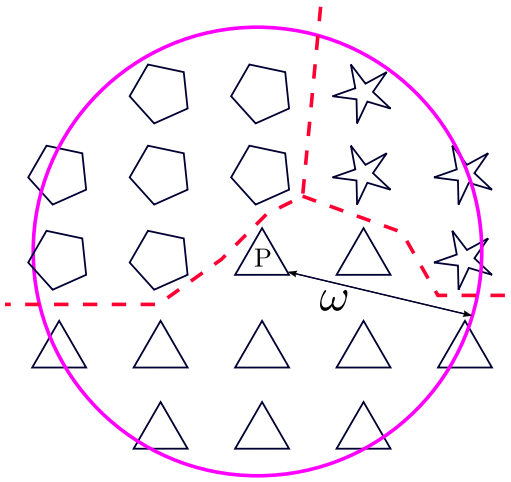


Figure 2.1: Boundary smoothness measurement for a pixel P. Each pixel is visualized as a polygon and its shape stands for the pixel's current cluster assignment.

an indicator function $\chi(i,j) : \mathbb{N}_\omega(i,j) \to \{0,1\}$ on the neighborhood of pixel (i,j) with radius ω as

$$\chi_{(i,j)}(i',j') = \begin{cases} 1 & \text{if } \pi(i',j') \neq \pi(i,j) \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

where $\pi(i,j)$ tells the cluster index that (i,j) belongs to. Then the edge energy is defined as

$$E_{edge}(\mathcal{C}) = \sum_{(i,j)\in\mathbb{I}} \sum_{(i',j')\in\mathbb{N}_\omega(i,j)} \chi_{(i,j)}(i',j'). \tag{2.4}$$

Figure 2.1 illustrates the boundary smoothness measurement on a single pixel. It has been shown in [44] that $E_{edge}(\mathcal{C})$ is proportional to the total length of boundaries in $\mathcal{C}$ in the limit. Finally, the edge-weighted CVT clustering energy can be defined as

$$E_{ewcvt}(\mathcal{C},\mathcal{W}) = E_{cvt}(\mathcal{C},\mathcal{W}) + \lambda E_{edge}(\mathcal{W}) \tag{2.5}$$

where λ is a weight parameter balancing the clustering energy and the edge energy. Construction of EWCVTs is equivalent to solving the minimization problem $\min_{(\mathcal{C},\mathcal{W})} E_{ewcvt}(\mathcal{C},\mathcal{W})$. An edge-weighted distance function from a pixel (i,j) to a cluster center (generator) $\vec{w}_k$ was derived for the energy E_{ewcvt} as

$$dist((i,j),\vec{w}_k) = \sqrt{\rho(i,j)\|\vec{u}(i,j) - \vec{w}_k\|^2 + 2\lambda\tilde{n}_k(i,j)} \tag{2.6}$$

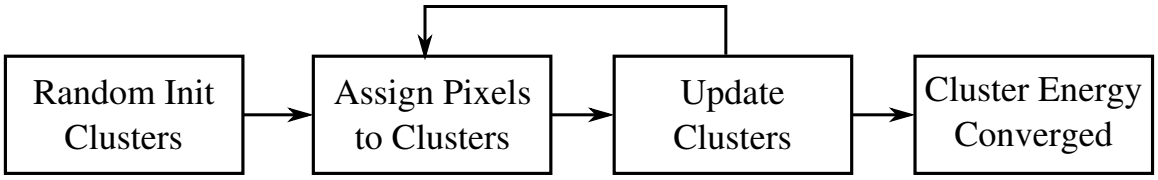


Figure 2.2: Overview process of the EWCVT construction algorithm based on the k-means type techniques.

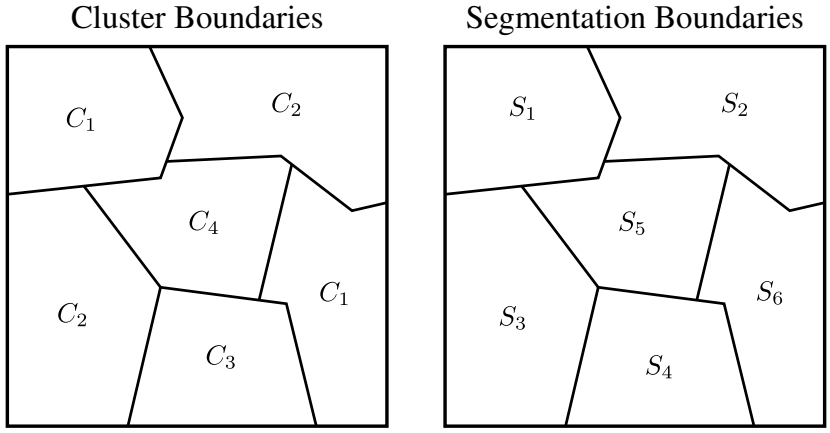


Figure 2.3: Image segmentation boundaries can be determined as the boundaries among constructed clusters.

where $\tilde{n}_k(i,j) = |\mathbb{N}_\omega(i,j)| - n_k(i,j) - 1$ with $n_k(i,j) = \sum_{(i',j')\in\mathbb{N}_\omega(i,j)} \pi(i',j') \neq k$.

Based on the above distance function, a few efficient algorithms for constructing EWCVTs are suggested in [44] based on the k-means type techniques. In general, the construction algorithm contains two steps:

1. Assignment step: pixels will be assigned to the closed cluster based on the defined EWCVT distance (Eq. 2.6).

2. Update step: based on the assignment results, cluster centers or generators will be updated according to the contained pixels.

The whole construction process is illustrated in Figure 2.2. Finally the image segmentation boundaries can be determined as the cluster boundaries, as shown in Figure 2.3.

EWCVT 3D Extensions Recently, two automatic 3D EWCVT-based methods (MCEWCVT and CMEWCVT) [7, 5] were proposed for superalloy image segmentation. These two clustering methods perform 3D segmentation by minimizing an energy function which considers both voxel intensity similarity and the smoothness of the segmentation boundaries. During the energy minimization, these two 3D clustering methods enumerate every voxel and collect smoothness information around the neighborhood of each voxel, which leads to very high algorithmic complexity. Additionally, the L^{∞}-norm used in these two methods makes the centroids difficult to calculate. Although these two EWCVT-based methods capture grain boundaries accurately, as claimed in [7], they typically require 10 hours to segment a 3D $671 \times 671 \times 170$ image. Furthermore, strong noise in the dense 3D image space is usually grouped into separate clusters.

2.2 Related Inter-Image Propagation Methods

Stream Hierarchical Graph-Based Image Segmentation (StreamGBH) In [51], the StreamGBH algorithm is proposed for segmenting streaming videos. StreamGBH segments a sequence of video frames by merging an over-segmentation on each frame, guided by the segmentation on the previous frame. Without specific constraints on boundary smoothness, StreamGBH usually generates highly fragmented and scattered segments.

Highly Consistent Sequential Segmentation In [13], the authors propagate the result of previous video frame as the initialization for segmenting the current frame. After a modified active-contour based segmentation, the algorithm further merges and splits segments according to the partial shape matching across video frames. However, this algorithm is not applicable to grain image segmentation for two reasons: 1) The inter-slice resolution of a grain image is much lower than the

intra-slice resolution which causes the 2D shape of a grain to vary substantially when serial-sectioned by two consecutive slices, and 2) unlike video, which may contain structures with different shapes, many grains in superalloy images bear very similar shapes, and this may increase the ambiguity of the partial shape matching.

Based on above observations, a suitable inter-image propagated segmentation method for grain images should 1) consider the consistency of segment structures among adjacent images; and 2) smooth the segment boundaries.

2.3 Related Intra-Image Propagation Methods

Hierarchical Graph-Based Image Segmentation (GBH) [22] extends the graph-based image segmentation (reviewed in Section 2.1) for video segmentation. The input video is considered as a single 3D image, which is different from the StreamGBH method introduced previously. In order to capture object boundaries in different scales, the authors proposed a hierarchical structure: in the lowest layer, objects in the finest scale can be captured using the smallest τ value (defined in Eq. 2.1); while in higher layers, τ will be scaled up in order to capture objects in coarser scales. Segmentation results of one layer (represented by a region graph) are propagated and utilized as the initialization for obtaining the segmentation on its above layer. The authors also propose new definitions of edge weights between two regions based on the dense optical flow, to utilize the motion information contained in the video.

Ultrametric Contour Map (UCM) Based on the boundaries detected by gPb (reviewed in Section 2.1), [3] further proposed a multiscale hierarchical contour detection method. Similar to the hierarchy proposed in GBH, the base level of the UCM hierarchy represents weak contours, resulting in an oversegmentation, and upper levels respect only strong contours, resulting in an undersegmentation. Later a hierarchy level can be selected based on additional knowledge. The UCM hierarchy is

constructed also by a greedy graph-based region merging algorithm. Each iteration, the algorithm merges the most similar regions.

Chapter 3

Inter-Image Propagation

Given a sequence of images, the inter-image propagation methods propagate and utilize the existing segmentation of an image to obtain new segmentation on adjacent images. As discussed in Section 1.1, because its efficiency and accuracy, the inter-image propagation is especially important for 3D material image segmentation, where the 3D image is represented by a serial-sectioned high-resolution 2D image sequence. Here we propose a EWCVT (introduced in Section 2.1) based inter-image propagation method that preserves stable segment structures during the propagation. An overview of the proposed propagation based method is illustrated in Figure 3.1, where an existing segmentation is first propagated from previous image slices to the current one, and then this segmentation is refined according to current image information while preserving detected stable segment structures. For cases where the inter-image distance is quite large and thus propagated stable segment structures may not easily find its correspondence on the next slice, we model the problem in the multi-target tracking framework, and further propose a non-rigid transformation based association algorithm.

In this chapter, we first define the stable segment structure and describe its propagation in Section 3.1, then we propose the cluster initialization method preserving the propagated stable segment structure in Section 3.2, the proposed method of constructing EWCVT with consistency constraints is described in Section 3.3, the non-rigid transformation based association algorithm is described in Section 3.4. Finally we describe the experiment results and discussions in Section 3.5 and Section 3.6

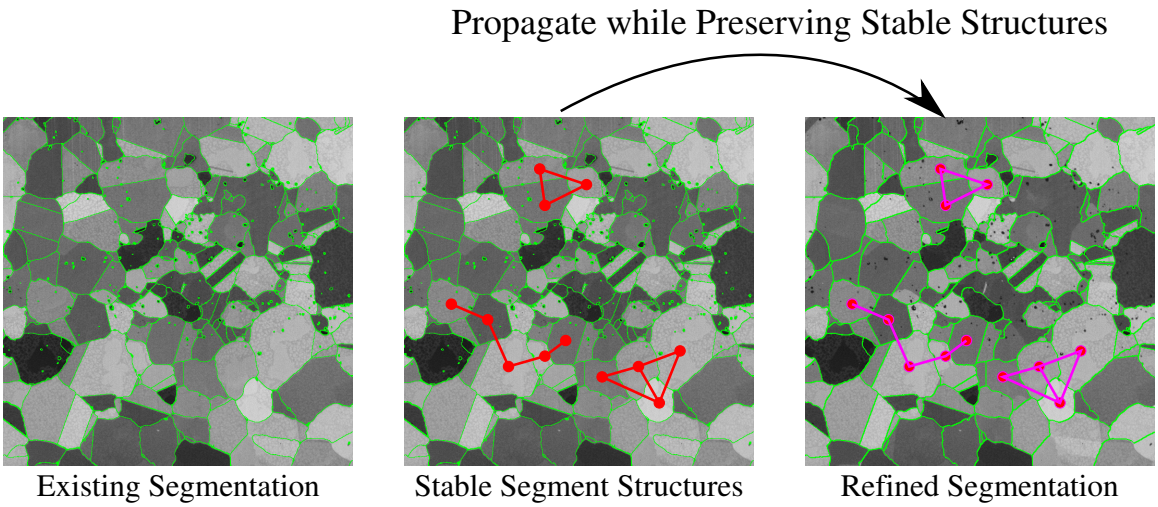


Figure 3.1: Overview of the proposed propagation method for constructing segmentations for an image sequence.

respectively.

3.1 Stable Segment Structure and Its Propagation

Given two consecutive image slices I^i and I^{i+1}, their segmentation results can be defined as $S^i = \{s_1^i, \ldots, s_{m_i}^i\}$ and $S^{i+1} = \{s_1^{i+1}, \ldots, s_{m_{i+1}}^{i+1}\}$ where m_i and m_{i+1} are the number of segments in I^i and I^{i+1} respectively. The segment structure of the segmentation S^i on the image slice I^i can be represented by a graph of segments in S^i, denoted as $G^i\left(\mathcal{V}^i, \mathcal{E}^i\right)$, where each vertex in $\mathcal{V}^i$ is a segment and the edge weights in $\mathcal{E}^i$ measure the strength of the adjacency of two neighbor segments (directly connected). Typically, given two segments, we use the number of pixels located on the boundary shared by them as their edge weight.

The stable segment structure of S^i on I^i can be defined as a connected sub-graph $G_*^i\left(\mathcal{V}_*^i, \mathcal{E}_*^i\right)$ of G^i. Specifically, it holds that

$$\mathcal{V}_*^i = \left\{ s_p^i \in \mathcal{V}^i \mid \left| s_p^i \right| \geq \alpha \right\} \tag{3.1}$$

and

$$\mathcal{E}_*^i = \left\{ \mathcal{E}_{(p,q)}^i \in \mathcal{E}^i \mid \mathcal{E}_{(p,q)}^i \geq \beta, \quad S_p^i, S_q^i \in \mathcal{V}_*^i \right\}, \tag{3.2}$$

where the parameter $\alpha > 0$ is the minimal size of segments that are defined as stable ones, and the parameter $\beta > 0$ is the minimal length of boundaries that are stable. In our 3D superalloy image segmentation problem, the stable segment structure of S^i on I^i should be preserved in S^{i+1} on I^{i+1}. Unstable segments and their adjacency, caused by the difference of two consecutive image slices, are determined by the image information on I^{i+1}.

We first use the segmentation S^i on the previous image slice I^i as the initialization, i.e., let $S^{i+1} = S^i$. Then by combining G^i_* with the image information I^{i+1}, we can construct the segment structure $\tilde{G}^{i+1}_*$ on S^{i+1} that is invariant to the change of image information from I^i to I^{i+1}. The vertexes in $\tilde{G}^{i+1}_*$ are the segments in S^{i+1} that are corresponding with stable segments in S^i, and the edges indicate corresponding stable segments' neighbor relationships. Thus $\tilde{G}^{i+1}_*$ can be viewed as a propagation of the stable segment structure G^i_* from image slice I^i to I^{i+1}.

Specifically, we define the corresponding distance between a segment s^i_p on I^i and a segment s^{i+1}_q on I^{i+1} as

$$d(s^i_p, s^{i+1}_q) = \frac{|u^{i+1}(s^i_p) - u^{i+1}(s^{i+1}_q)|}{|s^i_p \cap s^{i+1}_q|} \tag{3.3}$$

where $u^{i+1}(s^i_p)$ denotes the average intensity of pixels inside segment s^i_p on I^{i+1}:

$$u^{i+1}(s^i_p) = \frac{1}{|s^i_p|} \sum_{(x,y)\in s^i_p} u^{i+1}(x, y).$$

For each segment $s^i_p \in \mathcal{V}^i_*$, we find its nearest segment $s^{i+1}_q \in S^{i+1}$ and add s^{i+1}_q into $\tilde{\mathcal{V}}^{i+1}_*$, with respect to the distance defined in Eq. (3.3), i.e.,

$$s^{i+1}_q = \arg \min_{s^{i+1}_k \in S^{i+1}} d(s^i_p, s^{i+1}_k). \tag{3.4}$$

The intuition here is that two corresponding segments should be similar not only in the intensity space but also in the spatial domain. Otherwise the segment adjacency in G^i_* and $\tilde{G}^{i+1}_*$ are not consistent. The edges $\tilde{\mathcal{E}}^{i+1}_*$ can be simply derived from G^i_*.

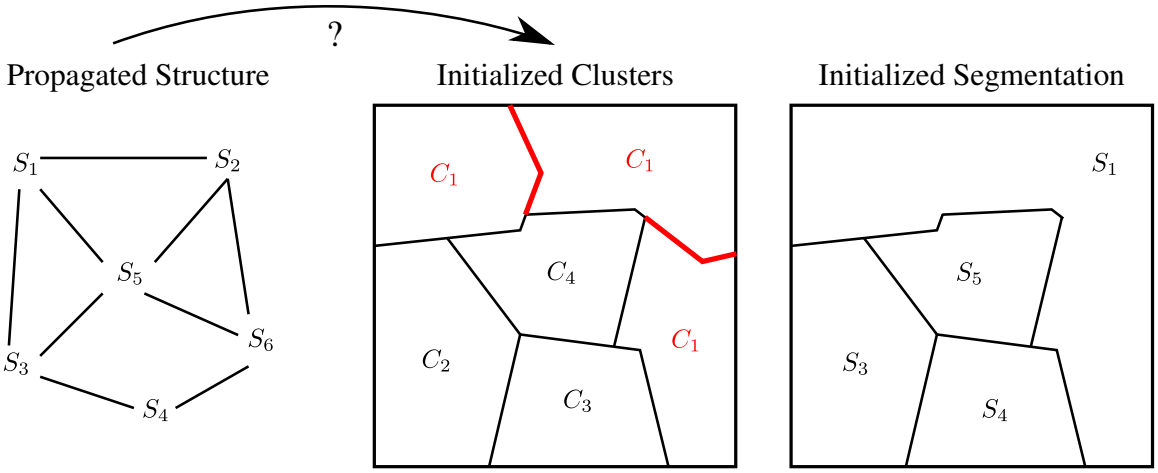


Figure 3.2: An illustration of an incorrect clustering initialization which may not be able to preserve the propagated segment structure. See text for detailed description.

3.2 Cluster Initialization Satisfying Consistency Constraint

Given the segmentation S^i and the propagated stable segment structure G^i_*, we now compute a good initial clustering $\mathcal{D}^{i+1} = \{D_l\}_{l=1}^L$ of the image slice I^{i+1} for the iterative construction of the EWCVT, and this initial configuration also must satisfy the segment structure $\tilde{G}^{i+1}_*$ inherited from G^i_*. An incorrect clustering initialization may not preserve the propagated segment structure. For example, as shown in Figure 3.2, if the cluster initialization process assigns S_1, S_2 and S_6 with the same cluster, then the propagated segment structure is violated since all three segments are merged into one and the segment structure is totally different from the propagated one.

Here, we proposed a CVT/k-means-type iterative process for initializing clusters on S^{i+1} while preserving propagated segment structures. Specifically, we treat each segment $s^{i+1}_q \in S^{i+1}$ as a point and define its value as $u^{i+1}(s^{i+1}_q)$ i.e., the average intensity of pixels inside segment s^{i+1}_q. Then we can define a new intensity domain on the segments as $\mathbb{U}_{S^{i+1}} = \left\{u^{i+1}(s^{i+1}_q) \mid s^{i+1}_q \in S^{i+1}\right\}$. Let $\hat{\mathcal{D}} = \{\hat{D}_l\}_{l=1}^L$ be a partition of S^{i+1} into L clusters, then the CVT tessellation of S^{i+1} can be constructed based

on the new average intensity according to the weighted CVT energy

$$\hat{E}(\mathcal{W}, \hat{\mathcal{D}}) = \sum_{l=1}^{L} \sum_{s_q^{i+1} \in \hat{D}_l} \left| s_q^{i+1} \right| \left| u^{i+1}(s_q^{i+1}) - w_l \right|^2. \tag{3.5}$$

and its corresponding CVT distance is

$$\widehat{dist}(s_q^{i+1}, w_l) = \left| u^{i+1}(s_q^{i+1}) - w_l \right|. \tag{3.6}$$

Notice that two initially neighbor segments may be merged into a single segment if they are assigned to the same cluster. In this way the propagated stable segment structure $\tilde{G}_*^{i+1}$ can not be preserved. In order to guarantee the resulting partition preserves the segment neighboring relationship defined in S^{i+1}, the clustering result should satisfy

$$\forall\ s_k^{i+1} \in \mathcal{N}_{s_q^{i+1}}, \quad \pi^{i+1}(s_q^{i+1}) \neq \pi^{i+1}(s_k^{i+1}), \tag{3.7}$$

where $\mathcal{N}_{s_q^{i+1}}$ denotes the neighbor segments of s_q^{i+1} and $\pi^{i+1}(s_q^{i+1})$ tells the index of the Voronoi region the segment s_q^{i+1} belongs to. During the classic CVT construction based on $\left\{ u^{i+1}(s_q^{i+1}) \mid s_q^{i+1} \in S^{i+1} \right\}$, any new cluster assignment that violates Eq. (3.7) should be prevented. The whole cluster initialization process is described in Algorithm 3.1, which will be used as part of input for further computing the EWCVT with consistency constraint.

3.3 Construction of EWCVT with Consistency Constraint

The initialization through Algorithm 3.1 guarantees the initialized partition $\mathcal{D}^{i+1} = \{D_l\}_{l=1}^{L}$ (obtained through π^{i+1}) preserves the propagated segment structure $\tilde{G}_*^{i+1}$. Now we present a modified EWCVT clustering algorithm such that the propagated segment structure is preserved during the clustering process. Meanwhile new segments can be identified using the image information. The basic idea is, during the EWCVT clustering process, 1) preventing any cluster assignment that breaks the propagated segment structure $\tilde{G}_*^{i+1}$; and 2) adjusting the number of segments in S^{i+1}.

Algorithm 3.1 (Cluster Initialization Satisfying Consistency Constraint)

Inputs: The image slice I^{i+1} and u^{i+1}
S^i: Segments of the image slice I^i
π^i: The cluster index function of the image slice I^i
L: Number of clusters
$niter$: Number of iterations

0 **Initialization**: Create the stable segment structure G^i_* of S^i. Set $S^{i+1} = S^i$ and create $\tilde{G}^{i+1}_*$ for S^{i+1}. Set $\pi^{i+1} = \pi^i$.
1 **FOR** $iter = 1, \ldots, niter$
2 **FOR** $l = 1, \ldots, L$
3 Compute the centroid
$$w_l = \frac{\sum_{s^{i+1}_q \in \hat{D}_l} \left|s^{i+1}_q\right| u^{i+1}(s^{i+1}_q)}{\sum_{s^{i+1}_q \in \hat{D}} \left|s^{i+1}_q\right|}$$
4 **FOR** each $s^{i+1}_q \in S^{i+1}$
5 Find the nearest $w_k \in \{w_l\}^L_{l=1}$ to s^{i+1}_q
w.r.t. the distance function $\widehat{dist}(s^{i+1}_q, w_k)$
6 **IF** the relation (3.7) and $\tilde{G}^{i+1}_*$ are satisfied
7 Set $\pi^{i+1}(s^{i+1}_q) = k$
8 **IF** there is no cluster change among S^{i+1}
9 Break

Output: The cluster index function π^{i+1} and $\{w_l\}^L_{l=1}$

Specifically, we only consider pixels located at the boundaries of stable segments in $\tilde{\mathcal{V}}^{i+1}_*$ and inside unstable segments $S^{i+1} - \tilde{\mathcal{V}}^{i+1}_*$, denoted as $(x, y) \in \Omega$. Those pixels can only be assigned to a cluster which is physically connected to them, i.e., $\pi^{i+1}(x, y) \in \left\{\pi^{i+1}(\mathcal{N}_{(x,y)})\right\}$ where $\mathcal{N}_{(x,y)} = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$. By checking the propagated stable segments adjacent relations defined in $\tilde{G}^{i+1}_*$, in each new cluster assignment, we only allow assignments that preserve the stable segments adjacency. After each assignment, if the clusters of the surrounding segments are different from the cluster of the center pixel, we identify this center pixel as a new segment and add it into S^{i+1}. The whole process can be described in Algorithm 3.2.

Notice that, we propagate the previous segmentation information by using S^i as the initialization directly. Therefore the segment indexes are consistent across image slices. Furthermore, new identified segments will be also included in the updated S^{i+1} as described in Algorithm 3.2. Finally, we can easily correspond segments across 2D

Algorithm 3.2 (EWCVT Construction Maintaining Consistency Constraint)

Inputs: The image slice I^{i+1} and u^{i+1}
S^i: Segments of the image slice I^i
π^i: The cluster index function of the image slice I^i
L: Number of clusters
$niter$: Number of iterations

0 **Initialization**: Run Algorithm 3.1.
1 **FOR** $iter = 1, \ldots, niter$
2 Construct the set of candidate pixels for transferring, Ω from S^{i+1}
3 **FOR** each $(x, y) \in \Omega$
4 Find the nearest $w_k \in \left\{ w_l \mid l \in \left\{ \pi^{i+1}\left(\mathcal{N}_{(x,y)}\right) \right\} \right\}$ to (x, y) w.r.t. the distance function $dist((x, y), w_k)$
5 **IF** no segments adjacency in $\tilde{G}_*^{i+1}$ is violated
6 Set $\tilde{k} = \pi^{i+1}(x, y)$, $\pi^{i+1}(x, y) = k$
7 Update w_k and $w_{\tilde{k}}$
8 **IF** $\pi^{i+1}(x, y) \notin \pi^{i+1}\left(\mathcal{N}_{(x,y)}\right)$
9 Add a new segment that contains (x, y) in S^{i+1}
10 **IF** there is no cluster change among Ω
11 Break

Outputs: The final cluster index function π^{i+1} and segmentation S^{i+1} of the image slice I^{i+1}

image slices and then construct 3D segments.

3.4 Segment Consistency under Long Distance Propagation

In Section 3.1, for two consecutive image slices I^{i-1} and I^i, we propagate the stable segment structure G_*^{i-1} derived from segmentation S^{i-1} on I^{i-1}, and utilize it to guide the segmentation on the next slice I^i. Later after constructing segmentation on all slices, our goal is to associate all segments together in order to reconstruct the underlying 3D object structures. The segment association problem can be illustrated as in Figure 3.3.

However, when the inter-slice distance is large, the neighboring relations among

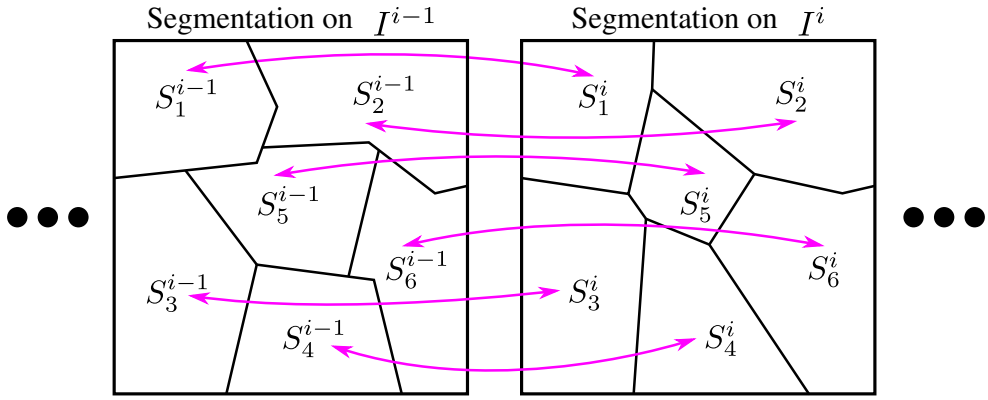


Figure 3.3: An Illustration of segment association problem for a set of segments constructed on an image sequence. Our goal is to find the correspondence among segments constructed on neighboring image slices, and eventually reconstruct the underlying 3D object structure by associating all segments together.

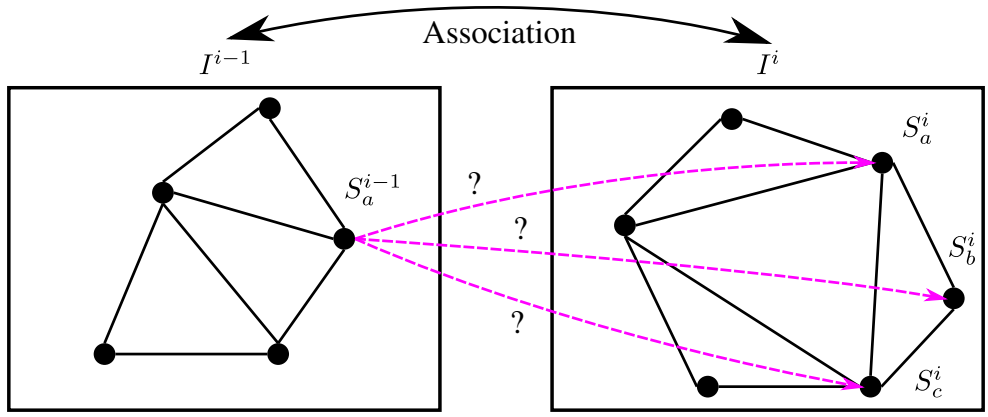


Figure 3.4: An illustration of correspondence problem for long distance propagation. Given a large inter-slice distance, it's hard to find the correspondence between propagated stable segment structure from previous image slice I^{i-1} and the segment structure on image slice I^i.

objects of interests may change a lot. And it becomes hard to find the correspondence between propagated stable segment structure G_*^{i-1} on image slice I^{i-1} and segment structure G^i on image slice I^i. For example, as illustrated in Figure 3.4, because of the change of neighboring relations among segments, it is difficult for a segment S_a^{i-1} in image slice I^{i-1} to find the correct corresponding segment in image slice I^i. Thus, the above mentioned segment consistency constraint proposed for Algorithm 3.2 may lead to incorrect segmentation on image slice I^i.

Naturally, we can model the finding of pairwise correspondences between segment

structures constructed on two neighboring slices in an image sequence as a multiple target tracking problem, where targets are segments or vertices of the segment structure defined in Eq. (3.1). In order to handle the cases where existing segments disappear due to object movement after long distance propagation or undersegmentation due to image noise, in this work we utilize a Kalman filtering based multi-target tracking framework that models the segment movements and predicts the locations of missing segments based on the motion model. We further improve the accuracy of segment motion model by proposing a non-rigid transformation based association method.

Kalman filter [47] is a type of recursive Bayesian filter and it is specialized for modeling linear target movement with additive Gaussian noise. The 2D target movement at image slice i is modeled as

$$\begin{aligned} \mathbf{x}^i &= A\mathbf{x}^{i-1} + \mathbf{w}^{i-1} \\ \mathbf{z}^i &= H\mathbf{x}^i + \mathbf{r}^i \end{aligned} \tag{3.8}$$

where the target state $\mathbf{x} = [c_x, c_y, v_x, v_y]^\top$, (c_x, c_y) is center location of a target, (v_x, v_y) is target velocities in horizontal and vertical directions respectively, $\mathbf{z}$ is observed target center location, $A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is the state transition matrix, $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ is the observation model, $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, Q)$ is the transition noise, and $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, R)$ is the observation noise. Then, the probabilistic state model can be defined as:

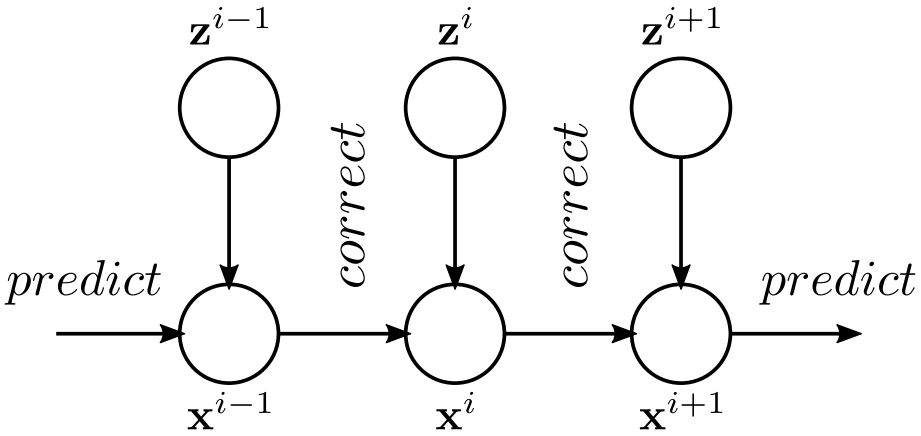


Figure 3.5: An illustration of the Kalman filter for recursively modeling target movement using observed target locations and predicting missing target locations on an image sequence.

$$\begin{aligned} \mathbf{x} &\sim p\left(\mathbf{x}^i \mid \mathbf{x}^{i-1}\right) = \mathcal{N}\left(A\mathbf{x}^{i-1}, Q^{i-1}\right) \\ \mathbf{z} &\sim p\left(\mathbf{z}^i \mid \mathbf{x}^i\right) = \mathcal{N}\left(H\mathbf{x}^i, R^i\right). \end{aligned} \tag{3.9}$$

Kalman filter for multi-target tracking contains three major steps: prediction, association and correction. As illustrated in 3.5, at image slice i, the Kalman filter first predicts the target state based on the movement model built upon previous slices, then after associated observed targets with filters, the movement models of filters can be corrected. These three steps are alternatively performed until reaching the end of an image sequence.

In the prediction step, for a target q, the filter predicts its state $\mathbf{x}_q^i$ on image slice i based on the model trained from data in previous image slices:

$$\begin{aligned} p\left(\mathbf{x}_q^i \mid Z_q^{i-1}\right) &= \int_{\infty} p\left(\mathbf{x}_q^i \mid \mathbf{x}_q^{i-1}\right) p\left(\mathbf{x}_q^{i-1} \mid Z_q^{i-1}\right) d\mathbf{x}_q^{i-1} \\ &= \mathcal{N}\left(\mu_q^{i|i-1}, P_q^{i|i-1}\right) \end{aligned} \tag{3.10}$$

where $Z_q^{i-1} = \left\{\mathbf{z}_q^1, \mathbf{z}_q^2, \ldots, \mathbf{z}_q^{i-1}\right\}$ are the observed target locations from image slice 1 to $i-1$, and $p\left(\mathbf{x}_q^{i-1} \mid Z_q^{i-1}\right) = \mathcal{N}\left(\mu_q^{i-1}, P_q^{i-1}\right)$. Since the Kalman filter models

noises, including both transition noise and observation noise, as additive Gaussian distributions, $\mu_q^{i|i-1}$ is the state mean predicted on image slice i using the model constructed on previous slice $i-1$, and the covariance matrix $P_q^{i|i-1}$ measures the uncertainty of the state prediction.

In the association step, given M predicted target states $X^i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_M^i\}$, and N observed targets $Z^i = \{\mathbf{z}_1^i, \mathbf{z}_2^i, \ldots, \mathbf{z}_N^i\}$, the goal is to find the correspondence between these predictions and observations. In this work, observations are the segments constructed on image slice I^i. Notice that in most cases $M \neq N$ since there are usually false positive and/or false negative observations. Also, for long distance propagation, because of large displacements due to targets movement, segment structures may change greatly, as illustrated in Figure 3.4. Thus, conventional one-to-one matching algorithms, such as the Hungarian algorithm [26], may produce incorrect associations since it enforces that every prediction must be associated with one observation. Erroneous associations result in inappropriate correction on the state models, and eventually it will influence the effectiveness of the proposed segmentation method.

In the correction step, after associating the predicted state models X^i with the observed target locations Z^i, for a target q, the state model can be updated through:

$$\begin{aligned} p\left(\mathbf{x}_q^i \mid \mathbf{z}_q^i, Z_q^{i-1}\right) &= \frac{p\left(\mathbf{z}_q^i \mid \mathbf{x}_q^i\right) p\left(\mathbf{x}_q^i \mid Z_q^{i-1}\right)}{p\left(\mathbf{z}_q^i \mid Z_q^{i-1}\right)} \\ &= \mathcal{N}\left(\mu_q^{i|i}, P_q^{i|i}\right) \end{aligned} \tag{3.11}$$

In this work, we propose a non-rigid transformation based association algorithm. The basic idea is to find non-rigid transformations that warp the predicted targets towards observed target locations, which constructs the association between the predictions and the observations. In order to achieve that, we utilize the Thin-Plate Spline Robust Point Registration (TPS-RPM) algorithm proposed in [9], where the problem is formulated as minimizing the following energy function:

$$E_{TPS-RPM}\left(\mathcal{H}, f\right) = \sum_{m=1}^{M}\sum_{n=1}^{N} h_{mn} \parallel z_n^i - f\left(x_m^i\right) \parallel^2 + \gamma \parallel f \parallel^2 + T\sum_{m=1}^{M}\sum_{n=1}^{N} h_{mn}\log h_{mn} - \varsigma\sum_{m=1}^{M}\sum_{n=1}^{N} h_{mn} \quad (3.12)$$

where $\mathcal{H}$ is the association matrix between M predictions and N observations, and the non-rigid transformation function f is obtained by minimizing the thin-plate bending energy:

$$E_{TPS}\left(f\right) = \sum_{m=1}^{M} \parallel z_m^i - f\left(x_m^i\right) \parallel^2 + \eta \iint \left[\left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2\right] dxdy. \quad (3.13)$$

The minimization of $E_{TPS-RPM}$ is performed via an annealing process as proposed in [9].

However, in practice, it is hard to find a single non-rigid transformation function f for a large number of targets with different velocities in a long distance propagation. A single function may not be sufficient to characterize complex non-rigid transformations. We address this issue in a divide-and-conquer manner.

On image slice I^i, given M predicted segment states $X^i = \left\{\mathbf{x}_q^i\right\}_{q=1}^{M}$, we first group them into K groups. Specifically, each predicted state is considered as a four dimension vector, i.e., $\mathbf{x} = [c_x, c_y, v_x, v_y]^\top$ where (c_x, c_y) is center location of a target and (v_x, v_y) is target velocities in horizontal and vertical directions respectively. Then we perform the k-means algorithm on the set of predicated states. In the experiments, we empirically set $K = 10$. Obtained prediction groups are visualized in different colors on the left panel of Figure 3.6.

For each prediction group $\mathcal{X}$, we look for the set of associated observations that is able to achieve the smallest Thin-Plate Spline bending energy. Specifically, as illustrated in Figure 3.7, we perform a sliding window search on observations of image

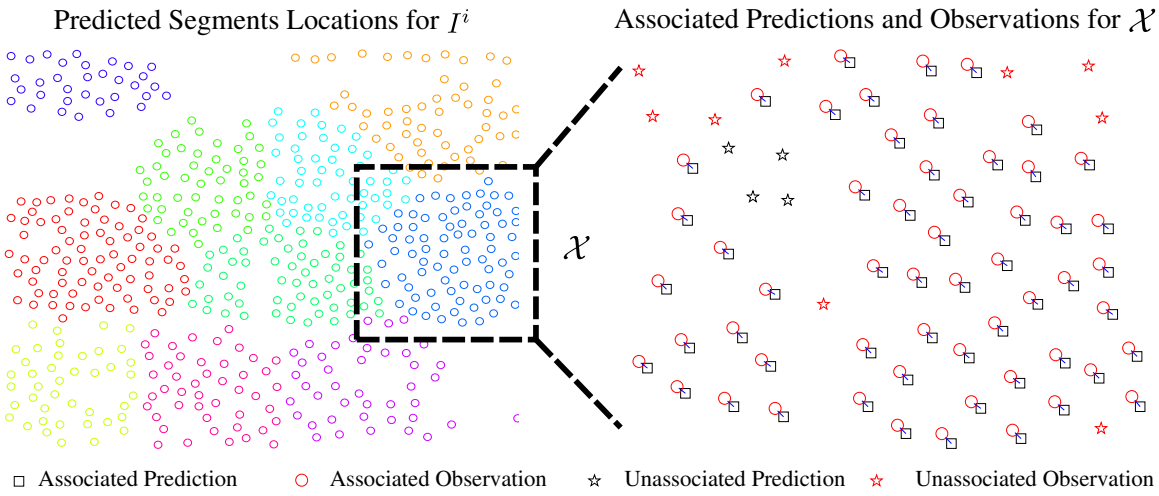


Figure 3.6: Divide-and-conquer approach to find associations between predicted targets and observed ones. On the left, we first divide predictions into compact groups, and then, on the right, for each group, the association is constructed through the non-rigid transformation based algorithm. Later, group-wise associations are combined together. See text for detailed description.

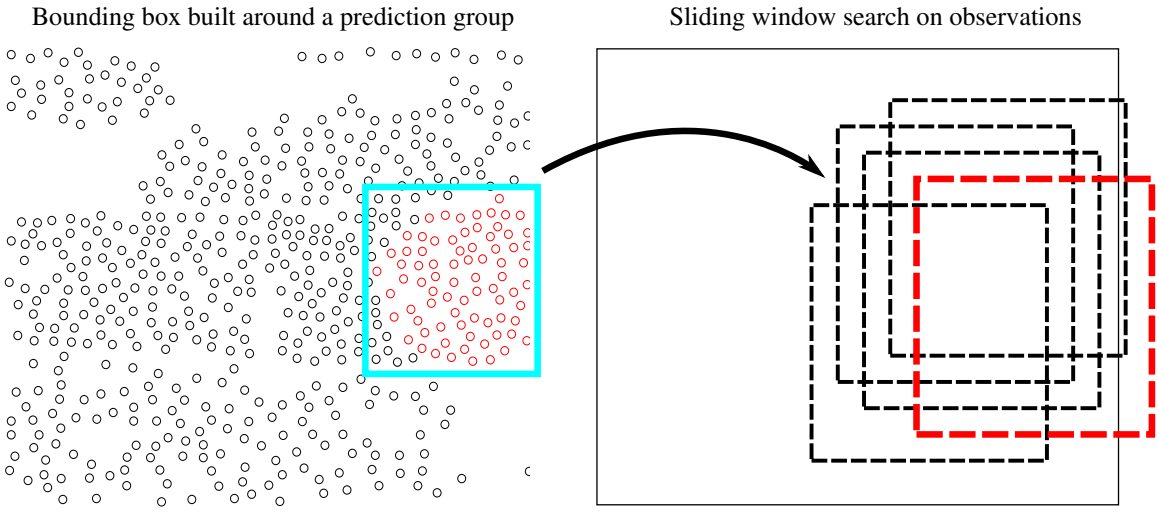


Figure 3.7: Sliding window search for constructing the observation association. Red window on the right indicates the set of associated observations that is able to achieve the smallest TPS energy.

Algorithm 3.3 (Group-wise Non-rigid Transformation based Association Algorithm)

Inputs: X^i: M predicted segment states on image slice I^i
Z^i: locations of N observed segments
K: Number of prediction groups

Cluster X^i into K groups using the k-means algorithm
FOR each prediction group $\mathcal{X}$
  Build an enclosing box $B_{\mathcal{X}}$
  Enlarge $B_{\mathcal{X}}$ into $\tilde{B_{\mathcal{X}}}$ on all four directions
  Obtain a set of sliding boxes by moving $\tilde{B_{\mathcal{X}}}$ around its center
  FOR each sliding box $\tilde{B_{\mathcal{X}}^+}$
    Find association H between $\mathcal{X}$ and
    observations located inside $\tilde{B_{\mathcal{X}}^+}$ by minimizing Eq. (3.12)
    Evaluate the TPS bending energy for H using Eq. (3.14)
  Keep the association H^*
  that achieves the smallest TPS bending energy
Merge associations obtained from all prediction groups together

Outputs: The association matrix between X^i and Z^i on image slice I^{i+1}

slice I^i with the objective looking for the association between predictions in $\mathcal{X}$ and observations in the window that achieves the smallest TPS bending energy.

Given a set of associated predictions and observations $H = \left\{(\mathbf{x}^i, \mathbf{z}^i)_q\right\}_{q=1}^{M}$, the TPS energy is given by

$$E_{TPS}(H) = \frac{1}{8\pi}\left(\mathbf{z}_x^\top \mathbf{L}\mathbf{z}_x + \mathbf{z}_y^\top \mathbf{L}\mathbf{z}_y\right) \tag{3.14}$$

where $(\mathbf{z}_x, \mathbf{z}_y)$ is the center location of observed target $\mathbf{z}$, $\mathbf{L}$ is the $M \times M$ upper left sub-matrix of

$$\begin{pmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{0} \end{pmatrix}^{-1}$$

given $\mathbf{K}$ is the pairwise distance matrix for all predictions in group $\mathcal{X}$, and $\mathbf{P} = (1, \mathbf{x}_x, \mathbf{x}_y)$ where $(\mathbf{x}_x, \mathbf{x}_y)$ is the center locations (in row vector) of predicted target $\mathbf{x}$.

Starting with an enclosing rectangle box $B_{\mathcal{X}} = [x_{tl}, y_{tl}, x_{br}, y_{br}]$ for $\mathcal{X}$ where (x_{tl}, y_{tl}) and (x_{br}, y_{br}) are the top-left and the bottom-right corner points of box $B_{\mathcal{X}}$ respectively, we first enlarge $B_{\mathcal{X}}$ by $\Delta t = 5$px in all four directions, i.e., $\tilde{B_{\mathcal{X}}} =$

$[x_{tl} - \Delta t, y_{tl} - \Delta t, x_{br} + \Delta t, y_{br} + \Delta t]$. Then we move the box around its center, with range $\Delta x \in [-10, 10]$, $\Delta y \in [-10, 10]$ and the step length 10. For each obtained box, we find the association between predictions in $\mathcal{X}$ and observations inside the box by minimizing Eq. (3.12), and evaluate the TPS bending energy using Eq. (3.14). Among all sliding boxes, we keep the one that is able to achieve the smallest TPS bending energy by associating predictions with inside observations. Then we consider obtained association as the prediction-observation correspondence for group $\mathcal{X}$.

Finally, by combining association results of all prediction groups together, we are able to obtain the correspondence between predicted targets and observed ones, i.e., associating propagated stable segment structure G_*^{i-1} with constructed segment structure G^i. The whole algorithm is summarized in Algorithm 3.3.

By stacking a set of associated 2D segmentation slices, we are able to reconstruct the 3D structures, as illustrated in Figure 3.8.

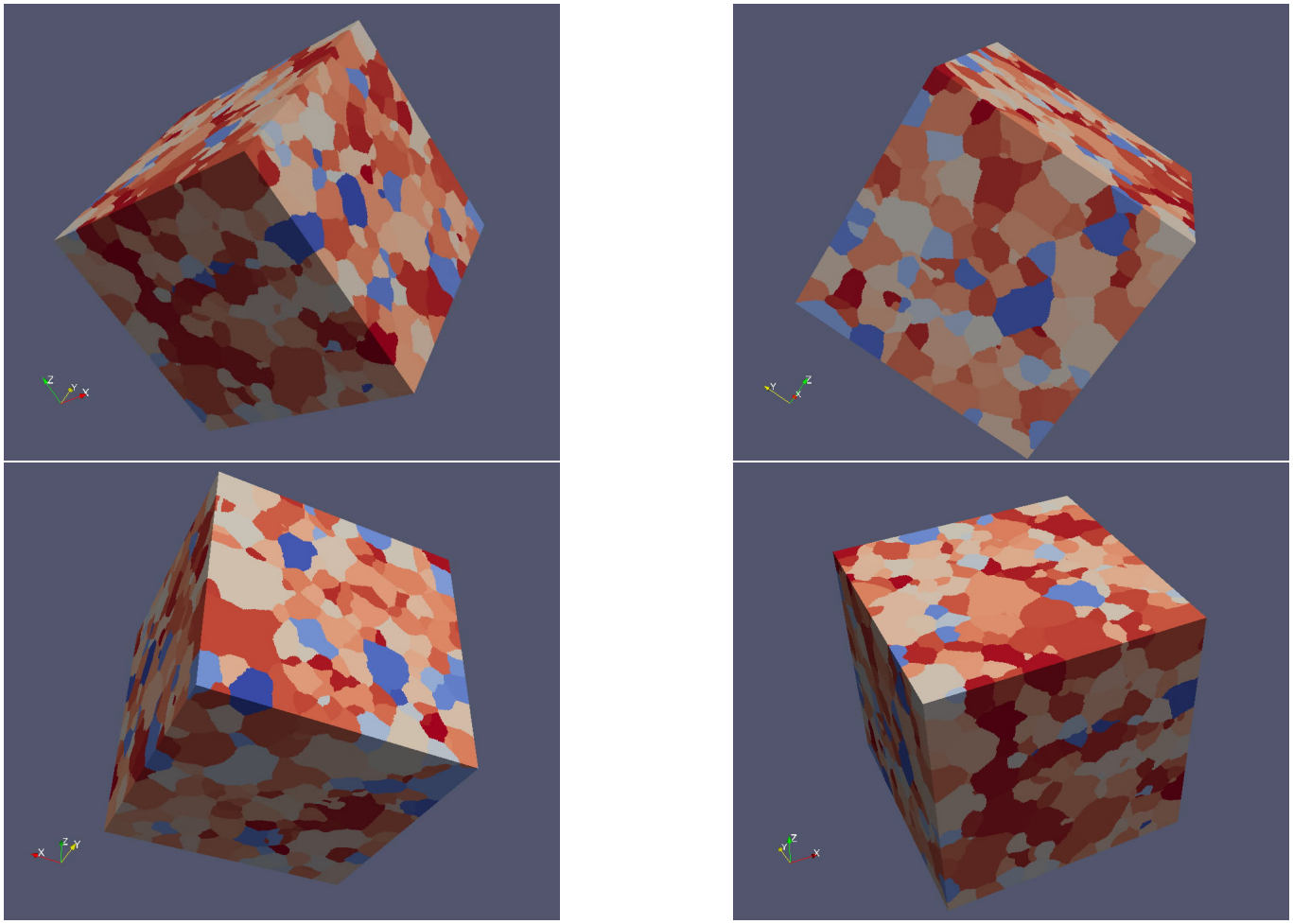


Figure 3.8: Illustrations of the reconstructed 3D structures. The structures are obtained by finding the correspondence among a set of 2D segmentations using proposed non-rigid transformation based association algorithm.

In this section, we first describe the test dataset, parameter settings, and evaluation criterion used in the experiments. Then we compare the proposed method with several 2D/3D segmentation algorithms in terms of accuracy and running time. Finally, we discuss how the accumulated propagation error affects the segmentation accuracy of the proposed method when segmenting a long sequence of 2D image slices.

IN100 Dataset

The experiments are conducted on the IN100 dataset[1] which contains 170 sequential 2D image slices of a superalloy material sample. These image slices are obtained by photographing (using microscope) the top surface of a superalloy sample block during a top-to-bottom abrading process.

Each slice in the IN100 dataset contains 4 gray-scale images taken under different microscope configurations. For the proposed method, we combine them into a single 4-channel image, analogous to typical RGB 3-channel images and use L^2-norm as the distance metric. We do not suppress any channel, which is different from the L^∞-norm used in MCEWCVT [7]. For comparison algorithms that cannot handle multi-channel images directly, we first apply such algorithms to each of the four gray-scale images independently. We then combine these independent segmentations into an additional fifth segmentation, either using the logic OR operation (for solid boundary segmentation algorithms, e.g. the NormalizedCuts [39] algorithm), or assigning the maximum probability boundary (pb) value to each pixel (e.g. for the gPb [3] algorithm). For each such comparison algorithm, we report the result (out of the above five results) which yields the best performance.

[1]Provided by our material scientist collaborates and can be downloaded at http://www.bluequartz.net/Data/.

Evaluation Metric

We use the boundary-overlap criterion suggested in the Berkeley segmentation benchmark [30] to evaluate the results quantitatively. Specifically, detected segmentation boundaries are compared with the ground-truth boundaries to calculate precision, recall and the F-score

$$\text{F-score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

As mentioned before, in the proposed method we use the segmentation obtained by the EWCVT algorithm on the first slice as an initialization, propagating it to segment the remaining 169 slices sequentially. We evaluate the segmentation accuracy on all 170 slices.

Parameter Settings

For the proposed method, there are two key parameters that can be tuned: the radius ω of the local smoothness region and the edge weight λ. We performed a grid search over this parameter space using the whole dataset and selected $\omega = 4$ and $\lambda = 30$ which achieved the best performance. Additionally, we set the number of clusters in the color space as $k = 40$. For the remaining parameters, the minimum size of stable segments α and the minimum length of stable boundaries β, are set according to the average grain size: $\alpha = 80$ and $\beta = 5$.

Comparison with 2D Segmentation Methods

We compare the proposed method with six automatic 2D segmentation algorithms, including MeanShift [10], the graph-based (GraphBased) algorithm of [18], SRM [33], gPb [3], and NormalizedCuts [39]. Additionally, we also compare the proposed method with the original EWCVT [44] algorithm on 2D image slices. The Normal-

Methods		Precision	Recall	F-score
2D Methods	EWCVT [44]	0.838385	0.962131	0.896005
	MeanShift [10]	0.911927	0.844106	0.876707
	GraphBased [18]	0.704163	0.928424	0.800891
	SRM [33]	0.81018	0.800006	0.805061
	gPb [3]	0.828988	0.866076	0.847126
	NormalizedCuts [39]	0.736609	0.691646	0.71342
3D/Propagated Methods	MCEWCVT [7]	0.845894	0.927918	0.885009
	3D Levelset [48]	0.739025	0.581001	0.650554
	3D Watershed [31]	0.864594	0.589135	0.700767
	StreamGBH [51]	0.454185	0.792653	0.577479
Proposed		**0.957377**	**0.896125**	**0.925739**

Table 3.1: Quantitative comparison of 2D/3D/Propagated segmentation methods on the IN100 dataset.

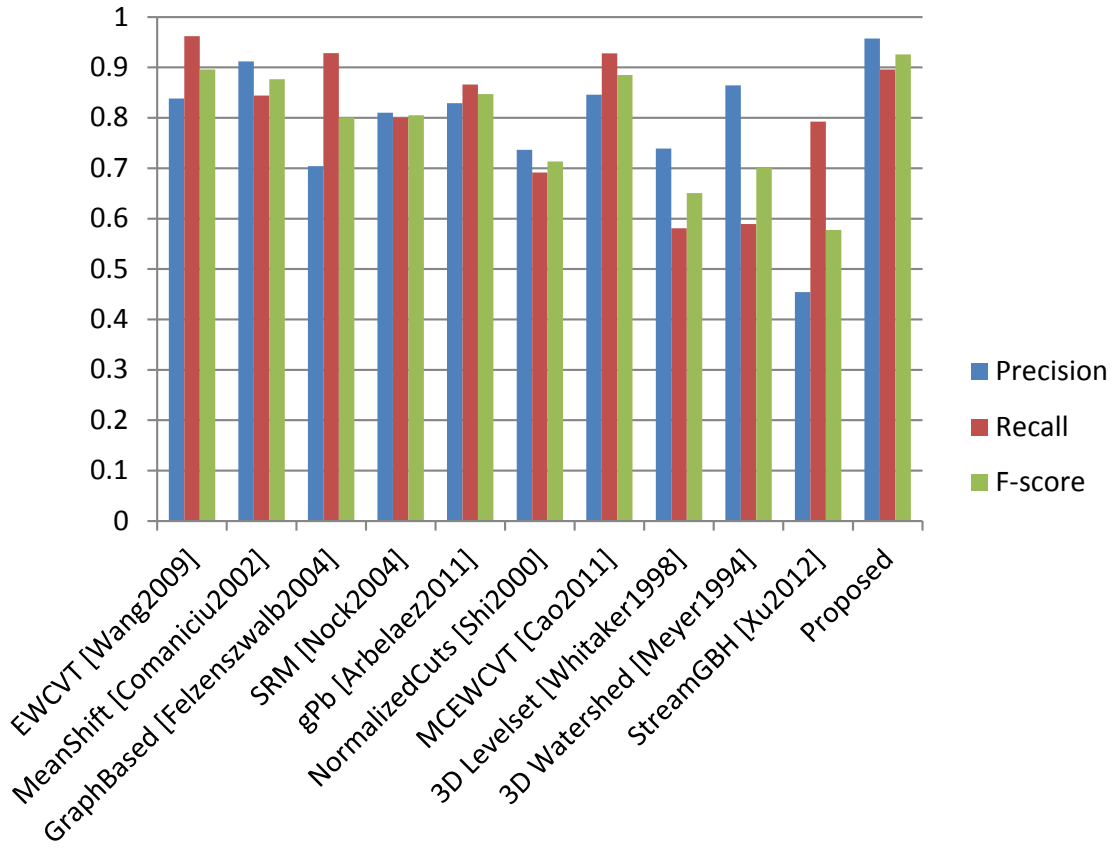


Figure 3.9: An illustration of quantitative comparison with 2D/3D/Propagated segmentation methods on the IN100 dataset.

Methods	MCEWCVT [7]	StreamGBH [51]	Proposed
Time (s)	37243.2	8808.7	**7651.7**

Table 3.2: Comparison on running times of 2D/3D/Propagated segmentation methods on the IN100 dataset.

izedCuts algorithm requires the number of desired segments, which we set to the number of ground truth segments in this dataset. Parameters of other algorithms are set either to their default values, or the setting that provides the best performance from a coarse grid search.

The quantitative results are illustrated in Figure 3.9, and specific numbers are shown in the middle six rows of Table 3.1, from which we can see that the original 2D EWCVT already achieves a very good performance with an F-score of 89.6%, while the proposed EWCVT-based propagation method further significantly improves the segmentation accuracy by another 3% to 92.5%. This indicates that, aside from the excellent performance of the EWCVT clustering algorithm, the consistency constraints in the proposed method indeed boost the performance further.

Comparison with 3D/Propagated Segmentation Methods

We compare the proposed method with the MCEWCVT algorithm in [7], the 3D levelset algorithm [48], the 3D watershed algorithm [31] and the StreamGBH algorithm in [51]. We select StreamGBH because it is a propagation-based algorithm and it achieves state-of-the-art performance on video segmentation tasks [51], which is similar to the 3D grain image segmentation application. Another related work is the algorithm in [13], however the authors have not released the implementation of this algorithm.

For the MCEWCVT algorithm, we use the parameter configuration provided in the original paper [7]. For the 3D levelset algorithm, the number of seeds are the same as the number of grains contained in the ground truth segmentation, and the

seeds are evenly distributed in the 3D space. For StreamGBH, we set the number of consecutive image slices involved in the propagation (i.e., the parameter "range") to be 2, which is equivalent to only using the previous image slice to do propagation, as in the proposed method. After performing a coarse grid search, other parameters of StreamGBH are set to be: $n_{hie} = 10$, $c = 60$, $c_{reg} = 200$, $min = 100$ and $\sigma = 0.8$.

The quantitative results are illustrated in Figure 3.9, and specific numbers are shown in the bottom two rows of Table 3.1, the proposed method clearly outperforms the comparison algorithms. MCEWCVT under-performs because it groups strong noise in the dense 3D image space into separate clusters. StreamGBH shows lower performance because it lacks structure consistency constraints and boundary smoothness in the propagation, leading to isolated and jagged boundaries.

Moreover, in order to demonstrate the computational efficiency of the proposed method, we also compare its running time with that of the MCEWCVT and the StreamGBH algorithms. All these algorithms are implemented in C/C++ and their running times are shown in the last column of Table 3.2. The proposed method exhibits the fastest running time, with a speedup of 5× compared with the MCEWCVT algorithm.

Qualitative Comparisons

Qualitative segmentation results on three consecutive image slices of both 2D image segmentation methods and 3D/Streaming segmentation methods are shown in Figures 3.10, 3.11 and Figures 3.12, 3.13 respectively. We can clearly see that the segmentation from the proposed method aligns with grain boundaries much better than the segmentation from the comparison algorithms. For 2D comparison algorithms, without considering inter-slice correspondence, they often produce isolated fragments inside a grain. In contrast, the proposed method maintains the consistency and correspondence among grains across image slices. For the 3D algorithms,

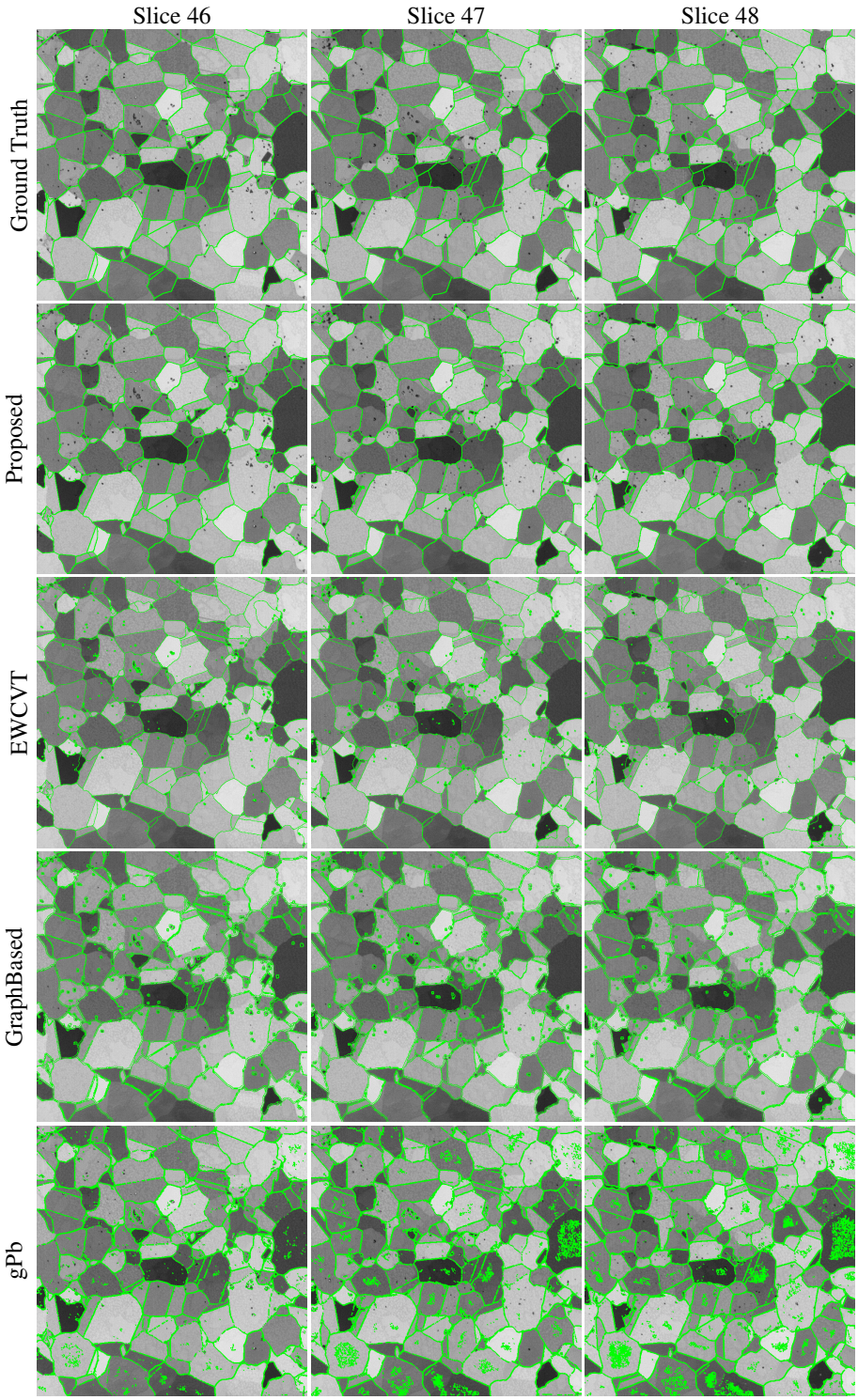


Figure 3.10: Qualitative comparisons of the four 2D/Propagated image segmentation methods (Proposed, EWCVT, GraphBased and gPb) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods.

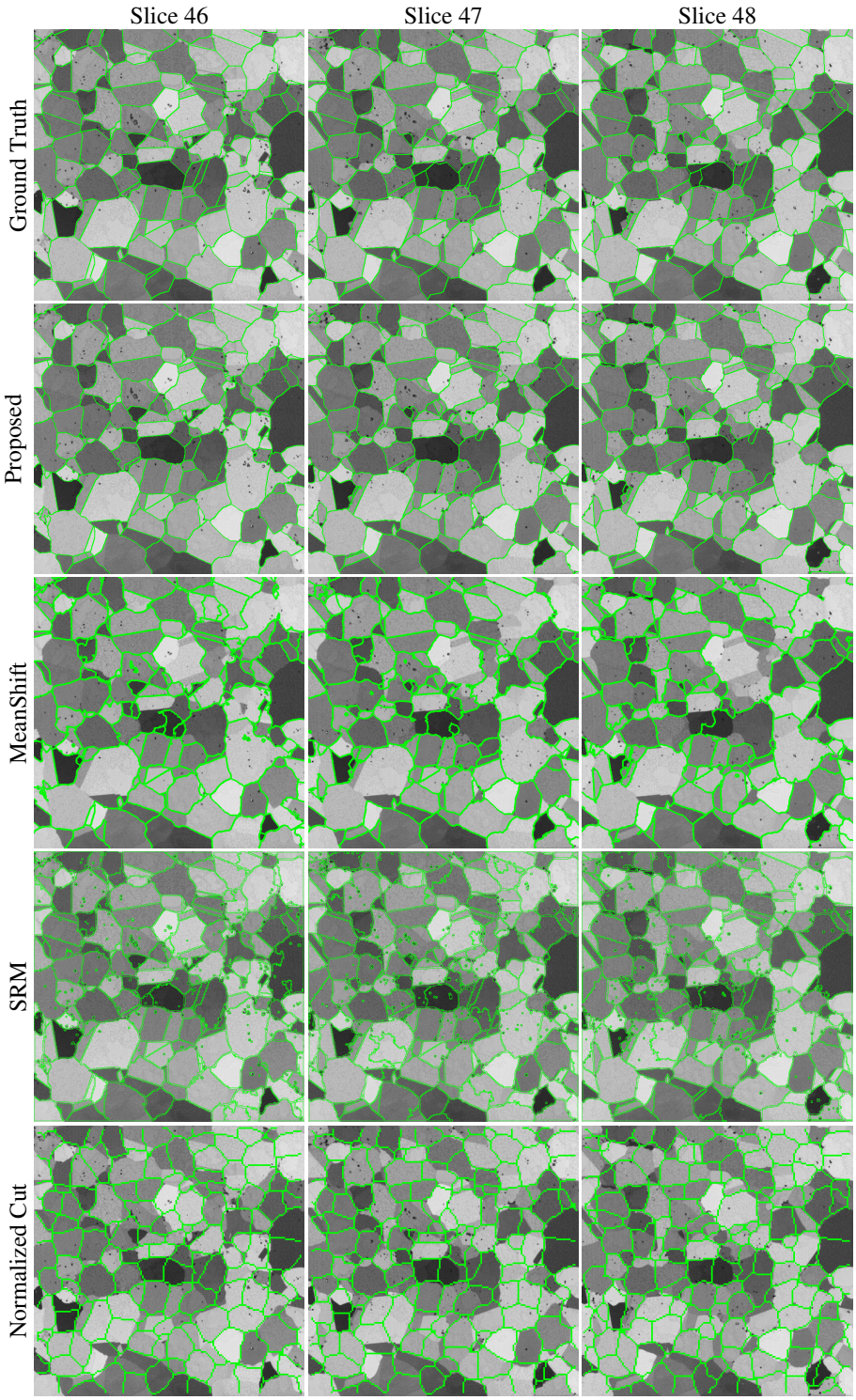


Figure 3.11: Qualitative comparisons of the four 2D/Propagated image segmentation methods (Proposed, MeanShift, SRM and NormalizedCuts) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods.

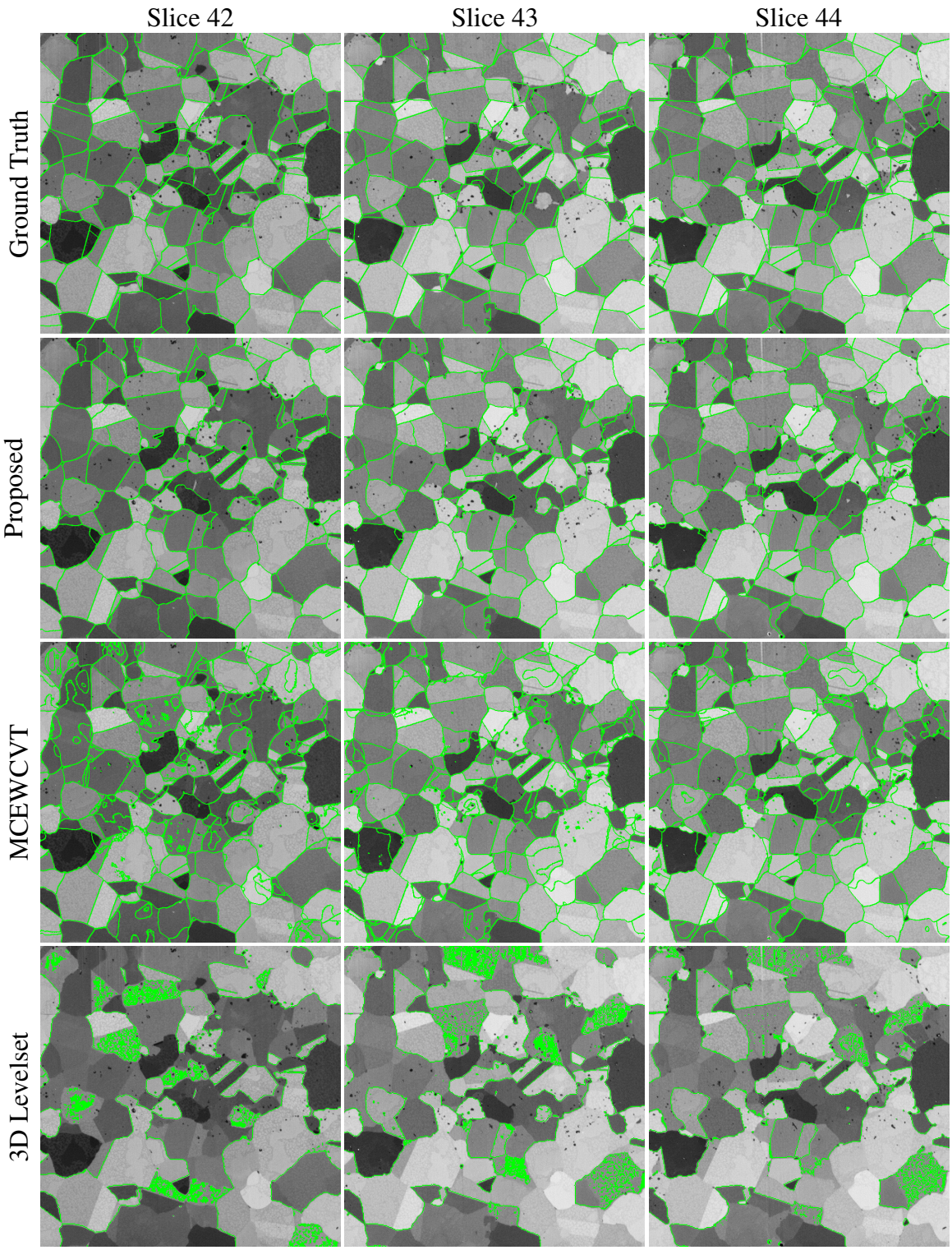


Figure 3.12: Qualitative comparisons of the three 3D/Propagated image segmentation methods (Proposed, MCEWCVT and 3D Levelset) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods.

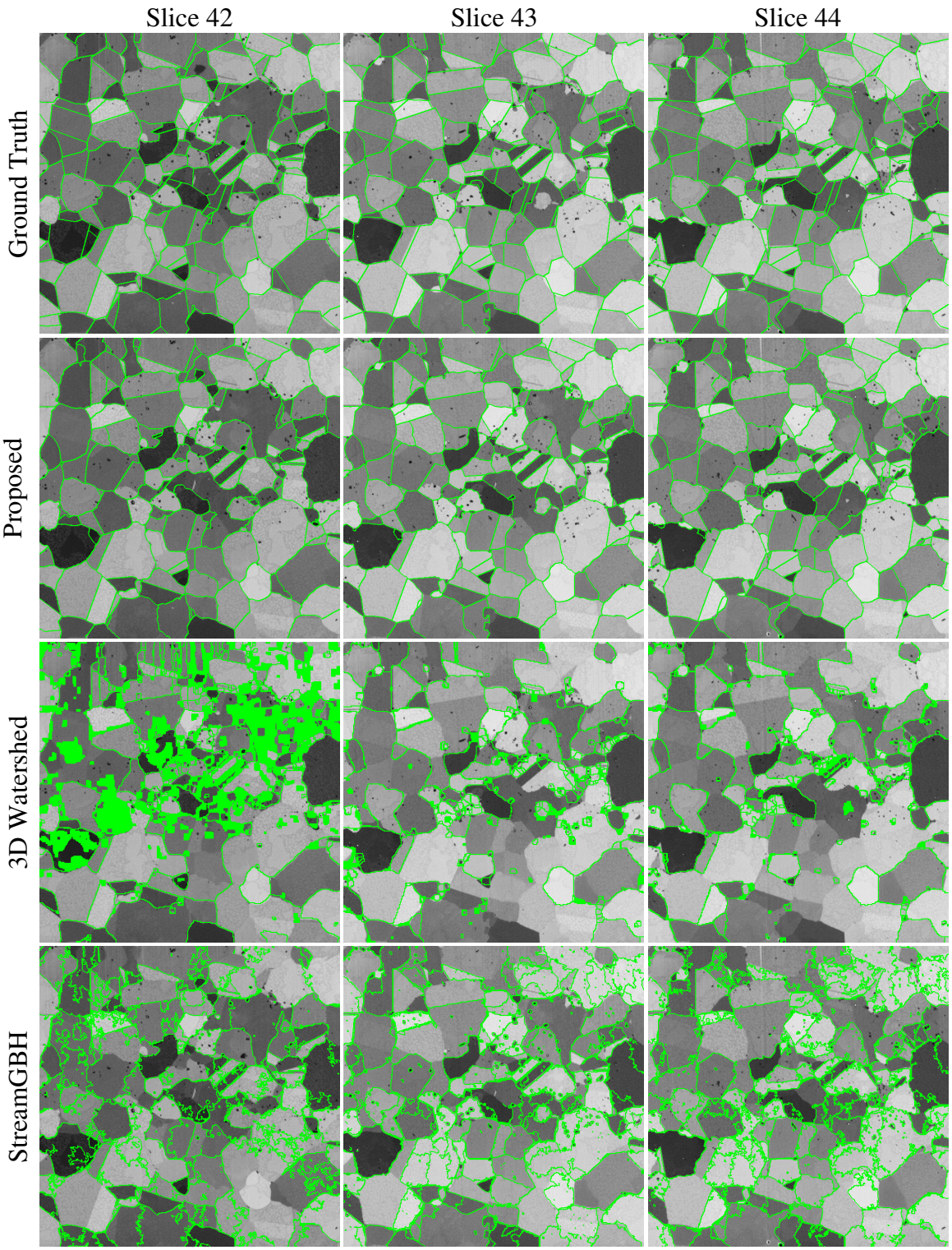


Figure 3.13: Qualitative comparisons of the three 3D/Propagated image segmentation methods (Proposed, 3D Watershed and StreamGBH) on three consecutive image slices from the IN100 dataset. From the top to the bottom are the original image with ground truth boundaries and results of different methods.

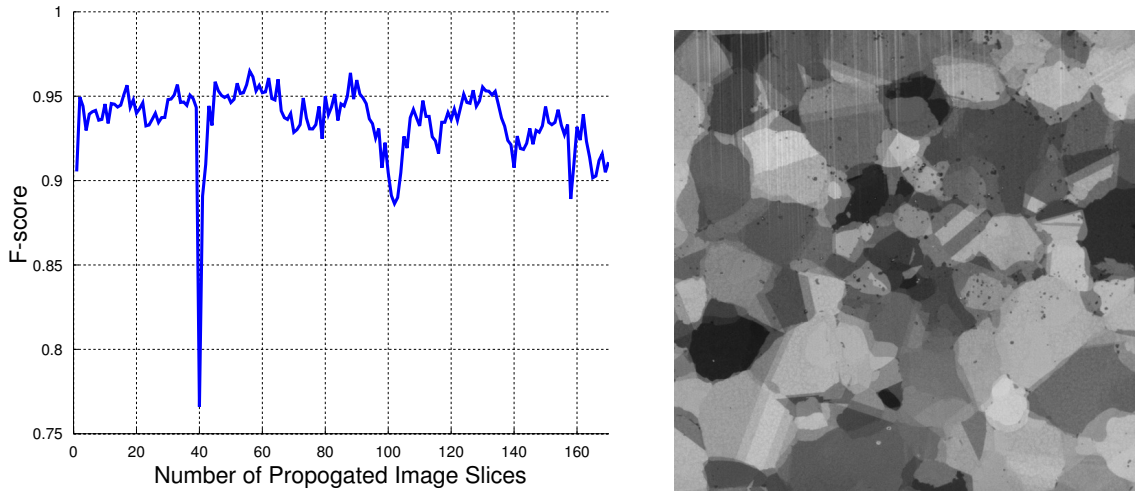


Figure 3.14: The F-score on each slice (left) during propagation. The sharp drop is caused by a corrupted slice (right) in the original IN100 dataset.

compared with the proposed method, they are sensitive to strong noises in the dense 3D space. For StreamGBH, without considering the structure consistency and the boundary smoothness, its results contain many fragments along grain boundaries.

Propagation Error Analysis

Intuitively, the segmentation error accumulates when the segmentation propagates through a large number of slices. Therefore, we may expect a monotonic decrease of the segmentation accuracy with more steps of propagation. However, as shown in the top panel of Figure 3.14, the segmentation accuracy only oscillates occasionally during the propagation. The main reason is that, aside from the propagation of structural consistency constraints, the proposed method also includes a EWCVT clustering process to refine the segmentation using the image information when processing a new slice. Also note hat the F-score on the first image is relatively low. This is due to the EWCVT result being used as the initialization. Even from this imperfect initialization, the proposed method is able to improve the performance by considering both structure and image information in propagation. Similarly, one slice of corrupted

or highly noisy image, as shown in the bottom panel of Figure 3.14, has little effect on the segmentation performance on the other slices because of the use of structure constraints in the proposed method.

Experiments on Synthesized Dataset

In order to verify the robustness of the proposed method, in this section, we compare its performance with it of several other comparison algorithms, on two synthesized datasets. In the following, we first propose a texture based algorithm to generate synthesized superalloy data, and then report quantitative evaluation results on two constructed datasets.

Synthesized Data Construction

Since acquiring real data from material samples is laborious and usually it requires advanced equipments in order to serial section the sample, synthesized data has been widely utilized in the material science for structural and physical properties analysis. Here, we utilize one recently popular tool developed by material scientists, named DREAM.3D [21], for constructing two 1-channel synthesized superalloy datasets: IN100-300 with dimension of $300 \times 300 \times 300$ and IN100-900 with dimension of $300 \times 300 \times 900$.

Proposed synthesizing process contains two major steps: 1) Constructing 2D image sequences of serial sectioned 3D synthesized grains; 2) Simulating intensities of the sectioning surface under microscope for each grain.

3D synthesized grains are constructed using DREAM.3D based on statistics collected from the real IN100 dataset. Specifically, we set the total number of grains as $2,000$ which is similar as in IN100, and the grain sizes are sampled from a Log-Norm distribution obtained from the grain sizes in IN100. Similar approach has been applied in [4]. Given a synthesized 3D grain volume, we split it along the z-axis and

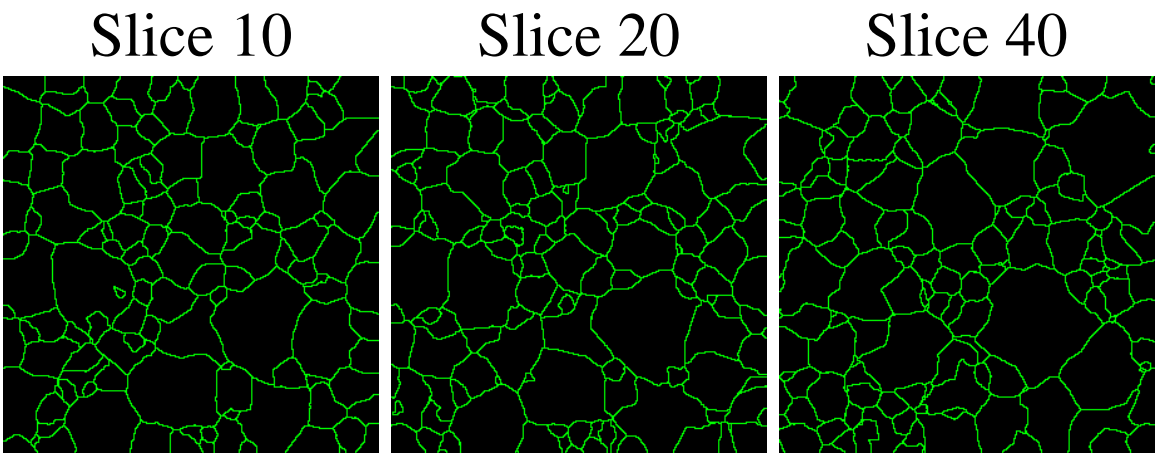


Figure 3.15: Illustrations of constructed 2D image slices of serial sectioned 3D synthesized grains.

represent it as a sequence of 2D image slices. Constructed 2D image sequences of serial sectioned 3D synthesized grains are illustrated in Figure 3.15.

The intensity for each grain is simulated through a non-parametric texture synthesis process [16]. Specifically, we first randomly crop 184 80×80 image patches from the real IN100 dataset. Notice that, cropping only be conducted within grains in order to avoid producing patches crossing grain boundaries. Figure 3.16 illustrates a set of sampled image patches. Then, for each synthesized grain, we fill its 2D sectioned surface with one randomly selected image patch using a texture synthesis algorithm proposed in [16]. Figure 3.17 shows a few examples of constructed 2D image slices for both two synthesized datasets.

Parameter Settings

Similar as for the real IN100 dataset, for the proposed method, we performed a grid search over the parameter space using the whole dataset, and selected $\omega = 4$ and $\lambda = 5$ which achieved the best performance. Compared with the real IN100 dataset, we use a smaller ω because the image scale becomes smaller from 671×671 to 300×300, and ω measures the area around objects for neighboring smoothness. We set the number of clusters in the color space as $k = 120$. The size of stable segments α and

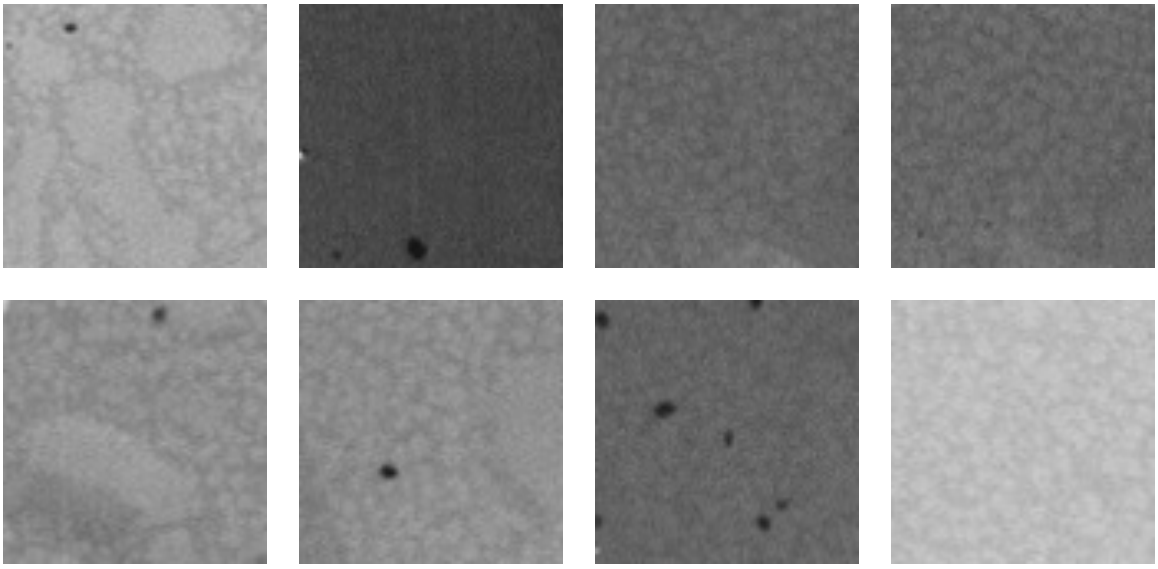


Figure 3.16: A set of image patches randomly sampled from the IN100 dataset.

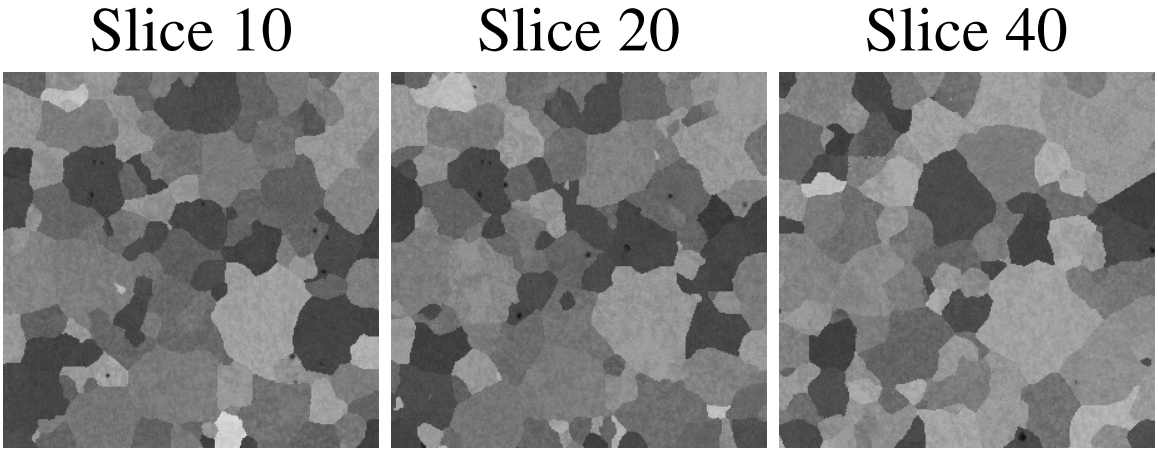


Figure 3.17: 2D image slices of serial sectioned 3D synthesized grains, after filling with sampled patches.

the minimum length of stable boundaries β are set as $\alpha = 80$ and $\beta = 15$. For the comparison algorithms, we have performed coarse grid searches, and report their best performances using obtained parameter settings.

Quantitative Results

Quantitative results for synthesized IN100-300 and IN100-900 datasets, are reported in Table 3.3 and Table 3.4 respectively. We also illustrate the results in Figure 3.18

Methods		Precision	Recall	F-score
2D Methods	EWCVT [44]	0.681457	0.920491	0.78314
	MeanShift [10]	0.944532	0.734022	0.826077
	GraphBased [18]	0.965788	0.640095	0.769914
	SRM [33]	0.992404	0.526994	0.688419
	gPb [3]	0.83115	0.719009	0.771023
	NormalizedCuts [39]	0.715336	0.82262	0.765236
3D/Propagated Methods	MCEWCVT [7]	0.819851	0.716402	0.764644
	3D Levelset [48]	0.676494	0.714562	0.695007
	3D Watershed [31]	0.879059	0.60455	0.716408
	StreamGBH [51]	0.604455	0.46982	0.528701
Proposed		**0.947256**	**0.865704**	**0.904646**

Table 3.3: Quantitative comparison of 2D/3D/Propagated segmentation methods on the IN100-300 dataset.

Methods		Precision	Recall	F-score
2D Methods	EWCVT [44]	0.683125	0.91619	0.782675
	MeanShift [10]	0.948733	0.732826	0.826919
	GraphBased [18]	0.987743	0.470167	0.637082
	SRM [33]	0.99107	0.540443	0.699461
	gPb [3]	0.827414	0.718046	0.76886
	NormalizedCuts [39]	0.712434	0.821097	0.762916
3D/Propagated Methods	MCEWCVT [7]	0.838043	0.712298	0.770071
	3D Levelset [48]	0.640096	0.707052	0.67191
	3D Watershed [31]	0.921154	0.516437	0.661826
	StreamGBH [51]	0.591641	0.540961	0.565167
Proposed		**0.945981**	**0.860409**	**0.901168**

Table 3.4: Quantitative comparison of 2D/3D/Propagated segmentation methods on the IN100-900 dataset.

and Figure 3.19 as well. From the results, we can see that almost all evaluated methods achieve lower performance than it in the real IN100 dataset. The reason is that the real IN100 dataset provides 4 different channels for easier and better locating grain boundaries than the constructed 1-channel synthesized datasets. The proposed method has achieved the best performance among all comparison 2D/3D/Stream algorithms on both two synthesized datasets. This suggests that, the proposed method is robust for even more challenging datasets.

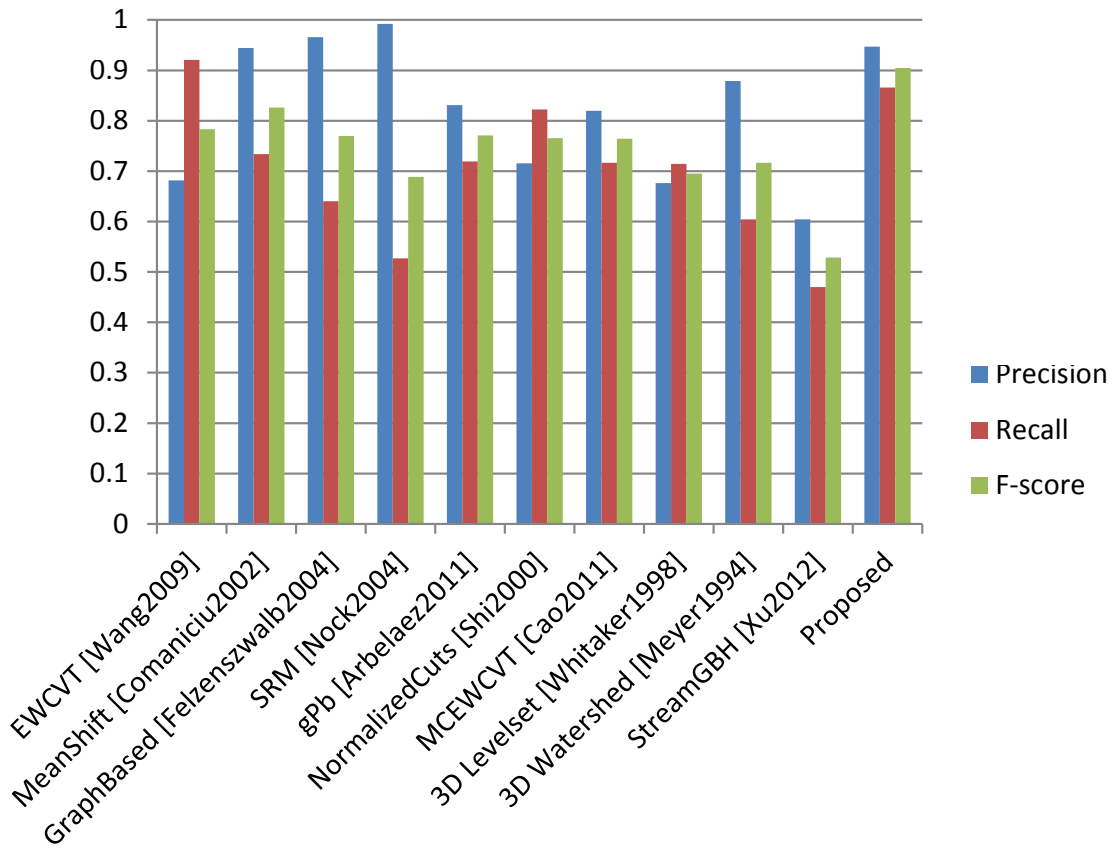


Figure 3.18: An illustration of quantitative comparison with 2D/3D/Propagated segmentation methods on the IN100-300 dataset.

Experiments on Long Distance Segment Propagation

In this section, we evaluate the performance of proposed non-rigid transformation based association method for long distance segment structure propagation. Then we compare its performance with it of several state-of-the-art association algorithms. In the following, we introduce the dataset utilized for long distance propagation evaluation, the measurement metric, and report quantitative evaluation results.

Fiber Dataset

As described in Section 3.5, and shown in Figure 3.10, 3.11, 3.12, and 3.13, in the IN100 dataset, the variation of segment structures between two neighboring image

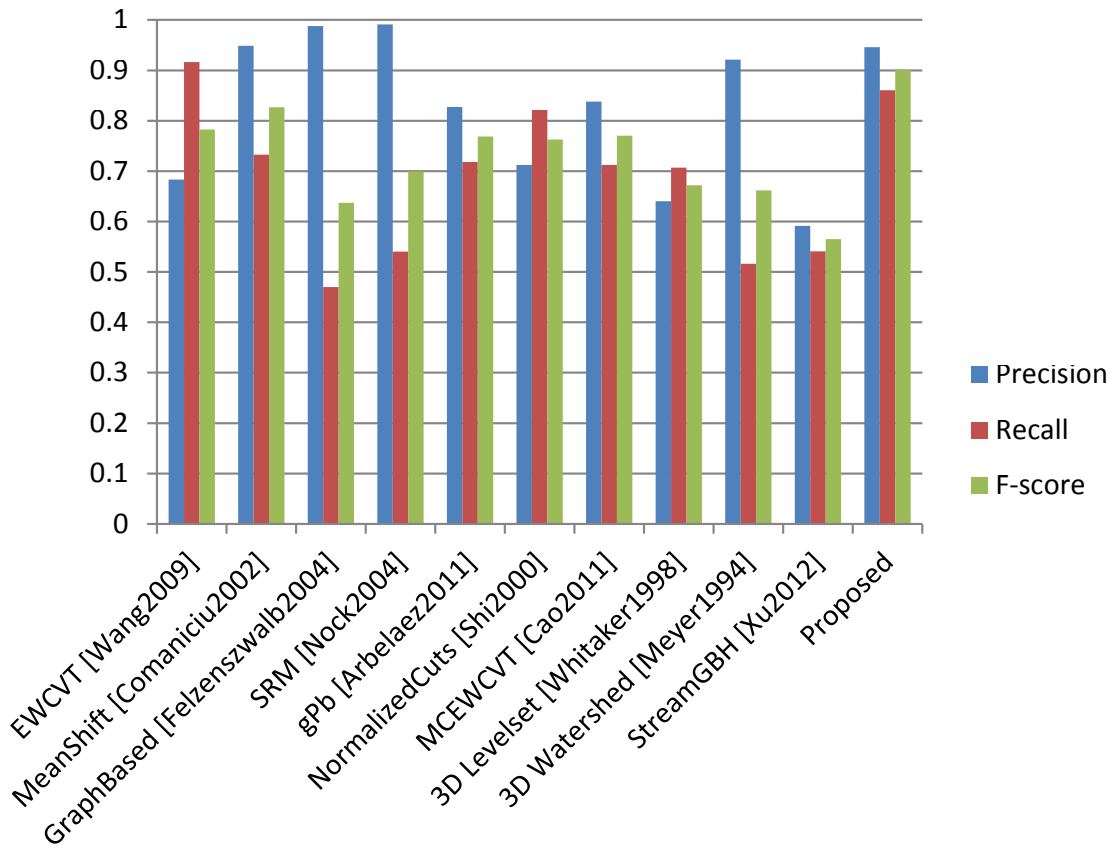


Figure 3.19: An illustration of quantitative comparison with 2D/3D/Propagated segmentation methods on the IN100-900 dataset.

slices is quite subtle. Thus the IN100 dataset may not be sufficient to verify the effectiveness of long distance association algorithms. Therefore, we select another serial-sectioned material image dataset provided by our collaborators, for long distance propagation evaluation.

The new dataset, named fiber dataset, contains 100 sequential 2D image slices of a fiber reinforced composite material sample. Each image slice consists of $6 \times 6 = 36$ individual 1292×968 image tiles. So the total number of images in this dataset is $100 \times 36 = 3,600$. Sample slices from different tiles are shown in Figure 3.20, where these white ellipses are the objects of interest, and they are 2D serial sectioning surfaces of fibers. Along the image sequences, this dataset also provides detected 2D fiber segments on each slice, as illustrated in Figure 3.20.

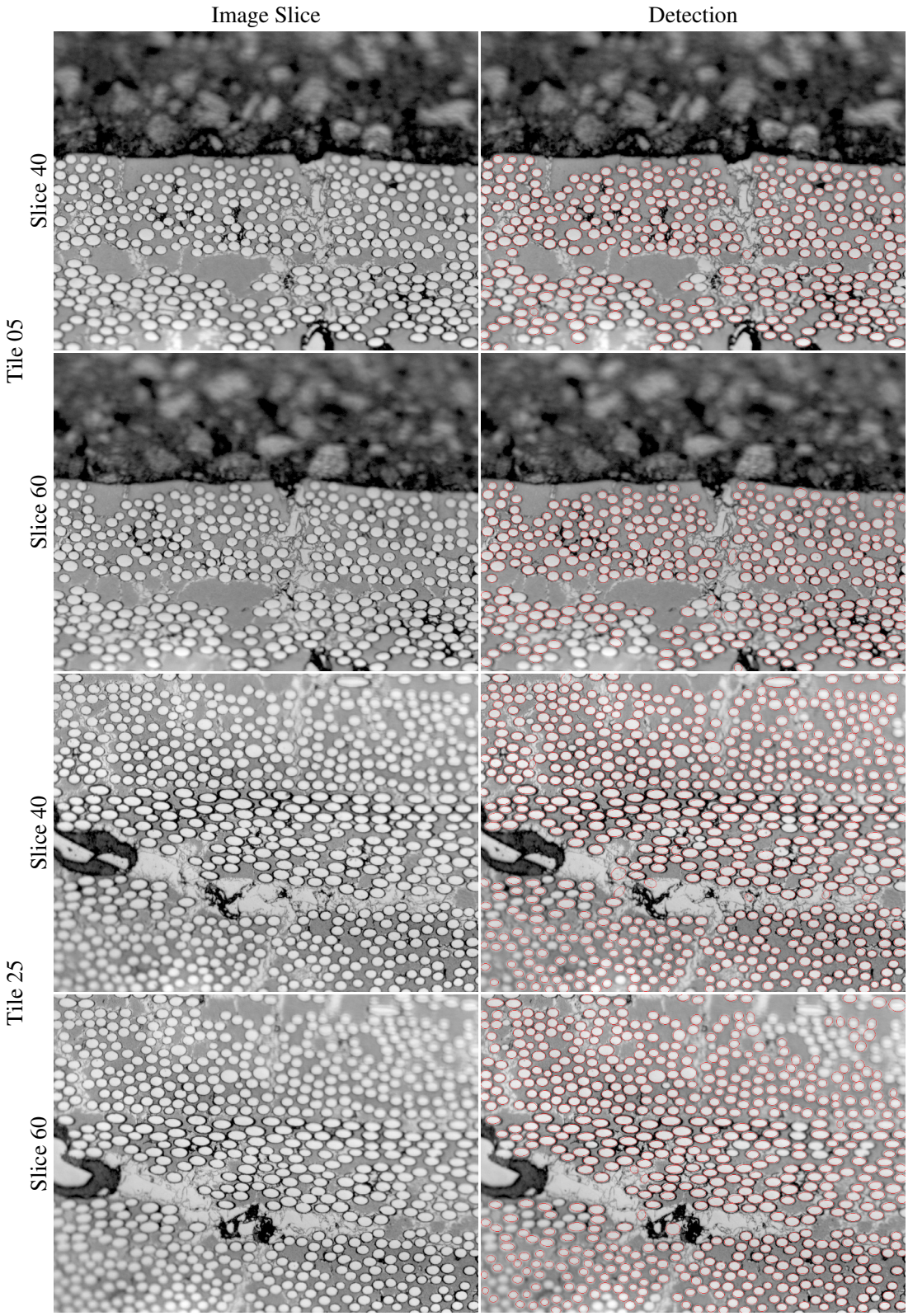


Figure 3.20: Illustrations of the fiber dataset. For each image slice, we also visualize provided fiber detections right next to each image, highlighted by red ellipses.

We can consider these detected fiber segments as grain segments in the IN100 dataset, then the stable segment structure constructed as a segment graph, and its propagation are same as for the grain image sequence. However, the major difference is that, compared with the IN100 dataset, the variation of segment structures among image slices is larger. For example, as shown in Figure 3.20, the structure of detected fibers in slice 20 and in slice 40 is quite different due to the fiber movement. Thus we propose to utilize the fiber dataset to evaluate the performance of long distance segment structure propagation.

To further investigate the robustness of a algorithm under different propagation distances, we construct a set of additional fiber datasets by manually increasing the inter-slice distance, and report evaluation results on each of them. Specifically, given a integer $s \in [0, 100]$, we can select a subset of slices by skipping every s intermediate slices. For example, for $s = 1$, the constructed subset sequence is $\{I^0, I^2, I^4, \ldots, I^{98}\}$; for $s = 4$, the constructed subset sequence is $\{I^0, I^5, I^{10}, \ldots, I^{90}\}$. In the experiments, we set $s \in \{0, 1, 2, \ldots, 9\}$ and report evaluation results of different algorithms under each slice skipping setting.

Evaluation Metric

As discussed in Section 3.4, it is naturally to model the segment structure propagation as a multi-target tracking problem. Therefore, we quantitatively measure the propagation performance using one standard multi-target tracking metric, the multiple object tracking accuracy (MOTA) [4], which is defined as

$$\text{MOTA} = 1 - \frac{\sum_i (\text{m}_i + \text{fp}_i + \text{mme}_i)}{\sum_i \text{gt}_i} \tag{3.15}$$

where m_i, fp_i and mme_i are the number of missing detections, of false positive detections, and of mis-matched associations, respectively, on image slice i. gt_i is the total

number of annotated ground truth detections on slice i. Thus, higher MOTA value indicates there is less tracking error in the obtained results.

Quantitative Comparisons

We compare the proposed non-rigid transformation based propagation method with four state-of-the-art multi-target tracking algorithms from both computer vision and biomedical communities, including, integer linear programming based tracking (DP-NMS) [34], motion dynamics based tracking (SMOT) [12], continuous energy based tracking (CEM) [32], and individual detection linking using the Viterbi algorithm (KTH) [29]. Parameters of all comparison algorithms are to their default values. Notice that, since the fiber dataset we utilized already provides object detection results, therefore, all involved algorithms share the same set of detections as input.

The quantitative results are illustrated in the left of Figure 3.21, from which we can see that, for the original fiber dataset, the proposed method outperforms other comparison algorithms, and is able to achieve more than 99% accuracy; while with the increasing of propagation distance, the proposed method still achieves higher performance than others. This indicates that, proposed non-rigid transformation based method indeed helps find correspondence between propagated stable segment structure and constructed segment structure especially for two image slices that are far away from each other. We also report the MOTA comparison results on a few individual tiles in Figure 3.22, where the proposed non-rigid transformation based method outperforms others by significant margins when increasing the number of skipped slices.

Moreover, we also compare the running time of different methods in order to demonstrate the computational efficiency of the proposed method, as illustrated in the right panel of Figure 3.21. We can see that the proposed algorithm achieves around a speedup of 9× compared with the CEM algorithm.

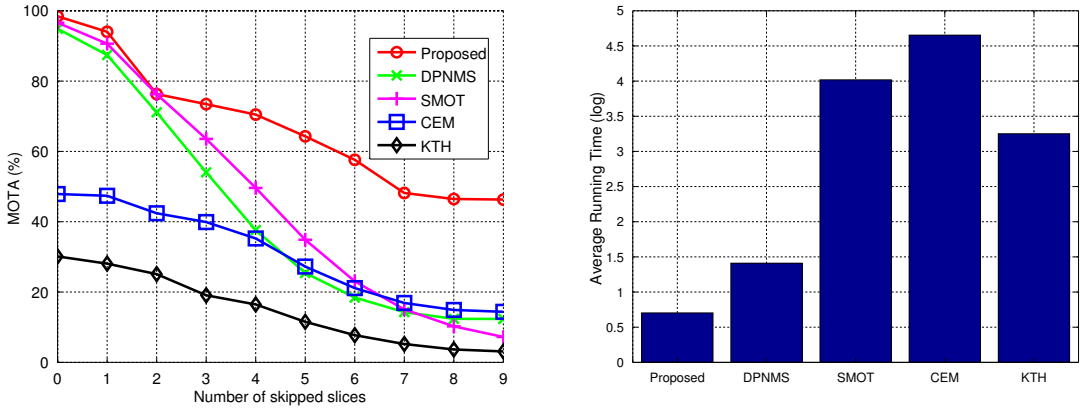


Figure 3.21: Quantitative comparison of different multi-target tracking methods for long distance segment structure propagation. MOTA is shown in left, and average running time (after log transformation) is shown in right.

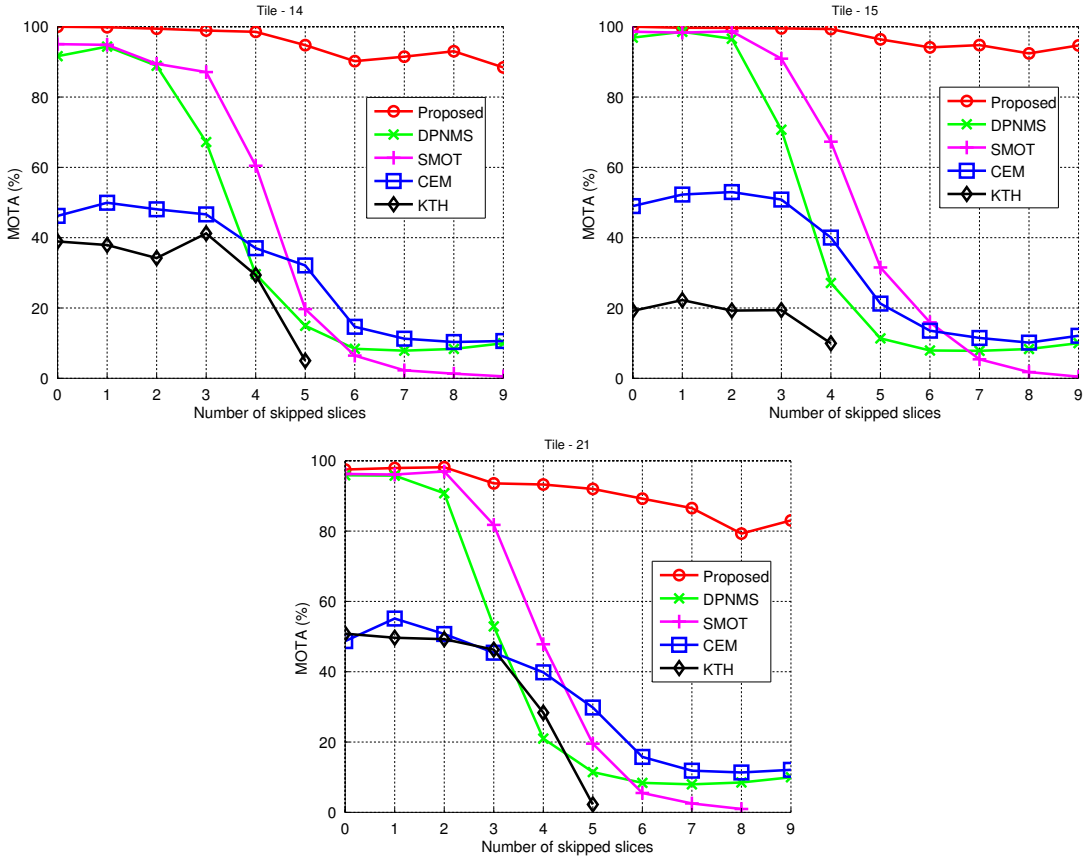


Figure 3.22: Quantitative comparison of different multi-target tracking methods for long distance segment structure propagation on a few individual tiles.

3.6 Discussion

In this chapter, we proposed an Edge-Weighted Centroid Voronoi Tessellation based method that can propagate structural consistency constraints from slice to slice, which is used to automatically segment 3D grain images. As a volumetric segmentation algorithm, the proposed method can automatically extract grain structures on all the slices, starting from the segmentation on the first slice, which can be constructed by any automatic 2D segmentation algorithm. The proposed propagation-based method is able to: 1) segment a large number of superalloy image slices efficiently, 2) preserve structural consistency across slices, and 3) easily correspond the segments across slices. For cases where the inter-slice distance is too large to find correspondence of propagated stable structures on the next slice, we further proposed a non-rigid transformation based association method. We conducted experiments on a 3D superalloy image dataset with 170 image slices. To verify the effectiveness of the proposed association method, we conducted experiments on a serial-sectioned fiber image dataset with $3,600$ image slices. Both qualitative and quantitative results indicate that the proposed method outperforms the comparison algorithms and is robust even when propagated through a large number of slices, and in the presence of strong noise and corruption.

Chapter 4

Intra-Image Propagation

For the intra-image propagation, the existing segmentation of an image is propagated and utilized for obtaining segmentation of the same image but under different conditions, for example, in different scales. As discussed in Section 1.1, in order to capture boundaries of objects with multiple scales in natural images, hierarchical or multiscale image segmentation is usually utilized. Here we propose a multiscale superpixel/supervoxel method, based on the EWCVT method introduced in Section 2.1, to capture object boundaries under different scales.

In this chapter, we first describe the proposed Hierarchical Edge-Weighted Centroidal Voronoi Tessellation (HEWCVT) for superpixel construction in Section 4.1, its extension to supervoxels in Section 4.4, then analyze its complexity and convergence condition in Section 4.3, discuss the importance of the simple-connectivity property for superpixels/supervoxels together with the proposed enforcement approach in Section 4.2, and finally describe the experiment results and discussions in Section 4.5

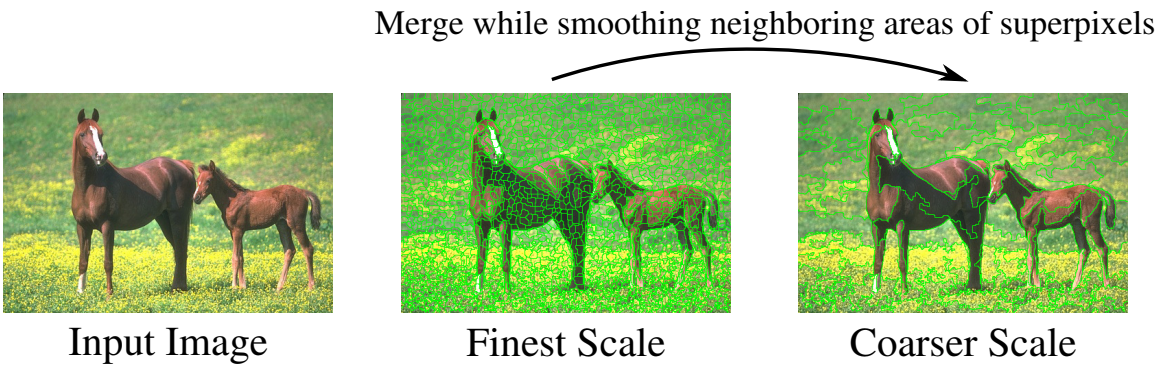
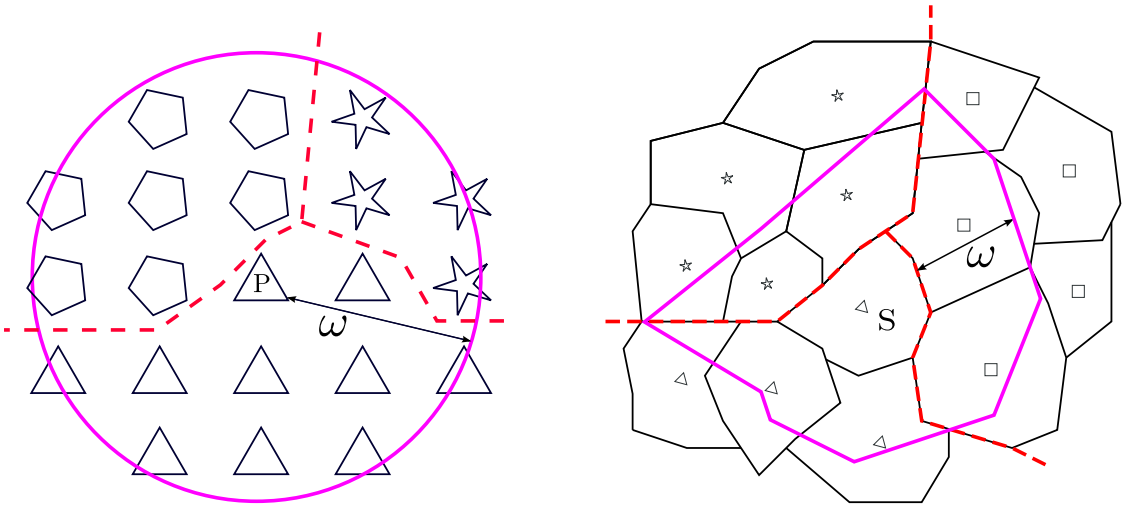


Figure 4.1: Overview of the proposed HEWCVT method for constructing superpixels/supervoxels in multiple scales.

and Section 4.6 respectively.

4.1 Hierarchical Edge-Weighted Centroidal Voronoi Tessellation

The proposed hierarchical method begins with an oversegmentation on pixels using the a modified EWCVT algorithm that strictly enforces the simple-connectivity of superpixels [45]. This oversegmentation is taken as the finest level of superpixels in the hierarchy. For the higher levels, we merge finer level superpixels with similar color features, meanwhile preserve superpixel connectivity and enforce the boundary smoothness of superpixels. An overview of the proposed HEWCVT method is shown in Figure 4.1.



(a) Boundary smoothness measurement for a pixel P. Each pixel is visualized as a polygon and its shape stands for the pixel's current cluster assignment.

(b) Boundary smoothness measurement for a superpixel S. Each polygon represents a superpixel and the shape of its center marker stands for the superpixel's current cluster assignment.

Figure 4.2: Boundary smoothness measurement illustrations. Dash lines are cluster boundaries. Pink curve indicates the local neighborhood area for smoothness measurement.

Superpixels Construction in the Finest Level

At the finest level we deal with the generation of superpixels directly from the image pixels. Let M_1 be the desired number of superpixels. Similar to the VCells algorithm proposed in [45], we first use the classic K-means with the Euclidean norm on pixel coordinates $\mathbb{I}$, to generate M_1 simply-connected and quasi-uniformly distributed superpixels on the input image. We also set $\rho \equiv 1$ here. Next we apply the VCells algorithm to the initial superpixel configuration where we only allow transferring of boundary pixels between neighbor clusters at each iteration. The whole algorithm is described in Algorithm 4.1. If $\pi(i,j)$ is different from the label of at least one of its 4 neighbors, i.e., $(i\pm 1, j)$ or $(i, j\pm 1)$, we say (i,j) is a boundary pixel, and denote $\mathcal{B}$ as the set of all boundary pixels. We remark that each pixel moving between neighbor clusters in Algorithm 4.1 will decrease the energy E_{ewcvt}, thus Algorithm 4.1 guarantees monotonic decreasing of E_{ewcvt} along the iterations till it terminates, see [44] for detailed discussions.

There is no guarantee to preserve the simple-connectivity property of each segment in the algorithm above. Thus in the end we perform a filtering step to further enforce the simple-connectivity of superpixels, which is widely used in other superpixel algorithms [1, 27, 43, 45] and will be described in Section 4.2.

Superpixels Construction in Higher Levels

At a higher level q ($q > 1$), we already have a superpixel from the previous level $q-1$, $\mathbb{S} = \{S_m\}_{m=1}^{M_{q-1}}$. Given the desired number of superpixels M_q ($M_q < M_{q-1}$) in Level q, we will merge adjacent superpixels to reach that goal according to minimization of certain energy function. Each superpixel is treated as a point and we will cluster them into M_q simply-connected parts, where $M_q < M_{q-1}$ is the desired number of superpixels in Level q. This way we can easily build a tree structure for superpixels between these two levels.

Algorithm 4.1 (Pixel-Level Superpixel Algorithm)

Input: The target 2D image $\mathbb{I}$ and the color function $\vec{u}$
M_1: Number of desired superpixels
niter: Maximum number of iterations

0 **Initialization**: Construct the initial superpixels of $\mathbb{I}$, $\{C_l\}_{l=1}^{M_1}$ using the k-means with the Euclidean distance and the feature function $\vec{u}^+$.
1 **FOR** each $C_k \in \{C_l\}_{l=1}^{M_1}$
2 Compute centroid $\vec{w}_k = \frac{1}{|C_k|} \sum_{(i,j)\in C_k} \vec{u}(i,j)$
3 **FOR** $iter = 1$ to $niter$
4 Create the set of boundary pixels $\mathcal{B}$
5 **FOR** each $(i,j) \in \mathcal{B}$
6 Find the closest centroid to the pixel (i,j)
$\vec{w}_k \in \{\vec{w}_l \mid l \in \pi(\mathcal{N}_4(i,j))\}$
w.r.t. the edge-weighted distance (Eq. (2.6))
7 **IF** $\pi(i,j) \neq k$
8 Set $\tilde{k} = \pi(i,j)$ and $\pi(i,j) = k$
9 Update $\vec{w}_k$, $\vec{w}_{\tilde{k}}$
10 **IF** there is no cluster index change
11 Break
12 Perform the simple-connectivity filtering

Output: The cluster/superpixel index function π

The initialization step is different from that in the finest level. The most intuitive idea is to apply the k-means clustering on the set of average coordinates of all superpixels constructed in the previous level. However, the merged superpixels may not be simply-connected. Instead, we first build a superpixel graph $G = (V, E, \mathcal{E})$, where V consists of all the previous level's superpixels $\{S_m\}_{m=1}^{M_{q-1}}$ and E is the set of all pairs of neighbor superpixels. The edge weight for $(S_a, S_b) \in E$ is defined as

$$\mathcal{E}(S_a, S_b) = \frac{\|\vec{u}(S_a) - \vec{u}(S_b)\|}{\max_{(S_a,S_b)\in E} \|\vec{u}(S_a) - \vec{u}(S_b)\|} \tag{4.1}$$

where $\vec{u}(S) = \frac{1}{|S|} \sum_{(i,j)\in S} \vec{u}(i,j)$ denotes the average color vector of all the pixels belonging to the superpixel S. Then the superpixel graph G will be partitioned into M_q subgraphs which are considered as initialized superpixels at level q. The proposed HEWCVT method will refine initialized superpixels later. Therefore, any graph partition algorithm can be used for this initialization. Based on algorithm

efficiency and code availability, we choose the METIS algorithm [24] here.

We define the density function ρ on $\mathbb{S}$ as $\rho(S_m) = |S_m|$, i.e., the number of pixels contained in the superpixel $S_m \in \mathbb{S}$. Let $\mathcal{C}^{sp} = \{C_l^{sp}\}_{l=1}^{M_q}$ be a clustering of $\mathbb{S}$ and $\mathcal{W} = \{\vec{w}_l\}_{l=1}^{M_q}$ be an arbitrary set of color vectors. Then we define the new CVT clustering energy as

$$E_{cvt-sp}(\mathcal{C}^{sp}, \mathcal{W}) = \sum_{l=1}^{M_q} \sum_{S \in C_l^{sp}} \rho(S) \|\vec{u}(S) - \vec{w}_l\|^2. \tag{4.2}$$

In order to measure the boundary length (or the smoothness) of superpixels, we propose an edge energy for the superpixel image. As illustrated in Figure 4.2b, we define the local neighborhood $\mathbb{N}_\omega(S)$ for a superpixel $S \in \mathbb{S}$ as

$$\mathbb{N}_\omega(S) = \bigcup_{(i,j) \in \mathcal{B}(S)} \mathbb{N}_\omega(i,j) - S$$

where $\mathcal{B}(S)$ denotes the set of all boundary pixels of the superpixel S. Then we define the edge energy as

$$E_{edge-sp}(\mathcal{C}^{sp}) = \sum_{S \in \mathbb{S}} \sum_{(i,j) \in \mathbb{N}_\omega(S)} \Gamma_S(i,j) \tag{4.3}$$

where $\Gamma_S(i,j) : \mathbb{N}_\omega(S) \to \{0,1\}$ is an indicator function, similar as $\chi(i,j)$ in Eq. 2.3, and is defined by

$$\Gamma_S(i,j) = \begin{cases} 1 & \text{if } \pi(i,j) \neq \pi(S) \\ 0 & \text{otherwise} \end{cases}$$

where $\pi(S)$ returns the cluster index of the superpixel S in $\mathcal{C}^{sp}$.

Finally, the edge-weighted CVT clustering energy for superpixels can be defined as

$$E_{ewcvt-sp}(\mathcal{C}^{sp}, \mathcal{W}) = E_{cvt-sp}(\mathcal{C}^{sp}, \mathcal{W}) + \lambda E_{edge-sp}(\mathcal{C}^{sp}). \tag{4.4}$$

We can derive the distance from a superpixel S to a cluster center $\vec{w}_k$ corresponding to the above energy as

$$dist(S, \vec{w}_k) = \sqrt{\rho(S) \parallel \vec{u}(S) - \vec{w}_k \parallel^2 + 2\lambda \tilde{n}_k(S)} \tag{4.5}$$

Algorithm 4.2 (Higher Level Superpixel Merging Algorithm)

Input: The superpixel image $\mathbb{S}$ and the color function $\vec{u}$
M_q: Number of desired superpixels
niter: Maximum number of iterations

0 **Initialization**: Construct the superpixel graph G and partition $\mathbb{S}$ into M_q simply-connected regions $\{C_l^{sp}\}_{l=1}^{M_q}$ using METIS [24]

1 **FOR** each $C_k^{sp} \in \{C_l^{sp}\}_{l=1}^{M_q}$

2 Compute centroid $\vec{w}_k = \frac{\sum_{S \in C_k^{sp}} \rho(S)\vec{u}(S)}{\sum_{S \in C_k^{sp}} \rho(S)}$

3 **FOR** $iter = 1$ to $niter$

4 Create the set of boundary superpixels $\mathcal{B}(\mathbb{S})$

5 **FOR** each $S \in \mathcal{B}(\mathbb{S})$

6 Find the closest centroid to S
$\vec{w}_k \in \{\vec{w}_l \mid l \in \pi(\mathcal{N}(S)\}$
w.r.t. the edge-weighted distance (Eq. (4.5))

7 **IF** $\pi(S) \neq k$

8 Set $\tilde{k} = \pi(S)$ and $\pi(S) = k$

9 Update $\vec{w}_k$, $\vec{w}_{\tilde{k}}$

10 **IF** there is no cluster index change

11 Break

12 Perform the simple-connectivity filtering

Output: The cluster/superpixel index function π

where $\tilde{n}_k(S)$ measures the number of inconsistent pixels in the neighborhood of the superpixel S: $\tilde{n}_k(S) = |\mathbb{N}_\omega(S)| - n_k(S)$ with $n_k(S) = \sum_{(i,j)\in\mathbb{N}_\omega(S)} \pi(i,j) \neq k$.

Furthermore, in order to keep superpixels simply connected, we follow the idea in the finest level (Section 4.1), i.e., only superpixels located at cluster boundaries will be considered during the clustering, and we only allow cluster index change among adjacent clusters. The whole algorithm is described in Algorithm 4.2. We again remark that similar to Algorithm 4.1, Algorithm 4.2 guarantees monotonic decreasing of $E_{ewcvt-sp}$ along the iterations till it terminates.

Adaptive Determination of the Edge Energy Weight

The energy weight parameter λ defined in Eqs. (2.5) and (4.4) balances the ratio between the CVT clustering energy E_{cvt} (or E_{cvt-sp}) and the edge energy E_{edge} (or

$E_{edge-sp}$). However, these energies are varying from different images/videos and/or change along different scale levels. Especially in video segmentation, different videos also have variant number of frames and frame rates. Thus a fixed λ is obviously inappropriate. Instead we aim at controlling the ratio between E_{cvt} and λE_{edge}. Therefore, given an predetermined energy ratio θ that $\frac{E_{cvt}}{\lambda E_{edge}} = \frac{E_{cvt-sp}}{\lambda E_{edge-sp}} = \theta$, we can adjust λ adaptively by setting $\lambda^{(iter)} = \frac{E_{cvt}^{(iter-1)}}{\theta E_{edge}^{(iter-1)}}$ at each iteration in Algorithm 4.1 and $\lambda^{(iter)} = \frac{E_{cvt-sp}^{(iter-1)}}{\theta E_{edge-sp}^{(iter-1)}}$ in Algorithm 4.2.

4.2 Simple-Connectivity Enforcement

Although we have enforced that the pixel/superpixel transferring can only occur among adjacent clusters, due to the image noises, few superpixels may still break into several disconnected parts and/or contain holes (especially in 3D cases). Thus after the HEWCVT clustering process, we merge small ($|S| \leq \varepsilon$) and isolated superpixels into their surroundings. Similar post-step has been applied in several state-of-the-art superpixel/supervoxel methods [1, 27, 43, 45].

Specifically, there are two cases: 1) in the finest level, for each pixel p in a small or isolated superpixel S, we first locate its nearest neighbor pixel p' in surrounding superpixels S' and then merge p into S'; 2) in the higher levels, we can associate each pixel p in S with a surrounding superpixels as in the finest level, and this association can be viewed as a vote from p to a surrounding superpixel. We merge S into S' that has the majority vote.

4.3 Complexity and Convergence Analysis

The Finest Level Superpixel Algorithm 4.1 is equivalent to the VCells, and it contains two major steps: 1) initializing boundary pixels $\mathcal{B}$ which takes $\mathcal{O}(N)$ where N is the total number of image pixels; 2) EWCVT algorithm only considering boundary pixels which takes $\mathcal{O}\left(K\sqrt{M_1 \cdot N}\right)$ where K is the total number of iterations and M_1

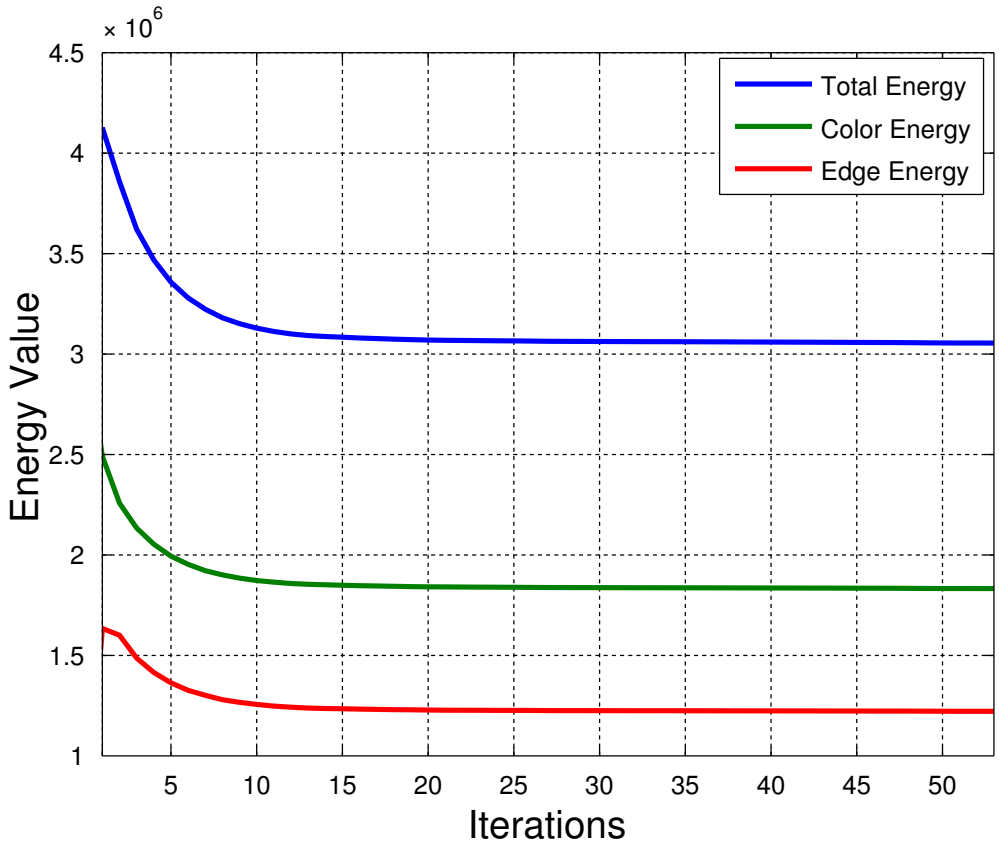


Figure 4.3: An illustration of energy convergence of the proposed HEWCVT, with $\theta = 1.5$, when constructing superpixels on a sample image.

is desired number of superpixels in the finest level. We refer the reader to [45] for more details about the complexity analysis of VCells.

Excluding the cost of boundary pixels initialization, for the Higher Level Superpixel Merging Algorithm 4.2, as we only consider the boundary superpixels, thus the computational cost in each iteration is $\mathcal{O}\left(n_{\mathcal{B}(S)} \cdot n_{\mathcal{B}}\right)$, where $n_{\mathcal{B}(S)}$ is the number of boundary superpixels, and $n_{\mathcal{B}}$ is the number of boundary pixels utilized for measuring proposed superpixel boundary smoothness. Each merged superpixel should contain approximately $\frac{M_{q-1}}{M_q}$ superpixels from the previous level, where M_{q-1} is the number of superpixels in the previous level and M_q is desired number of merged superpixels in current level, thus there are $\sqrt{\frac{M_{q-1}}{M_q}}$ boundary superpixels. Similarly the number of boundary pixels in a superpixel can be approximated by $\sqrt{\frac{N}{M_{q-1}}}$. Therefore,

$$\mathcal{O}\left(n_{\mathcal{B}(S)} \cdot n_{\mathcal{B}}\right) \sim \mathcal{O}\left(M_q \cdot \sqrt{\frac{M_{q-1}}{M_q}} \cdot \sqrt{\frac{N}{M_{q-1}}}\right) \sim \mathcal{O}\left(\sqrt{M_q \cdot N}\right).$$

For K iterations, we have $\mathcal{O}\left(K\sqrt{M_q \cdot N}\right)$. Overall, the complexity of proposed HEWCVT method is $\mathcal{O}\left(N + K\sqrt{M \cdot N}\right) = \mathcal{O}(N)$ where M is desired number of superpixels in a hierarchy level.

Both of Algorithm 4.1 and Algorithm 4.2 will converge to a local minimum of the defined HEWCVT energy. We illustrate the value change of the total energy, the color energy and the edge energy along iterations on a sample image in Figure 4.3. In this example, we set the desired ratio between the color energy and the edge energy, i.e., $\theta = 1.5$. We can see that, all three energies decrease quickly and converge to local minimal values, while the ratio between the color energy and the edge energy always remains the same as the desired value. For mathematical proofs on EWCVT-based energy convergence, please see [44] for details.

4.4 Extension to Supervoxels

We can easily extend the proposed hierarchical method into 3D case. The major difference is the neighbor system among voxels and supervoxels. Instead of 4-neighbor system in 2D case, we use 6-neighborhood for the voxel level oversegmentation. We note that more complex neighbor systems also can be used.

Another issue is that in the 2D case we assume the units of all coordinate directions are the same. For 3D images this assumption is still valid in most situations. However, for video data, the unit of the temporal direction could be different from those of spatial axises. Therefore for video data, one could use $\mathbb{I}_{3D} = H * (i, j, k)^T$ where $H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \gamma_k \end{bmatrix}$ is a scaling matrix and γ_k is data dependent. In the video experiments, we just simply used $H = I_{3\times 3}$ and it worked fine for the test video data.

We tested the proposed HEWCVT method on three standard image/video segmentation benchmarks which have been widely used for evaluating the performance of superpixels/supervoxels:

- the Berkeley Segmentation Dataset and Benchmark (BSDS300) [30], which consists of 300 color images of dimensions 481×321 or 321×481. Each image has been annotated by different subjects, thus obtained ground truth segments are at varying levels of granularity.

- the Weizmann image dataset [2], which consists of 200 color images of size approximately 300×225. Different from the BSDS300 dataset, subjects only annotated contours of the foreground objects. Based on the number of objects in an image, the whole dataset consists of two parts: images with single object (W1) and images with two objects (W2).

- the Xiph.org video dataset [8], which consists of 8 color videos of approximately 85 frames (240×160) each. The videos have been labeled frame by frame with temporal consistency taken into consideration.

Sample images overlaid with corresponding ground-truth boundaries are shown in Figure 4.4.

Evaluation Metrics

In order to quantitatively evaluate the performance of superpixels/supervoxels, we used human labeled segmentation as the ground truth because the superpixel/supervoxel boundaries should well align with the structural boundaries. Based on the ground truth, we applied three standard superpixel/supervoxel measurements: boundary recall, under-segmentation error and segmentation accuracy [1, 43, 51]. Note that

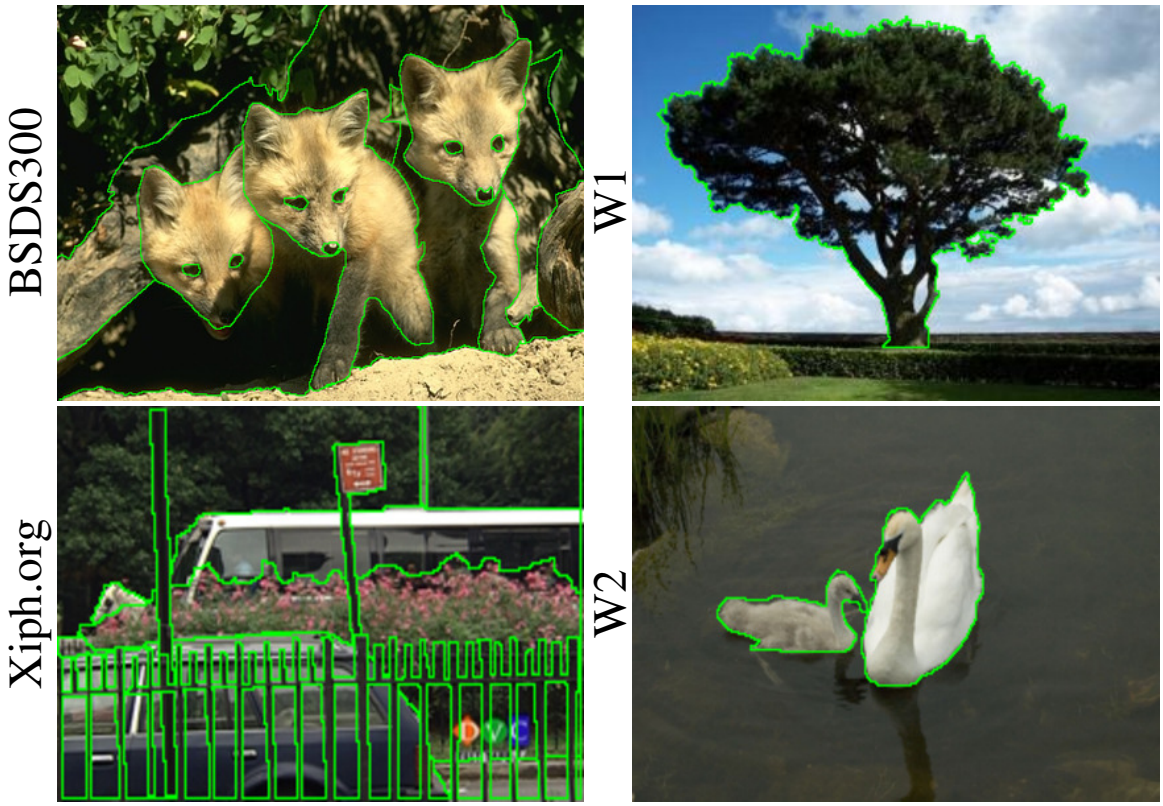


Figure 4.4: Sample images and their human annotated boundaries for superpixel and supervoxel evaluations.

in this paper we propose a superpixel/supervoxel method. Therefore, we use superpixel/supervoxel metrics instead of image segmentation metrics, such as metrics from the Berkeley segmentation dataset. For each of the three metrics we report the average values on each dataset.

Boundary Recall

This metric measures the fraction of ground truth boundaries that fall within a certain distance t of at least one superpixel/supervoxel boundary. It is formulated as

$$BR = \frac{\sum_{p \in \mathcal{B}(g)} \mathcal{I}\left[\min_{q \in \mathcal{B}(s)} \parallel p - q \parallel < t\right]}{|\mathcal{B}(g)|} \tag{4.6}$$

where $\mathcal{B}(g)$ is the union set of ground truth boundaries, $\mathcal{B}(s)$ is the union set of superpixel/supervoxel boundaries and $\mathcal{I}$ is an indicator function that returns 1 if a

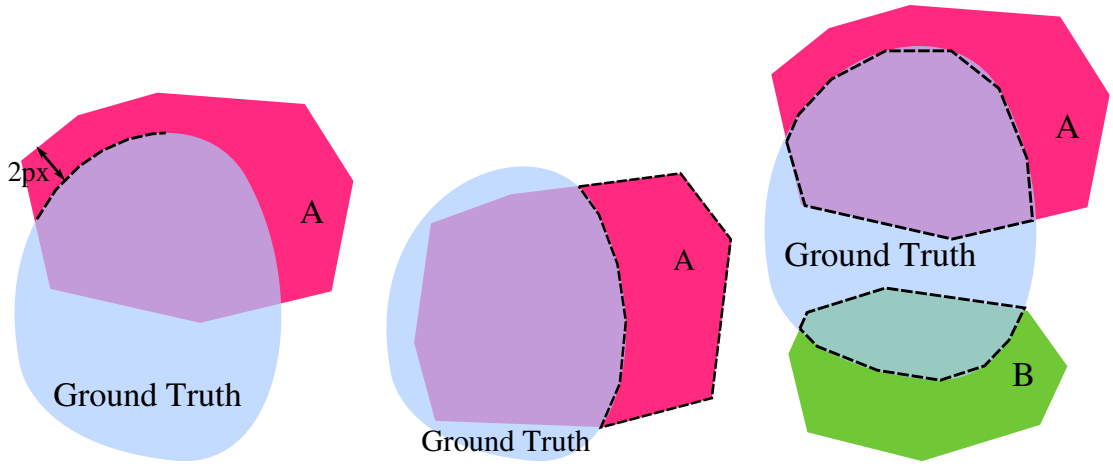


Figure 4.5: Illustrations of three superpixel/supervoxel evaluation metrics.

superpixel/supervoxel boundary pixel is close enough to the ground truth boundaries. We set $t = 2$ for both images and videos as in [1] and [51]. In general the larger the number of superpixels/supervoxels, the more boundaries, and the better the boundary recall. We illustrate this metric in the left panel of Figure 4.5.

Undersegmentation Error

This metric measures the fraction of superpixels/supervoxels that is leaked across the boundary of the ground-truth segments. For each ground truth segment g_i, we calculate the "bleeding" area of superpixels/supervoxels that overlap with g_i. It is formulated as

$$UE = \frac{\sum_{i=1}^{G} \left[(\sum_{s_j : s_j \cap g_i > r} \mid s_j \mid) - \mid g_i \mid \right]}{\sum_{i=1}^{G} \mid g_i \mid} \tag{4.7}$$

where $s_j \bigcap g_i$ is the overlapping between a superpixel/supervoxel s_j and a ground truth segment g_i. r is set to be 5% as in [1]. In general superpixels/supervoxels that tightly fit the ground truth segments result in a lower value of UE. We illustrate this metric in the middle panel of Figure 4.5.

	HEWCVT	VCells[45]	GraphCut[43]	SLIC[1]	LRW[38]
BSDS300	1.42s	1.32s	5.39s	0.27s	1090.55s
W1	0.24s	0.62s	1.97s	0.13s	1160.83s
W2	0.23s	0.55s	2.16s	0.12s	854.57s

	HEWCVT	GBH[22]	SWA[37]
Xiph.org	0.54s	0.47s	0.13s

Table 4.1: Average running time of different superpixel/supervoxel algorithms on several image/video datasets.

Segmentation Accuracy

This metric measures the fraction of a ground truth segment that is correctly classified by the superpixels/supervoxels, and we report the average fraction over all the ground truth segments. It is formulated as

$$SA = \frac{1}{G}\sum_{i=1}^{G}\frac{\sum_{s_j:s_j\cap g_i>c} \mid s_j \mid}{\mid g_i \mid} \tag{4.8}$$

where the overlapping ratio c specifies whether a ground truth segment is correctly classified or not and we set $c = 95\%$ as in [1, 43]. We illustrate this metric in the right panel of Figure 4.5.

In the following, we evaluate the proposed method under different parameter settings, discuss the principles of determining parameters, and compare the performance with 6 well known superpixel/supervoxel algorithms quantitatively and qualitatively. We do not include comparisons with other superpixel/supervoxel algorithms because according to the recent superpixel/supervoxel benchmark surveys [1, 51] the algorithms we have compared with have achieved the state-of-the-art performance and they have been widely used in different applications already.

We implemented the proposed method and the benchmark evaluation algorithm in C/C++. For the comparison algorithms, we used implementations published by their authors. All experiments were conducted on a Linux workstation with 8 GB memory and an Intel processor clocked at 2.4GHz with 8 cores. Average running time

M_q	4096	2048	1024	512	256	128
Image	$(2, 1.5)$	$(2, 1000)$	$(2, 1000)$	$(2, 1000)$	$(2, 1000)$	$(2, 1000)$

M_q	1000	...	100
Video	$(2, 1)$	$(2, 1)$	$(2, 1)$

Table 4.2: Parameter settings (ω, θ) that achieve the highest performance on each hierarchy level in the grid search on the validation sets.

of evaluated superpixel/supervoxel algorithms on all image/video datasets is shown in Table 4.1. Proposed HEWCVT method achieved comparable time efficiency among other algorithms in both superpixel and supervoxel constructions.

Parameters

There are two major parameters that can be tuned in the proposed HEWCVT method:

1. ω, which defines the radius of the 2D/3D local neighborhood region of a superpixel/supervoxel, as illustrated in Figure 4.2;

2. θ, the ratio between the CVT clustering energy and the edge energy, as defined in Section 4.1.

For the METIS algorithm utilized for initializing superpixel/supervoxel graphs in higher levels, we follow its default parameter settings, i.e., performing a k-way graph clustering with at most 10 iterations. We stop both Algorithm 4.1 and Algorithm 4.2 after $niter = 30$ and $niter = 200$ iterations for superpixel and supervoxel constructions respectively, since in practice we are already able to achieve converged HEWCVT energies.

To determine the values of ω and θ, we perform a grid search on validation sets, which consist of around 30% images and videos randomly selected among three datasets, and later we choose the parameter settings that achieve the highest performance across the three evaluation metrics. For the comparison algorithms, we did

the same grid search on the validation sets and reported performance on testing sets using obtained parameter settings.

It's impossible to enumerate all possible values and combinations of these two major parameters, thus we only tested values within certain ranges: $\omega \in \{2, 4, 8\}$, $\theta \in \{1, 1.5, 2, 4, 10, 200, 500, 1000\}$. For the hierarchy structure, we set $M_q \in \{4096, 2048, 1024, 512, 256, 128\}$ for the superpixel construction on images and $M_q \in \{1000, 900, \ldots, 100\}$ for the supervoxel construction on videos. For each hierarchy level, we pick the results with the highest performance in the previous level as the initialization. We list the parameter settings that achieve the highest performance on the validation sets in Table 4.2, and later we report performance of proposed method on testing sets using these fixed settings.

We also investigate the influence of different parameter values on the superpixel/supervoxel construction in the proposed HEWCVT method. Figure 4.6 illustrates the superpixel performance of the proposed HEWCVT under different (ω, θ) on the image validation set. From where we can see that, larger ω decreases the performance while larger θ always leads to better performance. The reason is that, compared with the whole image, the average size of constructed superpixels is quite small. Thus given a large local neighborhood region (large ω), the edge energy will dominant the total HEWCVT energy. Without balancing the energy ratio between the color energy E_{cvt} and the weighted edge energy λE_{edge} (requiring large θ), over-smoothed superpixel boundaries are not well adherent to the structural boundaries. Similar phenomenon has been observed on the video validation set as well.

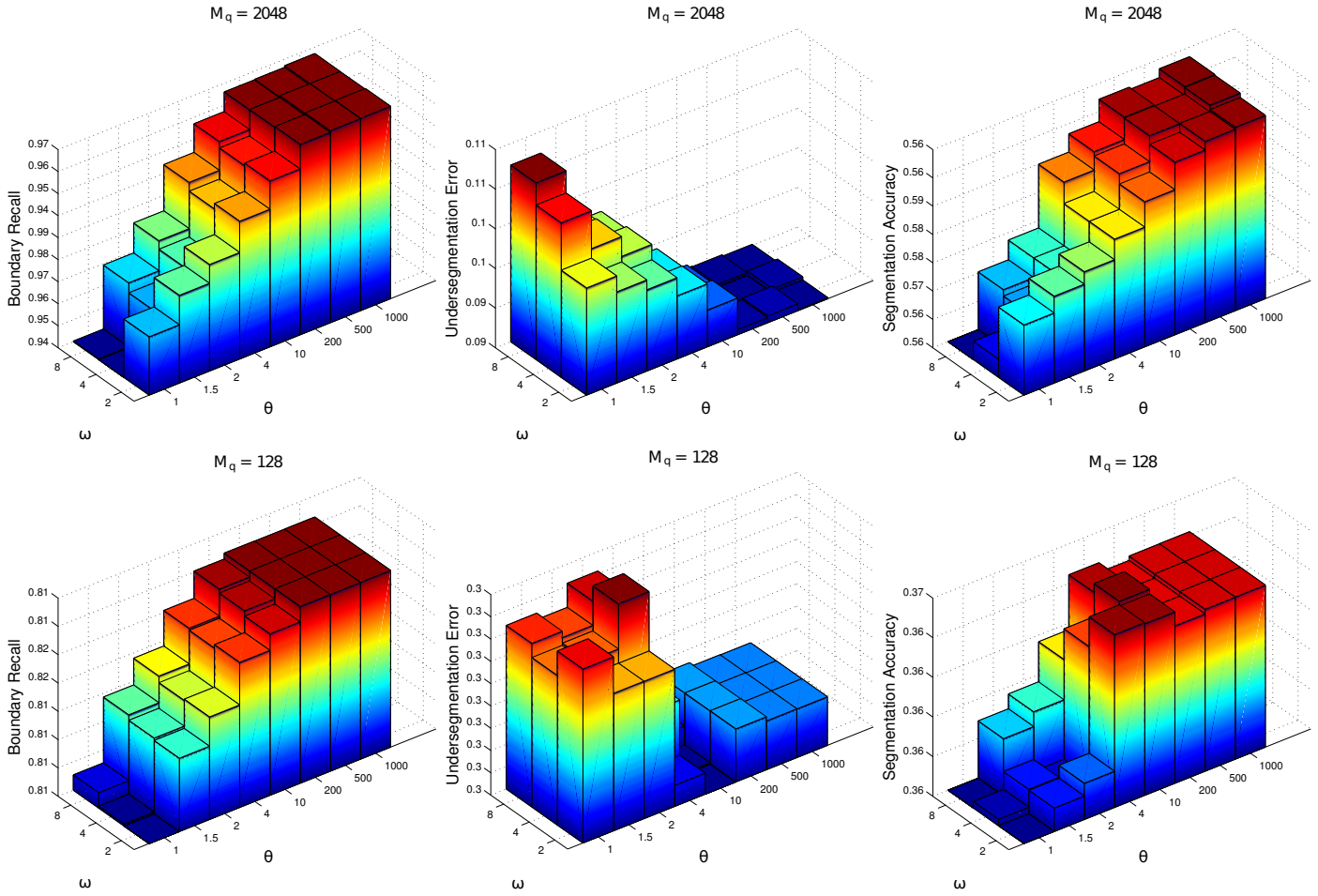


Figure 4.6: Superpixel performance, in terms of boundary recall, undersegmentation error and segmentation accuracy, of the proposed HEWCVT under different values of ω and θ in one hierarchy level on the image validation set. Better view in color.

Superpixel Evaluation

In order to evaluate the proposed HEWCVT method on the superpixel construction, we further compare it with other 4 state-of-the-art superpixel algorithms, including the VCells algorithm [45], a MRF model based algorithm (GraphCut) [43], the SLIC algorithm [1] and the LRW algorithm [38]. We do not include other superpixel algorithms such as the Turbopixel algorithm [27], and other segmentation based methods such as the NormalizedCut algorithm [36], Meanshift [10] and Quickshift [42] algorithms, into our comparison because: according to [1] SLIC outperforms many state-of-the-art superpixel and segmentation algorithms on the Berkeley dataset and we have included SLIC into our comparison.

We apply these superpixel algorithms on the two testing datasets: the BSDS300 dataset and the Weizmann dataset (including two parts: W1 and W2), and discuss both quantitative and qualitative results. Other than the standard superpixel metrics, we also evaluate the constructed superpixels in term of semantic image segmentation accuracy using the algorithm described in [20] and [19], where the constructed superpixels are utilized as an initialization for a CRF based pixel labeling algorithm. Similar evaluation approach has been used in [1] as well.

Quantitative Results

Quantitative results of superpixel construction for all the datasets are shown in Figure 4.7, 4.8, and 4.9 respectively. For the BSDS300 dataset, we can see that, proposed HEWCVT clearly achieves better performance in terms of both three metrics compared with other state-of-the-art methods. For both W1 and W2 datasets, EWCVT based methods: HEWCVT and VCells, outperforms other comparison algorithms, and HEWCVT achieves comparable performance with VCells on the W1 dataset. But for the W2 dataset, when the number of superpixels is small, VCells achieves better performance than HEWCVT in terms of undersegment error and segmentation

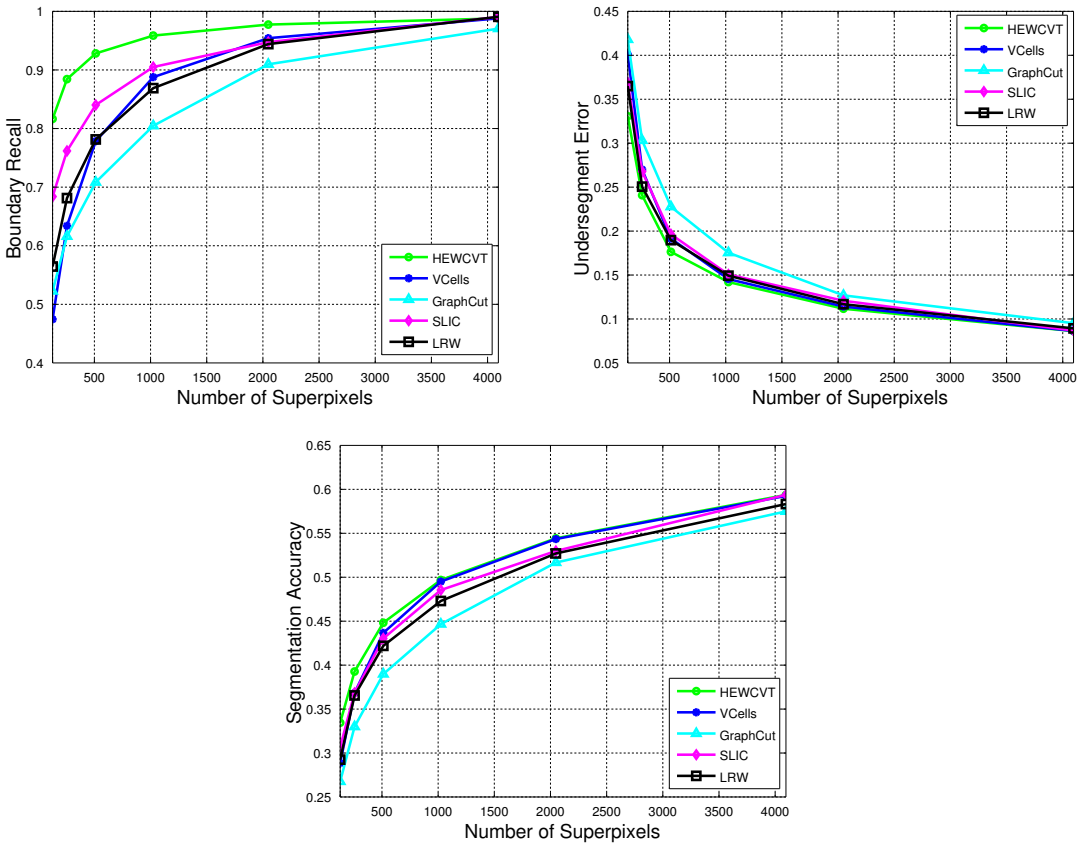


Figure 4.7: Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the BSDS300 dataset.

accuracy. The major reason is that, in the Weizmann dataset, only object's external contours have been annotated as the ground-truth, as shown in Figure 4.4, thus it favors superpixels constructed directly on a coarse scale without considering object's internal structures in finer scales. The proposed HEWCVT, however, achieves coarse scale superpixels using superpixels in finer scales, which leads to uneven external contours and lower performance than VCells that produces superpixels directly on the coarse scales.

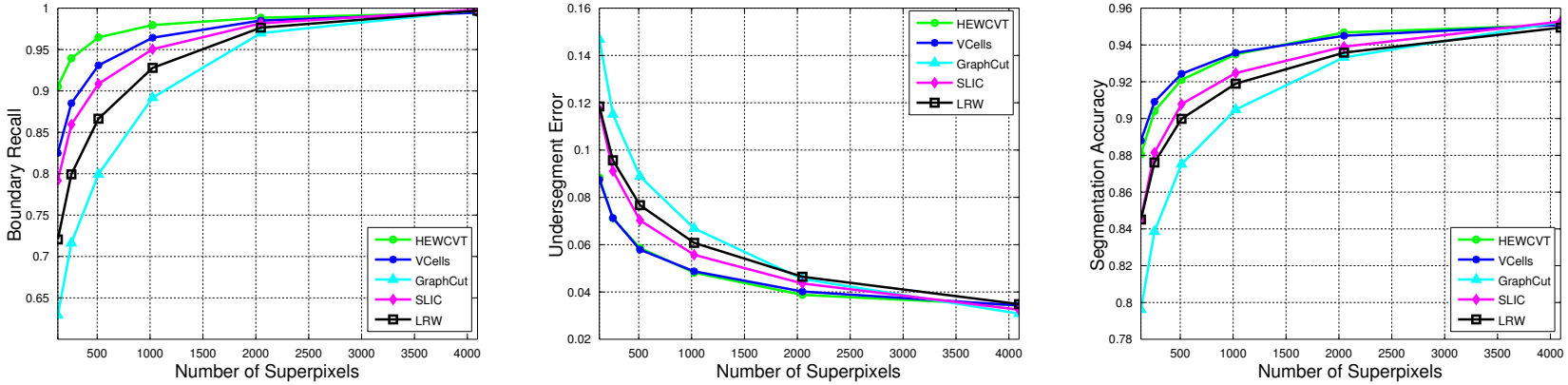


Figure 4.8: Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the W1 dataset.

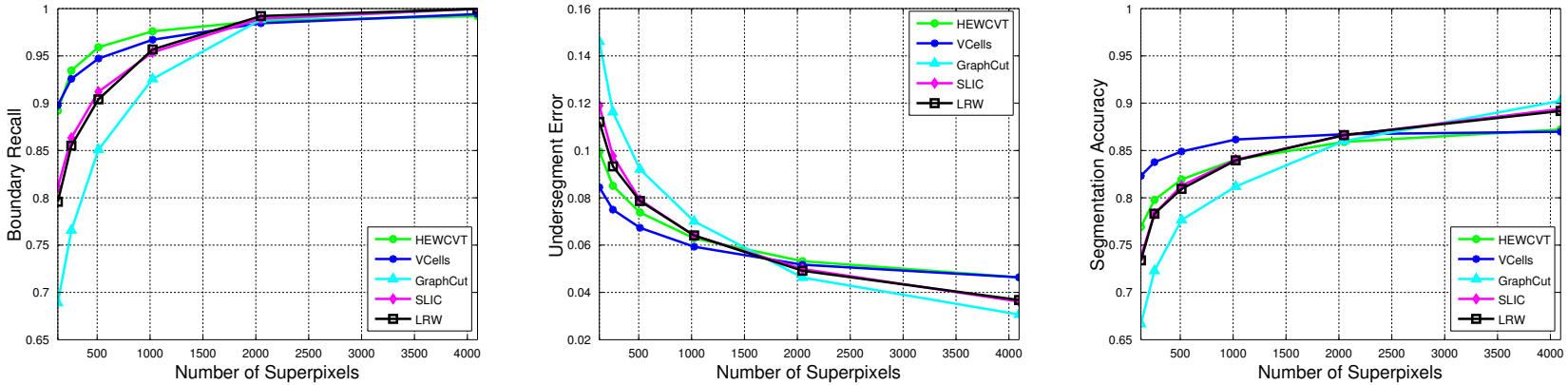


Figure 4.9: Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the W2 dataset.

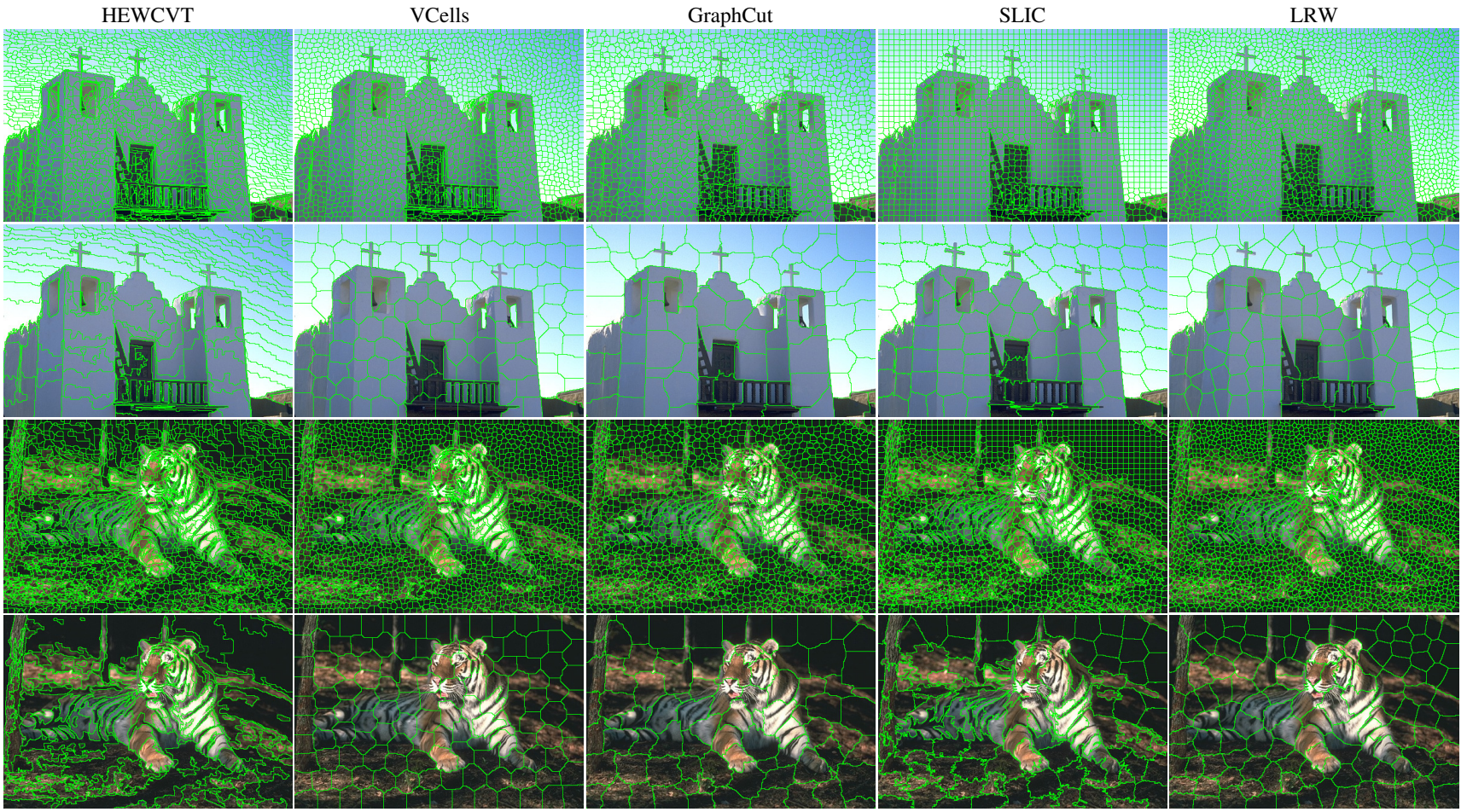


Figure 4.10: Qualitative comparisons of the four superpixel methods (HEWCVT, VCells, GraphCut, SLIC, LRW) on two images from the BSDS300 dataset. The numbers at the left indicate the desired number of superpixels. Better view in color.

Accuracy	HEWCVT	VCells[45]	GraphCut[43]	SLIC[1]	LRW[38]
MSRC[37]	76.2%	75.4%	73.2%	76.9%	74.6%
VOC2007[17]	25.9%	24.9%	23.9%	24.6%	24.7%

Table 4.3: Class-average segmentation accuracies on the MSRC dataset and the PASCAL VOC2007 dataset using superpixels constructed by different algorithms.

Qualitative Results

Sample results of constructed superpixels from all the datasets are shown in Figure 4.10 and 4.11. We can see that, compared with the four comparison methods, HEWCVT produces more uniform superpixels in the finest scale while catches structural boundaries more accurately in the coarsest scale.

Application on Semantic Image Segmentation

In order to comprehensively evaluate the effectiveness of constructed superpixels using different algorithms, we further investigate the performance of applications utilizing those produced superpixels, particularly, we focus on the application of superpixel based semantic image segmentation. Similar evaluation approach has been used in [1] as well.

In the semantic image segmentation task, the goal is to assign image pixels with predefined object class labels, e.x., tree, chair, and person. As illustrated in Figure 4.12, human annotated object labels for image pixels are visualized in different color. We investigate the semantic segmentation accuracy of two recently proposed semantic segmentation methods, [37] and [19], on two widely used datasets, [37] and [17], respectively.

Both two algorithms are superpixel based approach, where they first represent an image with superpixels, and then infer semantic labels using statistical models on superpixels. Thus, given the same semantic segmentation method, by plugging in superpixels constructed from different algorithms, we can evaluate the superpixel

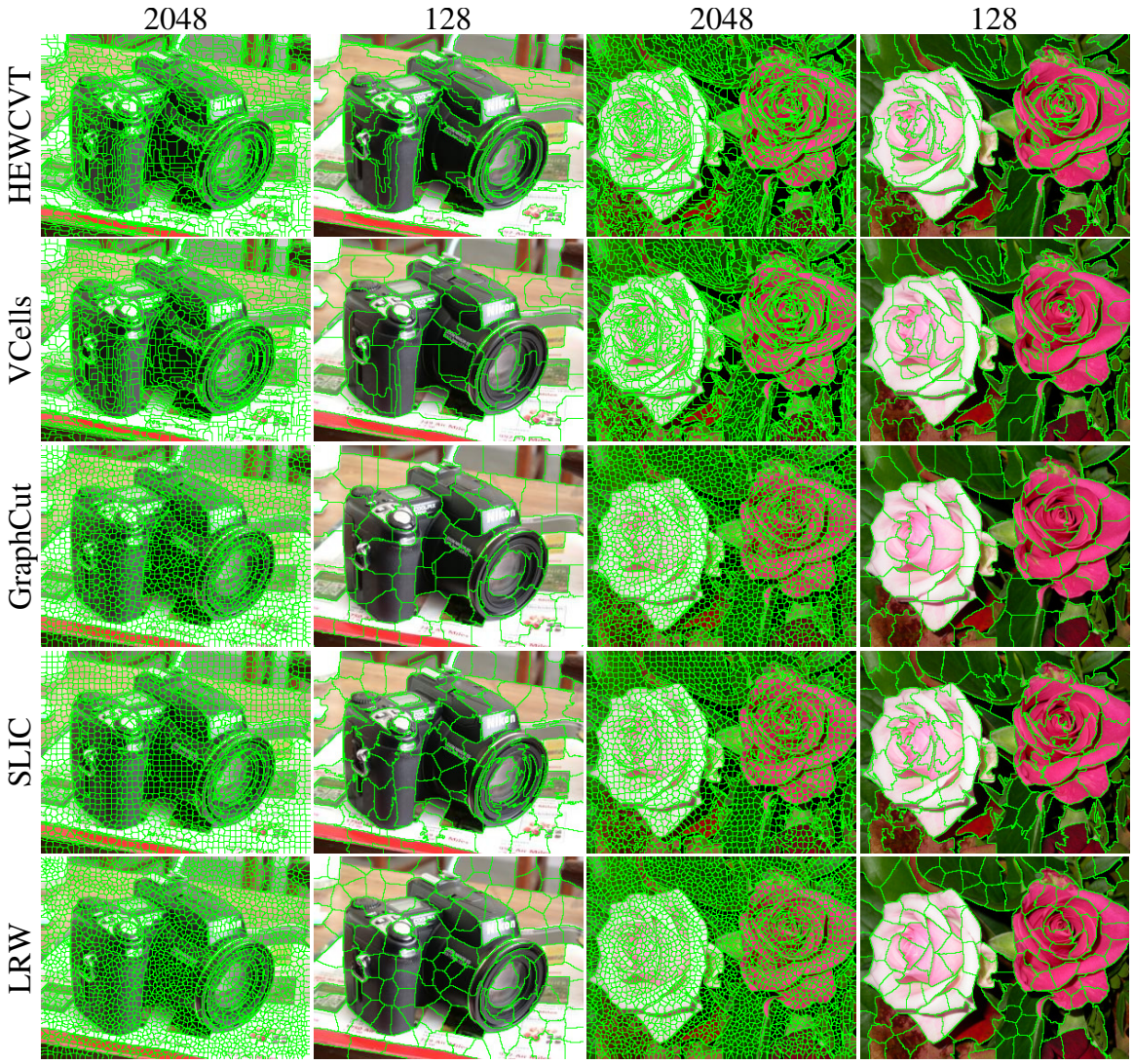


Figure 4.11: Qualitative comparisons of the four superpixel methods (HEWCVT, VCells, GraphCut, SLIC, LRW) on two images from the W1 and W2 datasets. The numbers at the top indicate the desired number of superpixels. Better view in color.

Datasets	# of images	# of training images	# of testing images	# of classes
MSRC[37]	591	296	295	21
VOC2007[17]	632	422	210	20

Figure 4.12: Sample images, human annotated semantic pixel labels, and statistics for the MSRC dataset and the PASCAL VOC2007 dataset.

performance in terms of semantic segmentation accuracy. The semantic segmentation accuracy is defined as the average accuracy for all predefined classes on all testing image pixels.

As listed in Table 4.12, the MSRC image dataset [37] contains 591 color images of size approximately 320×213 and 21 predefined labels. The PASCAL VOC2007 image dataset is larger and more complicate than the MSRC dataset, which contains 632 color images of size approximately 500×375 and 20 predefined labels. For the experiment setting, we fixed the number of desired superpixels as 512 and 2048 for the MSRC dataset and the VOC2007 dataset respectively.

The class-averaged semantic segmentation accuracies for two datasets using superpixels constructed from different algorithms are listed in Table 4.3, and visualized

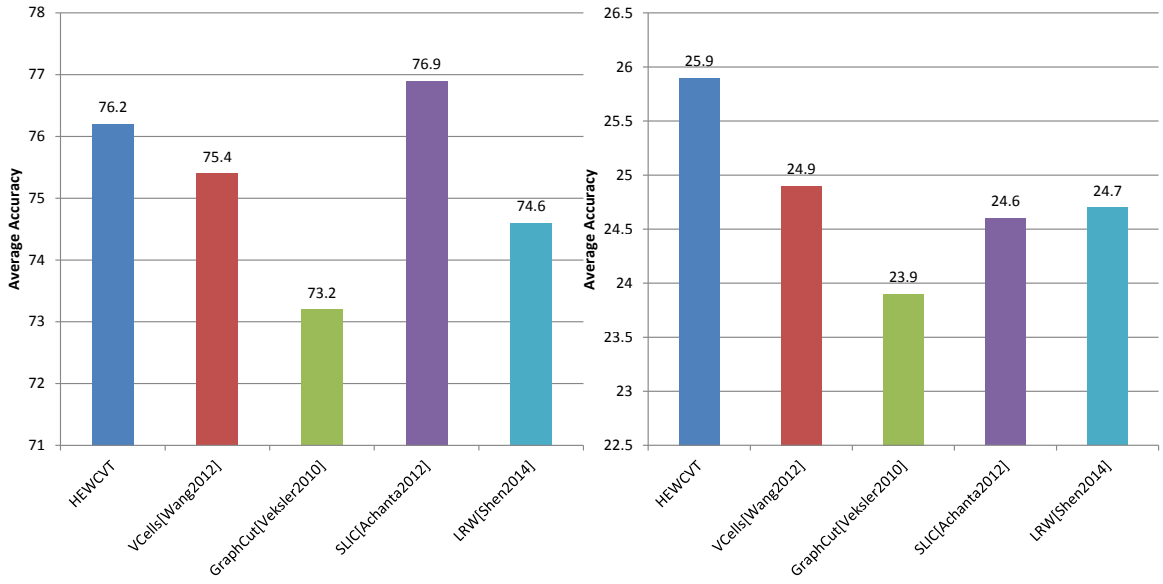


Figure 4.13: Illustrations of quantitative evaluation on the semantic image segmentation task using superpixels constructed by different algorithms. Results for the MSRC dataset are shown in the left, and results for the PASCAL VOC2007 dataset are shown in the right.

in Figure 4.13. From which we can see that, for the MSRC dataset the proposed HEWCVT method achieves comparable class-average segmentation accuracy as the state-of-the-art method SLIC, and outperforms other four superpixel methods; for the complicated PASCAL VOC2007 dataset, the proposed HEWCVT method outperforms all comparison methods.

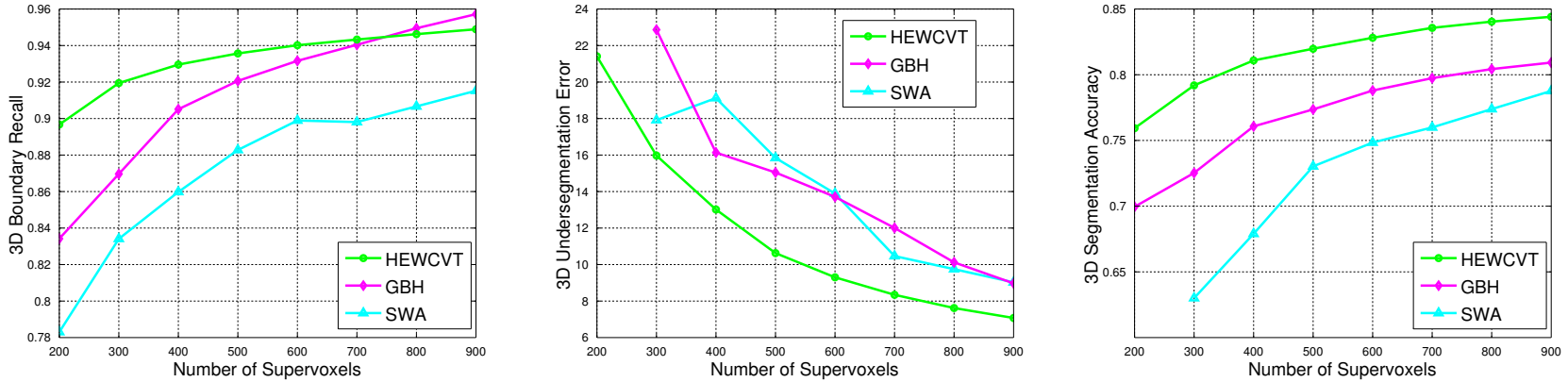


Figure 4.14: Supervoxel evaluation (w/ connectivity enforcement) of GBH, SWA, and HEWCVT on the Xiph.org dataset.

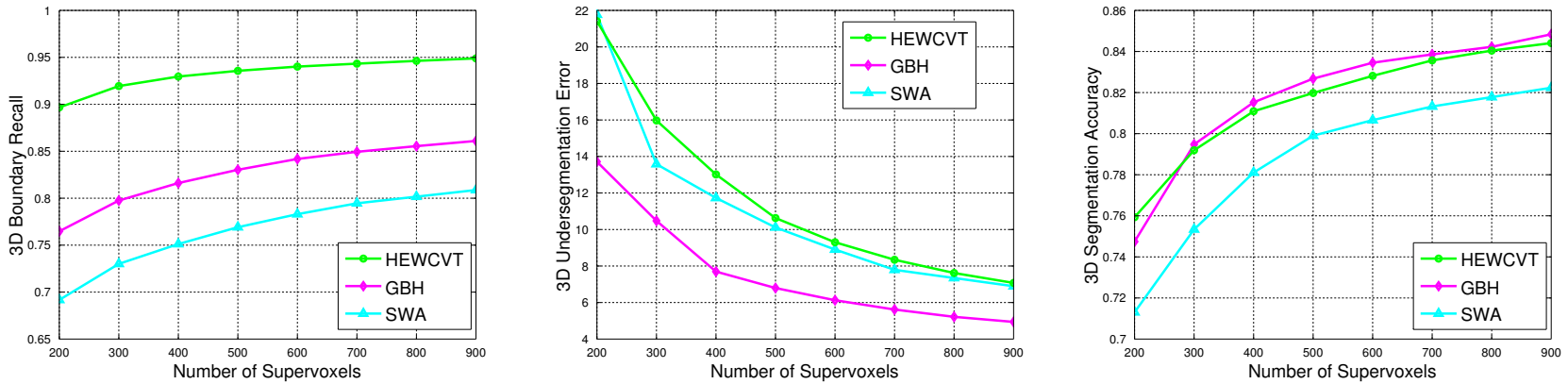


Figure 4.15: Supervoxel evaluation (w/o connectivity enforcement) of GBH, SWA, and HEWCVT on the Xiph.org dataset.

Supervoxel Evaluation

Similar to the superpixel evaluation, we compare the supervoxel construction performance of the proposed HEWCVT against two state-of-the-art supervoxel algorithms: the graph based hierarchical algorithm (GBH) and the weighted aggregation algorithm (SWA). All three algorithms consider a video as an entire 3D volume. We quantitatively and qualitatively evaluate these supervoxel algorithms on the Xiph.org video dataset. However, two comparison algorithms, GBH and SWA, do not enforce the connectivity of each supervoxel, which results in supervoxel fragments in the 3D space. For a fairer comparison, we apply the same connectivity enforcement ($\epsilon = 15$) to remove such fragments and then count each connected component as a separate supervoxel in evaluating GBH and SWA in this paper. Later we will still present the evaluation results without applying the connectivity enforcement and discuss the supervoxel fragment problem.

Quantitative Results

Quantitative results on the Xiph.org video dataset are shown in Figure 4.14. In terms of 3D boundary recall, proposed HEWCVT achieves comparable performance to GBH and better performance than SWA. When the number of supervoxels is very large, supervoxels generated by GBH become highly scattered with a large number of disconnected supervoxel fragments. Therefore GBH achieves better boundary recall. However, highly scattered supervoxels lead to lower accuracy in catching structural boundaries, which is measured by other two metrics. For the other two metrics, 3D undersegmentation error and 3D segmentation accuracy, HEWCVT clearly performs better than both GBH and SWA.

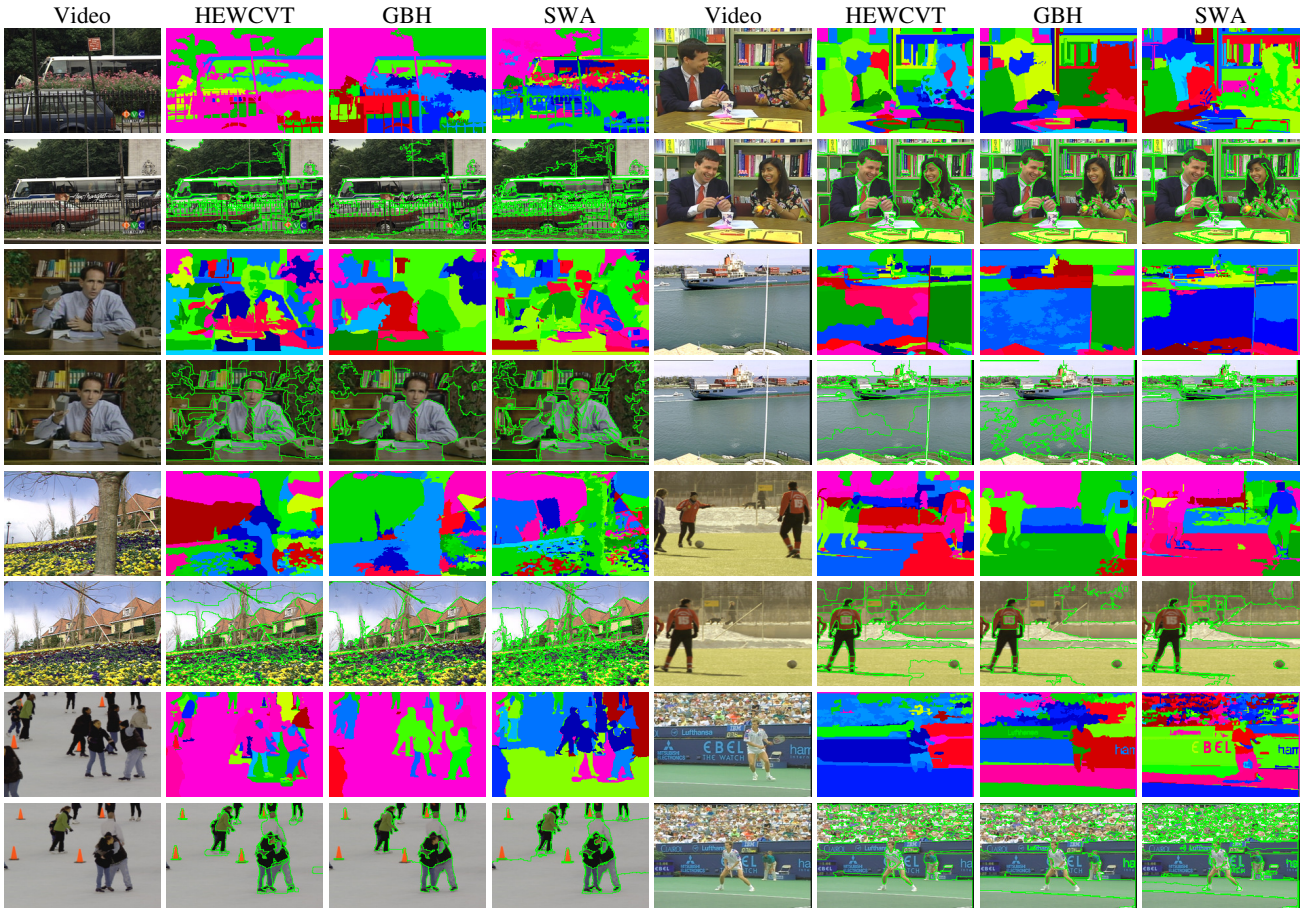


Figure 4.16: Qualitative comparisons of the three supervoxel methods (HEWCVT, GBH, SWA) on four videos. On each video, the number of supervoxels generated by these three methods are similar for fairer comparison. Neighboring supervoxels are shown in different color.

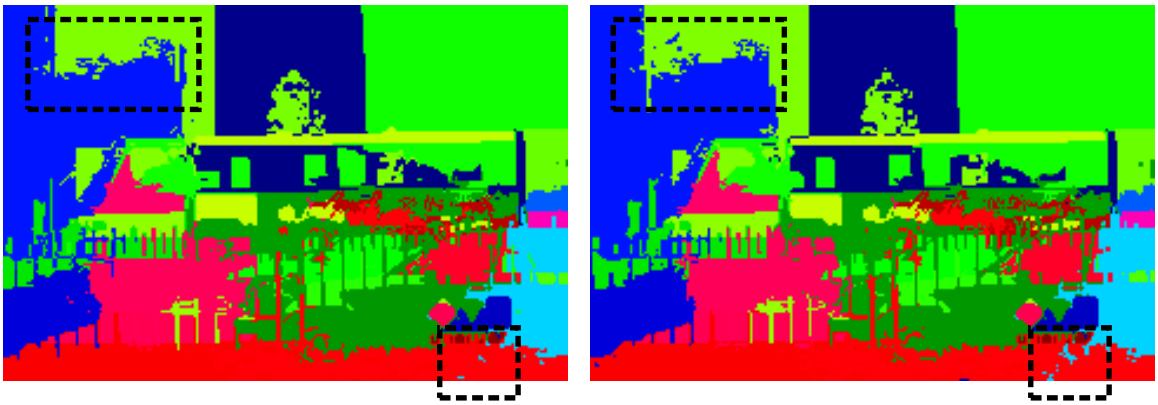


Figure 4.17: An illustration of the dis-connectivity issue in GBH. Constructed supervoxels in two adjacent video frames are visualized with specific colors. Each supervoxel from GBH actually contains many disjoint fragments. Highlighted by black bounding boxes. Better view in color.

Qualitative Results

Qualitative results of constructed supervoxels from different methods are illustrated in Figure 4.16. We can see that, with a similar number of supervoxels, proposed HEWCVT can produce more uniform supervoxels to catch the structural boundaries, but without generating many small fragments, when compared with GBH and SWA.

Discuss on Connectivity Enforcement

Unlike the proposed HEWCVT method and previous superpixel/supervoxel algorithms, recent supervoxel algorithms, GBH and SWA, do not enforce the simple-connectivity among constructed supervoxels, which leads to many disjoint fragments. An example is shown in Figure 4.17, where GBH generates 35 supervoxels on a video and these 35 supervoxels actually consist of 15226 connected components. Given that the three evaluation metrics are dependent on the number of supervoxels, it is clearly unfair and inaccurate to count only 35 supervoxels when evaluating the results in Figure 4.17. Therefore, in the previous evaluation, for GBH and SWA, we apply the same connectivity enforcement ($\epsilon = 15$) as for the proposed HEWCVT to merge such fragments, thus the performance curves of GBH and SWA reported in Figure 4.14

are different from those reported in [51]. Specifically, as described in Section 4.2, given small and/or isolated supervoxels constructed by GBH and SWA, we merge its voxels into neighboring supervoxels based on coordinate distances between voxel and supervoxel centers.

Here in Figure 4.15, we also report the evaluation results without applying the simple-connectivity enforcement for both three supervoxel algorithms, which is the same setting as in [51, 50]. By comparing them with the results illustrated previously in Figure 4.14 (with connectivity enforcement), we can see that, after applying the connectivity filtering:

- The performance of the proposed HEWCVT does not change too much, which indicates that the proposed multiscale supervoxel clustering process already preserves very well the simple-connectivity property of constructed supervoxels. However the performance of other two methods varies a lot, which is caused by merging fragmenting supervoxels into their neighbors.

- Since the voxel based connectivity enforcement produces more boundaries, the 3D boundary recall of GBH and SWA increases after the filtering. However, after merging isolated supervoxels, the area of supervoxels that leak across the boundary of the ground-truth segments increases, and the number of supervoxels that have large portion overlapping with ground-truth segments decreases, thus the undersegment error increases and the segment accuracy decreases.

Based on above observations, we can conclude that the proposed HEWCVT supervoxel method is able to achieve better performance than GBH and SWA, and meanwhile it can also preserve the simple-connectivity property as much as possible, which is important for 3D image segmentation.

4.6 Discussion

We have proposed a hierarchical edge-weighted centroidal Voronoi tessellation method for generating multiscale superpixels/supervoxels. In the finest scale, superpixels/ supervoxels are constructed directly from pixels/voxels. In the higher scales, larger size superpixels/supervoxels are obtained by clustering superpixels/supervoxels in the lower levels. The clustering energy involves both the color feature similarity and the boundary smoothness of superpixels/supervoxels. The obtained structural boundaries are consistent among superpixels/supervoxels in different scales. We have investigated the performance of the proposed method under different parameter settings, and discussed the principles of determining parameters. Quantitative and qualitative results from various experiments show that the HEWCVT method can achieve superior or comparable performances over several current state-of-the-art algorithms. In the future, we will further investigate utilization of motion based features in the clustering energy function for supervoxel construction and also consider extending the proposed method to handle streaming videos.

Chapter 5

Conclusion

In this work, we propose two propagated image segmentation methods for effective segmenting 3D material images and natural images/videos respectively, i.e., the *Edge-Weighted Centroid Voronoi Tessellation with Propagation of Consistency Constraint* (CCEWCVT) and the *Hierarchical Edge-Weighted Centroidal Voronoi Tessellation* (HEWCVT). The CCEWCVT is able to efficiently obtain segmentations on a sequence of 2D serial-sectioned images of 3D material samples by propagating a 2D segmentation from slice to slice, i.e., the inter-image propagation. Experiments conducted on a real high-resolution 3D grain image dataset indicate that, compared with several 2D, 3D and propagated algorithms, the proposed method achieves the best performance in terms of both segmentation accuracy and time efficiency. The HEWCVT, on the other hand, is proposed to capture object boundaries using superpixels/supervoxels in different scales on natural images/videos. Superpixels/supervoxels are constructed iteratively by propagating segmentations on the finest scale to coarser scales. Both quantitative and qualitative evaluation results on several standard datasets show that the proposed HEWCVT method achieves superior or comparable performances to other state-of-the-art algorithms.

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, *SLIC superpixels compared to state-of-the-art superpixel methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **34** (2012), 2274–2282.

[2] Sharon Alpert, Meirav Galun, Ronen Basri, and Achi Brandt, *Image segmentation by probabilistic bottom-up aggregation and cue integration*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, *Contour detection and hierarchical image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **33** (2011), 898–916.

[4] Keni Bernardin and Rainer Stiefelhagen, *Evaluating multiple object tracking performance: The clear mot metrics*, EURASIP Journal on Image and Video Processing **2008** (2008), 1–10.

[5] Yu Cao, Lili Ju, and Song Wang, *Grain segmentation of 3d superalloy images using multichannel ewcvt under human annotation constraints*, European Conference on Computer Vision (ECCV), 2012.

[6] Yu Cao, Lili Ju, Youjie Zhou, and Song Wang, *3D superalloy grain segmentation using a multichannel edge-weighted centroidal voronoi tessellation algorithm*, IEEE Transactions on Image Processing (TIP) **22** (2013), 4123–4135.

[7] Yu Cao, Lili Ju, Qin Zou, Chengzhang Qu, and Song Wang, *A multichannel edge-weighted centroidal voronoi tessellation algorithm for 3d super-alloy image segmentation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[8] A. Chen and J. Corso, *Propagating multi-class pixel labels throughout video frames*, Proceedings of Western New York Image Processing Workshop, 2010.

[9] Haili Chui and Anand Rangarajan, *A new point matching algorithm for non-rigid registration*, Computer Vision and Image Understanding (CVIU) **89** (2003), 114–141.

[10] Dorin Comaniciu and Peter Meer, *Mean shift: A robust approach toward feature space analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **24** (2002), 603–619.

[11] SvenJ. Dickinson, Alex Levinshtein, and Cristian Sminchisescu, *Perceptual grouping using superpixels*, Pattern Recognition **7329** (2012), 13–22.

[12] Caglayan Dicle, Octavia I Camps, and Mario Sznaier, *The way they move: Tracking multiple targets with similar appearance*, IEEE International Conference on Computer Vision (ICCV), 2013, pp. 2304–2311.

[13] Michael Donoser, Martin Urschler, Hayko Riemenschneider, and Horst Bischof, *Highly consistent sequential segmentation*, Proceedings of Scandinavian conference on Image analysis, 2011.

[14] Qiang Du, Vance Faber, and Max Gunzburger, *Centroidal voronoi tessellations: Applications and algorithms*, SIAM Review **41** (1999), 637–676.

[15] Qiang Du, Max Gunzburger, and Lili Ju, *Advances in studies and applications of centroidal voronoi tessellations*, Numerical Mathematics: Theory, Methods and Applications **3** (2010), 119–142.

[16] Alexei A. Efros and Thomas K. Leung, *Texture synthesis by non-parametric sampling*, IEEE International Conference on Computer Vision (ICCV), 1999.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The pascal visual object classes (voc) challenge*, International Journal of Computer Vision (IJCV) **88** (2010), 303–338.

[18] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, *Efficient graph-based image segmentation*, International Journal of Computer Vision (IJCV) **59** (2004), 167–181.

[19] Brian Fulkerson, A. Vedaldi, and S. Soatto, *Class segmentation and object localization with superpixel neighborhoods*, IEEE International Conference on Computer Vision (ICCV), 2009.

[20] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller, *Multi-class segmentation with relative location prior*, International Journal of Computer Vision (IJCV) **80** (2008), 300–316.

[21] Michael Groeber and Michael Jackson, *Dream.3d: A digital representation environment for the analysis of microstructure in 3d*, Integrating Materials and Manufacturing Innovation **3** (2014), 5.

[22] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa, *Efficient hierarchical graph-based video segmentation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[23] Xuming He, Richard S. Zemel, and Debajyoti Ray, *Learning and incorporating top-down cues in image segmentation*, European Conference on Computer Vision (ECCV), 2006.

[24] George Karypis and Vipin Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing (SISC) **20** (1998), 359–392.

[25] Pushmeet Kohli, L'Ubor Ladický, and Philip H. Torr, *Robust higher order potentials for enforcing label consistency*, International Journal of Computer Vision (IJCV) **82** (2009), 302–324.

[26] H. W. Kuhn and Bryn Yaw, *The hungarian method for the assignment problem*, Naval Research Logistics Quarterly **2** (1955), 83–97.

[27] A. Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, and K. Siddiqi, *Turbopixels: Fast superpixels using geometric flows*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **31** (2009), 2290–2297.

[28] Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang, *Segmentation using superpixels: A bipartite graph partitioning approach*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[29] K Magnusson, J Jalden, P Gilbert, and H Blau, *Global linking of cell tracks using the viterbi algorithm*, IEEE Transactions on Medical Imaging **34** (2014), 911–929.

[30] 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*, IEEE International Conference on Computer Vision (ICCV), 2001.

[31] Fernand Meyer, *Topographic distance and watershed lines*, Signal Processing **38** (1994), 113–125.

[32] Anton Milan, Stefan Roth, and Konrad Schindler, *Continuous energy minimization for multitarget tracking*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **36** (2014), no. 1, 58–72.

[33] Richard Nock and Frank Nielsen, *Statistical region merging*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **26** (2004), 1452–1458.

[34] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes, *Globally-optimal greedy algorithms for tracking a variable number of objects*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1201–1208.

[35] Roger C. Reed, *The superalloys fundamentals and applications*, Cambridge Press, 2001.

[36] Xiaofeng Ren and Jitendra Malik, *Learning a classification model for segmentation*, IEEE International Conference on Computer Vision (ICCV), 2003.

[37] Eitan Sharon, Achi Brandt, and Ronen Basri, *Fast multiscale image segmentation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2000.

[38] Jianbing Shen, Yunfan Du, Wenguan Wang, and Xuelong Li, *Lazy random walks for superpixel segmentation*, IEEE Transactions on Image Processing (TIP) **23** (2014), 1451–1462.

[39] Jianbo Shi and Jitendra Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **22** (2000), 888–905.

[40] Purdy Vander Voort, Warmuth and Szirmae, *Metallography: Past, present, and future, 75th anniversary volume*, ASTM, 1993.

[41] Amelio Vazquez-Reina, Shai Avidan, Hanspeter Pfister, and Eric Miller, *Multiple hypothesis video segmentation from superpixel flows*, European Conference on Computer Vision (ECCV), 2010.

[42] A. Vedaldi and S. Soatto, *Quick shift and kernel methods for mode seeking*, European Conference on Computer Vision (ECCV), 2008.

[43] Olga Veksler, Yuri Boykov, and Paria Mehrani, *Superpixels and supervoxels in an energy optimization framework*, European Conference on Computer Vision (ECCV), 2010.

[44] Jie Wang, Lili Ju, and Xiaoqiang Wang, *An edge-weighted centroidal voronoi tessellation model for image segmentation*, IEEE Transactions on Image Processing (TIP) **18** (2009), 1844 –1858.

[45] Jie Wang and Xiaoqiang Wang, *Vcells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **34** (2012), no. 6, 1241–1247.

[46] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang, *Superpixel tracking*, IEEE International Conference on Computer Vision (ICCV), 2011.

[47] Greg Welch and Gary Bishop, *An introduction to the kalman filter*, Tech. report, University of North Carolina at Chapel Hill, 1995.

[48] Ross T. Whitaker, *A level-set approach to 3d reconstruction from range data*, International Journal of Computer Vision (IJCV) **29** (1998), 203–231.

[49] David G. Rethwisch William D. Callister, *Materials science and engineering: An introduction*, Wiley, 2010.

[50] Chenliang Xu and Jason J. Corso, *Evaluation of super-voxel methods for early video processing*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[51] Chenliang Xu, Caiming Xiong, and Jason J. Corso, *Streaming hierarchical video segmentation*, European Conference on Computer Vision (ECCV), 2012.

[52] Yi Yang, S. Hallman, D. Ramanan, and C. Fowlkes, *Layered object detection for multi-class segmentation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

Related Publications

[1] **Youjie Zhou**, Lili Ju, and Song Wang. Multiscale superpixels and supervoxels based on hierarchical edge-weighted centroidal voronoi tessellation. *IEEE Transactions on Image Processing (TIP)*, 2015.

[2] Jarrell Waggoner, **Youjie Zhou**, Jeff Simmons, Marc De Graef, and Song Wang. Graph-cut based interactive segmentation of 3D materials-science images. *Machine Vision and Applications*, 25(6):1615–1629, 2014.

[3] Jarrell Waggoner, **Youjie Zhou**, Jeff Simmons, Marc De Graef, and Song Wang. 3D materials image segmentation by 2D propagation: A graph-cut approach considering homomorphism. *IEEE Transactions on Image Processing (TIP)*, 22(12):5282–5293, 2013.

[4] Yu Cao, Lili Ju, **Youjie Zhou**, and Song Wang. 3D superalloy grain segmentation using a multichannel edge-weighted centroidal voronoi tessellation algorithm. *IEEE Transactions on Image Processing (TIP)*, 22(10):4123–4135, 2013.

[5] **Youjie Zhou**, Lili Ju, and Song Wang. Multiscale superpixels and supervoxels based on hierarchical edge-weighted centroidal voronoi tessellation. In *Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[6] **Youjie Zhou**, Hongkai Yu, Jeff Simmons, Yuewei Lin, Yang Mi, and Song Wang. Large-scale fiber tracking from microscopic composite image. In *Joint NSRC Workshop: Big, Deep, and Smart Data Analytics in Materials Imaging*, 2015.

[7] **Youjie Zhou**, Hongkai Yu, and Song Wang. Feature sampling strategies for action recognition. *CoRR*, 2015.

[8] **Youjie Zhou**, Lili Ju, Yu Cao, Jarrell Waggoner, Yuewei Lin, Jeff Simmons, and Song Wang. Edge-weighted centroid voronoi tessellation with propagation of consistency constraint for 3D grain segmentation in microscopic superalloy images. In *CVPR Workshop on Perception Beyond the Visible Spectrum (PBVS)*, 2014.

[9] Jarrell Waggoner, **Youjie Zhou**, Jeff Simmons, Marc De Graef, and Song Wang. Topology-preserving multi-label image segmentation. In *Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[10] Hongkai Yu, **Youjie Zhou**, Hui Qian, Min Xian, Yuewei Lin, Dazhou Guo, Kang Zheng, Kareem Abdelfatah, and Song Wang. Loosecut: Interactive image segmentation with loosely bounded boxes. *CoRR*, 2015.

[11] Jarrell Waggoner, **Youjie Zhou**, Jeff Simmons, Ayman Salem, Marc De Graef, and Song Wang. Interactive grain image segmentation using graph cut algorithms. In *Proceedings of SPIE (Computational Imaging XI)*, 2013.

[12] Dhaval Salvi, Kang Zheng, **Youjie Zhou**, and Song Wang. Distance transform based active contour approach for document image rectification. In *Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[13] Xiaochuan Fan, Kang Zheng, **Youjie Zhou**, and Song Wang. Pose locality constrained representation for 3D human pose reconstruction. In *European Conference on Computer Vision (ECCV)*, 2014.

[14] Kang Zheng, Yuewei Lin, **Youjie Zhou**, Dhaval Salvi, Xiaochuan Fan, Dazhou Guo, Zibo Meng, and Song Wang. Action recognition in videos of multiple wearable cameras. In *ECCV Workshop on Looking at People (ChaLearn)*, 2014.

[15] Yuewei Lin, Jing Chen, Yu Cao, **Youjie Zhou**, Lingfeng Zhang, and Song Wang. Cross-domain recognition by identifying compact joint subspaces. In *International Conference on Image Processing (ICIP)*, 2015.