1-1-2013

Multi-Label Segmentation Propagation for Materials Science Images Incorporating Topology and Interactivity

Jarrell Waggoner
University of South Carolina

Multi-Label Segmentation Propagation for Materials Science Images Incorporating Topology and Interactivity

By

Jarrell Waggoner

Associate of Science
University of South Carolina at Lancaster 2004

Bachelor of Science
Bryan College 2006

Master of Engineering
University of South Carolina 2009

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosphy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2013

Accepted by:

Song Wang, Major Professor

Michael N. Huhns, Committee Member

Manton M. Matthews, Committee Member

Yan Tong, Committee Member

Andrew B. Greytak, External Examiner

Lacy Ford, Vice Provost and Dean of Graduate Studies

Dedication

To my family.

Acknowledgments

There are more contributors to this dissertation than I can name. I would particularly like to thank my advisor, Dr. Song Wang, for guiding this work and introducing me to the wide world of scholarly research. I'd also like to thank my committee, Dr. Huhns, Dr. Matthews, Dr. Tong, and Dr. Greytak, whose questions and feedback have helped shape this manuscript. I'm thankful for the support of other research collaborators, including Dr. Jeff Simmons, and Dr. Marc De Graef.

Many colleagues, labmates, and friends have made working on this dissertation both easier and more fun. A special thanks to Dr. Brent Munsell, Dr. Pahal Dalal, Dr. Andrew Temlyakov, Dr. Yu Cao, Dr. Zhiqi Zhang, Dhaval Salvi, and Youjie Zhou, who have made the computer vision lab a great place to do research. Of particular note is Todd Lewis who generously provided numerous opportunities to grow professionally while completing my degree.

I would like to thank my earliest mentors, including Dr. Noni Bohonak, Dr. Danny Faulkner, and Earl Reed, without whose encouragement and continued belief that I could succeed, I would never have begun this journey.

Last and most importantly, I'd like to thank my parents and siblings for their support and understanding with the long hours and endless work, Charis for walking alongside me in this journey, and Sage for keeping us entertained.

Abstract

Segmentation propagation is the problem of transferring the segmentation of an image to a neighboring image in a sequence. This problem is of particular importance to materials science, where the accurate segmentation of a series of 2D serial-sectioned images of multiple, contiguous 3D structures has important applications. Such structures may have prior-known shape, appearance, and/or topology among the underlying structures which can be considered to improve segmentation accuracy. For example, some materials images may have structures with a specific shape or appearance in each serial section slice, which only changes minimally from slice to slice; and some materials may exhibit specific topology which constrains their structure or neighboring relations.

In this work, we develop a framework for materials image segmentation that segments a variety of materials image types by repeatedly propagating a 2D segmentation from one slice to another, and we formulate each step of this propagation as an optimal labeling problem that can be efficiently solved using the graph-cut algorithm. During this propagation, we propose novel strategies to enforce the shape, appearance, and topology of the segmented structures, as well as handling local topology inconsistency. Most recent works on topology-constrained image segmentation focus on binary segmentation, where the topology is often described by the connectivity of both foreground and background. We develop a new multi-labeling approach to enforce topology in multiple-label image segmentation. In this case, we not only require each segment to be a connected region (*intra-segment topology*), but also require specific adjacency relations between each pair of segments (*inter-segment*

topology). Finally, we integrate an interactive approach into the proposed framework that improves the segmentation by allowing minimal and simplistic human annotations. We justify the effectiveness of the proposed framework by testing it on various 3D materials images, and we compare its performance against several existing image segmentation methods.

Contents

List of Tables

List of Figures

Introduction

Materials science is a wide and varied field of great importance to numerous civil and military applications. Of particular interest to the materials field is the detailed microscopic structures of a material [108, 86, 14, 85, 62, 5, 50]. Such structures may be "grains" in a metal, cells in biomaterial, etc., and can be imaged optically, tomographically, or with electron microscopy. The composition and relationships (microstructure) among these structures are strong determinants of a material's electrical, thermal, chemical, explosive, organic, diffusive, structural, and manufacturing properties [84, 110], and thus rapid and accurate localization (i.e., segmentation) and analysis of structures is of great benefit to a wide range of materials applications. Modern materials analysis often involves procedures such as finite element modeling to infer specific properties from a material's structure [42]. While such analysis is sometimes performed with simulated inputs [39], it is preferable to use a real 3D volume made up of a sequence of 2D images, extracted tomographically [41] or via serial-sectioning [125] for this purpose [115]. In this work, we focus on addressing the problem of accurately segmenting the underlying structures of various 3D materials science images.

With many desirable mechanical, electrical, thermal, chemical, and manufacturing properties [84], metallic materials are specifically important. These aforementioned properties are strongly dependent on the detailed microscopic structures of the material [108, 86, 14, 85, 62, 5, 50]. In particular, most metallic materials consist of a large number of microscopic crystals, or "grains," of random orientation with respect to one another. The size and the shape of these crystals are among the strongest

determinants of many material properties such as mechanical strength or fracture resistance.

Furthermore, the area of biomaterials is also a wildly diverse, and important field [8]. Biomaterials are typically made up of many varied cells, whose states determine different properties of the biomaterial [2], with application to vaccine development [59, 83], tissue reconstruction [46, 92, 81], stem cell development [102, 73], and plant structure and vascular analysis [112, 113, 7, 126].

While some materials properties can be derived from a single image, it is common to obtain multiple images from material volumes [45]. These images can be extracted at varying angles or depths using a tomographic [41] approach. It is also possible, however, to create a serial-sectioned volume by physically slicing a material and imaging it at increasingly varying depths [125], which is useful for materials that are not amenable to wave-based tomographic approaches. Since tomographic sequences are generally reconstructed using specialized algorithms [53] which aggregate the source signal, we focus on serial-sectioned volumes, imaged in various ways (optically, using electron microscopy, etc.), rather than tomographically reconstructed volumes.

Materials science image segmentation of serial-sectioned volumes is a highly challenging problem, for several reasons. First, many materials images are large and high resolution, which increases demand for an efficient segmentation algorithm. Second, materials images usually contain various kinds noise, blurring and ambiguities, as shown by the two etched [77] serial-section images of a titanium sample in Figure 0.1 (a) and (b), where some grain boundaries may be less distinct than other boundaries (e.g., g_1). In addition, some undesired scratches or other noise inside the grains may exhibit high intensity similar to grain boundaries (e.g., g_2). Such image noise and ambiguity makes it difficult to accurately extract all the segment boundaries by edge detection [15, 64] or intensity thresholding [36, 96, 94]. Furthermore, chemical processing, together with lighting changes during the microscopic imaging,

often leads to inconsistent image intensity across slices. For example, grain g_3 exhibits different intensities between the two slices shown in Figure 0.1 (a-b). This makes it difficult to directly segment the entire image volume directly [9, 124, 72] instead of segmenting the constituent slices independently. Different imaging types can introduce distinct ambiguities as well. For example, the PSF of a confocal microscope can introduce z-axis blurring. Some materials have structures that are directly adjacent to other structures (e.g., Figure 0.1 (a-b)) and preserving these adjacencies is critical to a suitable segmentation, while others are made of structures that are largely *not* adjacent to each other, e.g., Figure 0.1 (c), and capturing the gaps between structures, as shown in Figure 0.1 (d), is as important as identifying the exact structure boundaries themselves. Unetched materials, as shown in Figure 0.1 (e), exhibit further challenges. Indistinct boundaries, and noise that is the same size and appearance as salient structures, compound the ability of segmentation algorithms to obtain good accuracy without human intervention. Often, a human segmenter will manually examine a volume and choose only a subset of the visible structures to segment, as shown in Figure 0.1 (f). Incorporating this prior knowledge, rather than attempting to segment every possible structure (an oftentimes impossible task for human or machine) is a difficult but important design characteristic for materials segmentation algorithms. Complicating this further is the possibility of structures exhibiting multiple phases as shown in Figure 0.1 (g), multiple modalities for each image (e.g. images from various angles), various imaging types, and differing inter- and intra-slice resolutions, all of which make segmenting 3D materials volume a difficult, non-trivial challenge. Third, the segments of many material samples are highly complex. For example, a small metal sample may consist of hundreds of grains with different sizes and shapes, which must be accurately segmented in every slice. Fourth, a typical way to address various image complications is to incorporate available domain knowledge on the shape, appearance and topology of the material segments.

However, in practice, it can be a very challenging problem to model and incorporate such prior knowledge, which is largely unexplored in materials segmentation. Finally, given the variety of materials and materials imaging techniques, we may obtain large variation among different materials images. One key challenge is to develop a generalizable approach that can be easily adapted to accurately segment different materials images.

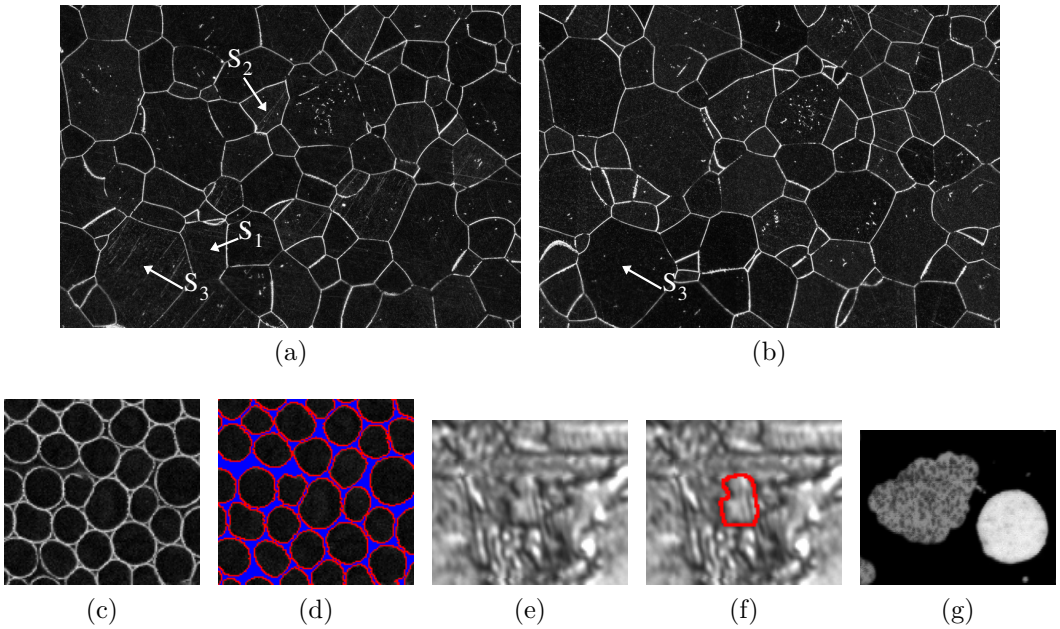


Figure 0.1: An illustration of the challenges in materials image segmentation. **(a-b)** Two consecutive slices of a titanium material, where grain g_1 and g_2 exhibit indistinct boundary segments and undesired image noise respectively, and grain g_3 exhibits inconsistent intensity between these two slices. **(c-d)** 2D slice from a cotylendon biomaterial and its segmentation, where the space between cells is important to analyzing the material properties. **(e-f)** 2D slice of a noisy steel material, and a martensite lath structure of interest. **(g)** 2D slice of NanoEnergetic Epoxy, where two structures have different phases that manifest as different intensity distributions.

Segmentation Propagation

In Chapter 1, we present a graph-cut based approach for materials image segmentation where we segment a sequence of images by repeatedly propagating a given 2D segmentation from one slice to another. Different from most existing tracking

methods, our propagation approach focuses on the ability to track a large number of segments through a 3D materials image. To do this propagation, we enforce *homomorphism* during the propagation, i.e., identifying the same set of objects in all propagated images. Additionally, we show how prior properties of the desirable segments—namely shape, intensity, and non-homomorphism—can be incorporated into this approach to segment different materials images, which illustrates the versatility and broad applicability of the proposed approach. We formulate this propagation approach as an optimal labeling that can be efficiently solved using the graph-cut algorithm. To maintain general consistency across slices, we first run a homomorphic "global" labeling to produce an initial segmentation on the new slice. We then run a series of local operations to refine the segmentation by identifying and correcting possible 2D topology inconsistencies, which allows the proposed method to handle non-homomorphism.

Multi-Label Topology

Topology is among the most important priors for many applications. In medical imaging [68, 109], different organs or tissues in the human body may show specific contextual relations. In materials science [89, 16], polycrystalline materials are usually made up of numerous grains with specific adjacency relations. Enforcing topology in image segmentation can substantially improve the segmentation accuracy and reliability [19, 28, 40, 60, 127, 49, 76, 120, 128, 75].

However, enforcing prior topology in image segmentation is a very challenging problem because topology is not a simple local property. In particular, advanced image segmentation models that can better handle large-scale images in real applications usually involve various boundary and region properties, and have complex cost functions. Further enforcing a topology constraint may make it infeasible to find an efficient algorithm to optimize the segmentation cost functions.

One of the most widely used image segmentation models is the Markov Random Field (MRF)[55], where the cost function contains a unary term for each individual pixel and a binary term for pixel pairs. In particular, many efficient algorithms, such as graph cut, have been used to solve the MRF optimization problem and derive an image segmentation. Enforcing topology in an MRF-based image segmentation has attracted much attention in recent years [120, 127, 76, 19]. However, even the state-of-the-art work has only focused on the simple case of foreground/background (binary) segmentation, as shown in Figure 0.2 (a). In this special case, the topology of such a segmentation is defined solely in terms of the connectivity of the foreground and/or background segments. We refer to such topology as *intra-segment topology*.

In practice, however, most applications require the segmentation of multiple structures of interest. The goal of topology preservation in this general case is to obtain multiple segments (> 2) such that each segment is a single connected region (intra-segment topology), and each pair of segments has pre-specified (non-)adjacency relations, i.e., *inter-segment topology*. Enforcing multi-segment topology in MRF-based image segmentation is of substantially greater complexity compared with binary segmentation. Even without considering the segmentation topology, MRF-based multi-labeling segmentation is NP-hard [118] and only locally-optimal solutions can be obtained.

Our goal is to expand our method for global topology propagation to handle topology-enforced multi-label segmentation within the MRF framework. This is very different from the topology-enforced binary segmentation because, in a multi-label segmentation, we not only require the connectivity of each segment (*intra-segment topology*), but also require pre-specified adjacency relations between each pair of resulting segments (*inter-segment topology*), and as such is not a trivial extension from an algorithmic perspective. Taking the Topology Cut algorithm [127] as an example, which can be treated as a form of the α-β swap [1] approach, each step of α-β swap

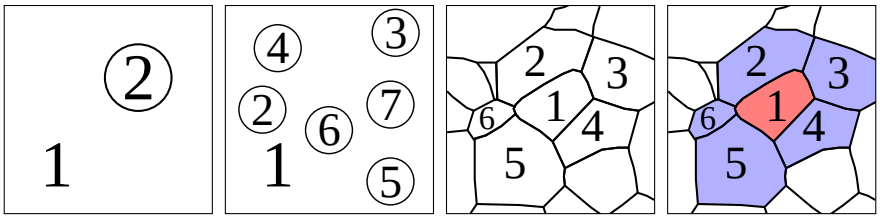


Figure 0.2: Examples of segmentation topology complexity. **(a)** Foreground/background segmentation, which has no inter-segment topology. **(b)** Simple inter-segment topology, showing few adjacency relationships among segments. **(c)** Complex inter-segment topology, with many and varied adjacency relationships between segments. **(d)** Example of a ring structure with segment 1 (red) as the ring center, and its immediate neighbors (blue).

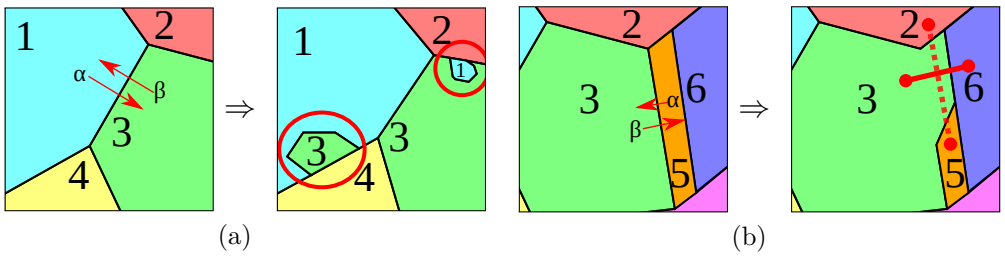


Figure 0.3: Examples of topology types. **(a)** Inter-segment topology (adjacency relations) preserved, but intra-segment topology (connectivity) not preserved due to segments 1 and 3 becoming disjoint regions. **(b)** Intra-segment topology preserved, but inter-segment topology not preserved due to segments 3 and 6 becoming adjacent, and segments 2 and 5 becoming non-adjacent.

will select a pair of segments with different labels and a subset of pixels swap their labels to minimize the MRF cost function. In a binary segmentation, only segment connectivity need be enforced in the α-β swap since there are only two labels and the resulting two segments are always adjacent. However, for more than two labels, enforcing segment connectivity in each α-β swap is not sufficient, because such a swap may change adjacency relations among other segments, as illustrated in Figure 0.3. α-expansion, another widely used algorithm for multi-label segmentation, exhibits the same problem.

In Chapter 2, we develop a new method for enforcing topology in multi-label segmentation within the MRF framework. In practice, multi-segment topology may

show different levels of complexity. In the most simple case, there is a single background segment, and all other segments neighbor this background segment, as shown in Figure 0.2 (b). In the work discussed in Chapter 2, we focus on one of the most complex multi-segment topologies, as shown in Figure 0.2 (c), where each segment has multiple neighboring segments. An example in a real materials image is shown in Figure 0.4.

To make this challenging problem tractable, we introduce one simplification. To segment a target image, we assume a similar segmented template image is available which both defines the topology of the segmentation that we wish to preserve, and provides an initialization for the segmentation on the target. This simplification is reasonable for a number of applications, including 3D volume segmentation by propagating a 2D segmentation slice-by-slice (the application discussed in Chapter 1), and segmentation propagation from one image to another collected for stereo matching.

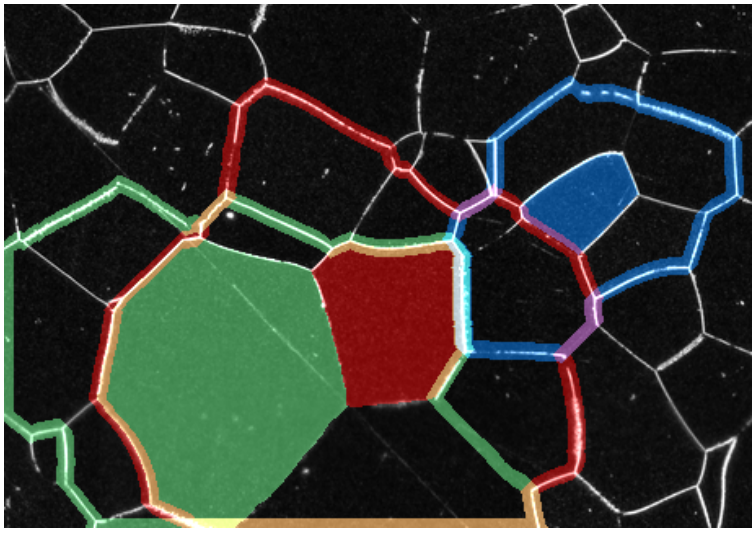


Figure 0.4: Selected grains as ring centers, with ring boundaries outlined in matching colors. Notice that local ring structures may overlap, sharing grains.

To preserve the inter-segment topology, we independently consider each subset of segments that form a ring structure, as illustrated in Figure 0.2 (d), which consists of a center segment and its immediate neighbors. We map these ring structures to

the target image by incorporating the underlying image information into the cost function and conduct an MRF-based segmentation within each ring structure. By encoding non-adjacency constraints in the binary term, we show that the adjacency constraints are implicitly satisfied in a ring structure. To preserve the intra-segment topology, we set the unary term at each pixel to be to either 0 or ∞, determined by the spatial proximity to the segmentation boundaries in the template image. To evaluate the performance of the proposed method, we apply it to segment 3D metallic image volumes slice-by-slice for the underlying grain structures, and quantitatively compare its performance to several existing state-of-the-art methods. Finally, we also apply the proposed method to interactive segmentation and stereo matching applications.

Segmentation Interaction

Interactive segmentation is a rapidly-growing area of computer vision and has seen heightened interest recently[56, 106]. While traditional segmentation seeks to identify objects/structures within an image in a fully-automated fashion, interactive segmentation, similar to active learning [93], accomplishes the goal of image segmentation while incorporating a sparse number of user interactions which are included as additional constraints or guidance in the segmentation model or algorithm. These interactions may take on different forms, and may include drawing a bounding box [87], roughly outlining a boundary [70], or drawing brush strokes inside and/or outside the object of interest [91, 116, 12, 119]. A desired property of an interactive segmentation approach is that the user interaction be as convenient (i.e., low cognitive load) and sparse (i.e., few in number) as possible, while simultaneously providing immediate feedback to the user on every interaction.

In Chapter 3, we present an interactive segmentation approach to segment materials science image volumes. We show that our approach discussed in Chapter 1 can be extended to allow for convenient interactive segmentation. We illustrate the perfor-

mance of the interactive approach by using it to segment a materials image volume using a smaller number of interactions compared with general-purpose interactive segmentation methods that do not incorporate materials-specific priors. Finally, we develop three strategies that increase the effectiveness of the proposed interactive segmentation tools, which include an annotation repropagation method, a parameter estimation technique, and a salient region detection approach, which further reduce the number of user-required interactions in the segmentation process.

Related Work

Many existing image segmentation methods have been—or have the potential to be—used for segmenting 3D materials images. In general, they can grouped into two classes: 3D methods that segment an entire volume directly, and 2D methods that segment individual slices within the volume. After discussing these two related categories, we will review the literature relating to segmentation topology and interactive segmentation.

3D methods are widely used for medical applications [57, 33, 30] and have seen more recent use in materials science also [16]. These methods can be formulated as large optimization problems [9], random walks [37], 3D extensions of methods that are also used in 2D applications [4, 31, 122], clustering problems [16], or learning-based approaches [114]. There are several issues when applying 3D methods for material image segmentation. First, 3D segmentation algorithms are usually computationally expensive given the large number of voxels in a high-resolution 3D material image. Second, many 3D methods require consistency of image information, e.g., intensity, for the same segment over the entire 3D volume. This may not hold for many materials images, such as serial-sectioned microscopy images: a 3D segment may show different intensity from one slice to another due to inconsistent imaging conditions. Third, many 3D methods require uniform resolution (i.e., sampling rate) along all three

axes. This may not hold for many materials images, where the resolution along the serial-section direction is often much lower than the image resolution on each slice. Finally, it is desirable to have some prior knowledge about the segments, which can be in the form of a model, or an initialization that is interactively specified on selected 2D slices, and it is often difficult to specify and incorporate such prior knowledge into a 3D segmentation method.

Because of these drawbacks, 2D methods are more applicable to the materials science field due to the many unique challenges posed by materials images. A huge variety of general 2D segmentation methods (some of which can also work in 3D) are available, including watershed [69] and watershed cuts [26, 28] , region merging [74, 130], mean shift [21], level sets [61, 72], EM-based methods [17], statistical methods [35], anisotropic diffusion [82], normalized cut [97], general graph cut [80, 34], ratio regions [29], ratio cut [123], active contour [54], and many others. Overall, 2D methods are often designed to run faster, require less human interaction, and be robust to noise compared to 3D methods, but do not often directly incorporate intra-slice information. Thus, intra-slice ambiguities might be missed by such methods. To address this problem, tracking based methods have been developed for segmenting a sequence of 2D slices (or frames) by enforcing the segmentation consistency between slices [99, 90, 129]. Such tracking methods strike a balance between 2D and 3D methods, keeping the performance advantages of 2D methods while still incorporating 3D information between slices. However, most existing tracking methods are designed to track a single object, or a small number of disjoint objects and it is nontrivial to apply them to segment materials images which may contain hundreds of segments of interest.

Some of these 2D methods have been applied to segment materials images specifically [20, 100]. Most prominent is the work of Comer et al. [23, 24] on the EM/MPM algorithm, originating from [65] . Other methods that have been specifically used on

materials images include graph cut [44], stabilized inverse diffusion equations [43], Bayesian methods [22, 101], and the watershed [63] method. While each of these methods achieves acceptable performance on a single slice or volume, there is no guarantee of applicability for other types of materials images. Most often, materials images are opportunistically segmented by the simplest tools available, such as thresholding [36, 96], or out-of-the-box methods such as watershed or normalized cut. For some of these 2D segmentation methods, such as watershed, 2D topology consistency can be imposed across slices to achieve more consistent image sequence segmentation [18]. In Chapter 1, we conduct experiments to compare the performance of the proposed method with the normalized cut and watershed methods.

There has also been important research progress on integrating prior models in segmentation methods, both 2D and 3D, which is of particular relevance to materials volumes, which contain structures of constrained size, shape, and composition. A selection of contour-grouping methods allow prior models of various sorts [32, 103, 104, 105] to segment a limited number of salient objects. Graph-cut methods incorporating a prior have been an important research topic in recent years as well [117, 4]. Unfortunately, work specifically incorporating priors relating to materials volumes has only more recently risen to prominence [44].

Intra-segment topology (i.e., segment connectivity) has been incorporated into MRF-based segmentation using graph-cut algorithms [120, 127, 76, 19] and level sets [40]. More specifically, in [120] a DijkstraGC algorithm was developed to enforce segment connectivity using interactively-specified points. In [76], a more general approach was developed to optimize the MRF cost function using a series of LP relaxations, with additional checks for segment connectivity. In [127], a Topology Cut algorithm was developed to insure segment connectivity in a MRF. In [19], a TopoCut algorithm achieves the desired intra-segment topology by modifying the underlying MRF before applying the graph-cut algorithm to minimize the cost func-

tion. This algorithm guarantees that the chosen connectivity property is preserved without explicitly enforcing it in the cost minimization and without using any kind of initialization. All these methods are formulated for binary segmentation with two labels: foreground and background. As discussed before, these two segments are always adjacent to each other and there is no explicit enforcement of inter-segment topology.

Inter-segment topology involves more than two segments (labels). In [128], a topology-preserving algorithm is designed to work with a small collection of semantic segments which can encode spatial relationships between objects using a quadratic programming approach on a conditional random field (CRF). In [28], Watershed Cut, a variant of watershed, is proposed that better fits the segmentation to local regions in an image by removing "destructible" points during a morphological operation. In [49], image edges are identified without small gaps or other properties that may result in topological inconsistencies, as determined by a learning algorithm. In [60], a combination of region merging and an edge map is used to integrate a consistent homolology measure to roughly enforce the "complexity" of the resulting segmentation using the Betti number β_1 during the merging process. These works tend to enforce a more general notion of topology and do not allow arbitrary (non-)adjacency constraints to be specified, or only handle a very small number of segments for which the topology is enforced.

Finally, regarding interactive segmentation, we note that materials science image segmentation is a domain that has been unaddressed in interactive segmentation literature, where there are no existing techniques focusing solely on segmenting materials images using an interactive approach. Other non-materials specific methods segment the object of interest using a model learned from user interactions [12, 116, 87]. Other approaches incorporate interaction into morphological operations (watershed) [106], co-segmentation [6], or incorporate machine-learning to aid in the interactive pro-

cess [111, 56]. These interactive methods have been applied to a number of domains, including natural images [87], medical images [10], and neuroimages [107, 106].

Examining the landscape of related work around materials image segmentation, it is very clear that the application of segmentation methods to 3D materials images is still in its infancy. The prevailing strategy when encountering a new material is to try off-the-shelf methods, extending them as necessary to produce reasonable performance. This has resulted in a variety of strategies, each highly tuned to a single dataset—and sometimes only a single slice from a dataset—with the expectation that further human intervention will be required to obtain a usable segmentation. No existing method has been thoroughly structured into a complete framework for addressing multiple, yet general, materials imaging problems across varied datasets.

Research Contribution

In this paper, we present a graph-cut based approach for materials image segmentation where we segment a sequence of images by repeatedly propagating a given 2D segmentation from one slice to another. Different from most existing tracking methods, we focus on the ability to segment a large number of segments through a 3D materials image. We show how topology inconsistency in the underlying image can be handled using a local relabeling method to improve performance during a segmentation propagation. Additionally, we show how prior properties of the desirable segments—namely shape, appearance, and topology—can be incorporated into this approach to segment different materials images, which illustrates the flexibility and broad applicability of the proposed approach. Furthermore, we extend this system to incorporate strict topological preservation in a multi-label framework. We introduce the notion of intra-segment topology and intersegment topology, and show how the proposed method can be extended to strictly preserve these types of topology while still propagating a large number of individual segments when propagating. Finally,

we integrate an interactive segmentation approach that allows minimal and simplistic human annotations which improve the underlying segmentation.

In summary, we

- Introduce a new multi-label propagation method to segment materials images using a graph-cut approach
- Handle topology inconsistency of the underlying structures by describing a local relabeling method
- Describe extensions for incorporating shape and appearance
- Define the notion of intra-segment topology and inter-segment topology, and extend the proposed method to maintain both types of topology
- Incorporate interactive annotations that can further improve the performance of the proposed method, along with three strategies to enhance this interactivity

The remainder of this paper is organized as follows: In Chapter 1, we describe the proposed global approach in its baseline form, and illustrate how important properties can be incorporated, including shape (Section 1.4) and appearance (Section 1.5). We then show the performance of the proposed method in Section 3.5 compared with competing methods. In Chapter 2, we extend the previous method to fully enforce segmentation topology during the segmentation, and show that this extension leads to better performance on datasets that have specific inter-segment topology. Finally, in Chapter 3, we show how the proposed method can be used in an interactive segmentation approach that further improves the segmentation performance.

Chapter 1

Global Propagation

1.1 Method

Similar to the traditional segmentation problem [71], we define *segmentation propagation*, as the problem of transferring a segmentation from a segmented image U to an unsegmented image V, subject to predefined constraints. Specifically, given an image U and a segmentation S^U of U such that S^U is a partition of the pixels in U into n segments

$$S^U = \{S_1^U, \ldots, S_n^U\}$$

where

$$U = S_1^U \cup \ldots \cup S_n^U \text{ and } S_1^U \cap \ldots \cap S_n^U = \emptyset,$$

we wish to obtain a segmentation S^V of an image V, which contains the same objects as U, by *propagating* S^U to V, yielding S^V. Notice that image U may not be directly required by this process, so S^U may be considered an initialization or derived from some other source or imaging modality.

The partition S may consist of many segments (i.e., $|S|$ may be large) and, as such, we further introduce the notion of *adjacency* between segments. We define $\mathcal{A}$ to be the set of unordered pairs $\{S_i, S_j\}$, indicating segment S_i and S_j are neighbors. A pair $\{S_i, S_j\}$ is a member of $\mathcal{A}$ iff there exists a pixel $p \in S_i$ and a pixel $q \in S_j$ such that p and q are 4-connected[1] pixels. For brevity, we notate $\{p, q\} \in \mathcal{P}_n$, where $\mathcal{P}_n$ is the set of pixel pairs that are neighbors. An example is shown in Figure 1.1,

[1] 4-connected pixels are pixels whose x or y coordinate differs by no more than 1

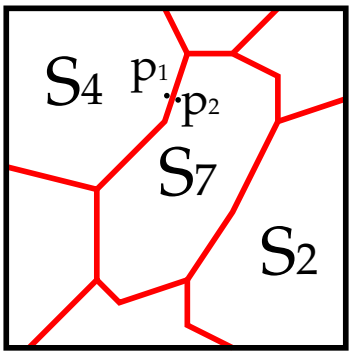


Figure 1.1: Example structure to illustrate the definition of $\mathcal{A}$.

where $\{S_4, S_7\} \in \mathcal{A}$ and $\{S_7, S_2\} \in \mathcal{A}$, but $\{S_4, S_2\} \notin \mathcal{A}$. Specifically, $\{S_4, S_7\} \in \mathcal{A}$ because pixel $p_1 \in S_4$ and $p_2 \in S_7$, and $\{p, q\} \in \mathcal{P}_n$.

In this work, we focus on the constraint that the mapping between the RAGs (Region Adjacency Graphs) of S^U and S^V is a *homomorphism*. There are two important properties that arise from enforcing homomorphism between the RAGs of the segmentations:

Property (1): $S_i^U \in S^U \Leftrightarrow S_i^V \in S^V$

Property (2): $\{S_i^U, S_j^U\} \notin \mathcal{A}^U \Leftrightarrow \{S_i^V, S_j^V\} \notin \mathcal{A}^V$

More descriptively, Property (1) requires that iff segments $S_i^U \in S^U$ then there is a corresponding segment $S_i^V \in S^V$, which results in all segments in S^U having a non-empty matching segment in S^V, and also implies $|S^U| = |S^V|$. Secondly, Property (2) results in $\{S_i^U, S_j^U\} \in \mathcal{A}^U$ iff $\{S_i^V, S_j^V\} \in \mathcal{A}^V$, meaning that adjacency relations among the segments between S^U and S^V remain the same. These properties have important application to materials image segmentation. Given the strictness of these properties, we approximate these properties in our formulation of segmentation propagation described below.

Given this problem definition and the above homomorphic constraint, we seek to derive a means to carry out such a propagation. We formulate a solution to this problem as an energy minimization over the partitioning of pixels in V to find S^V, given in the following form:

$$E(S^V) = \sum_{p \in V} \Theta_p(S_i^V) + \sum_{\{p,q\} \in \mathcal{P}_n^V} \Phi_{pq}(S_i^V, S_j^V). \tag{1.1}$$

This energy includes a unary term Θ_p which describes a cost for assigning a particular pixel p to a segment S_i^V, and a binary term Φ_{pq} which describes a cost for assigning two neighboring pixels p and q (i.e., $\{p, q\} \in \mathcal{P}_n$) to two segments S_i^V and S_j^V.

Incorporating Region Adjacency

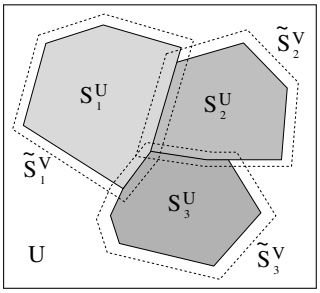
For this minimization formulation, we must define Θ_p and Φ_{pq} such that the aforementioned homomorphic constraint is integrated. For the unary term, we make an assumption that all pairs S_i^U and S_i^V have some overlap, varying most significantly around their boundaries. With this assumption we can formulate an approximation of $\Theta_p(S_i^V)$ that incorporates Property (1). Specifically, for every segment S_i^U, we construct a bounding region $\tilde{S}_i^V$, which contains all $p \in V$ that are within distance d of any pixel in S_i^U. Using the bounding region $\tilde{S}_i^V$, we set $\Theta_p(S_i^V) = 0$ for all $p \in \tilde{S}_i^V$ and $\Theta_p(S_i^V) = 0$ for all $p \notin \tilde{S}_i^V$. This bounding region stipulates the largest possible region that S_i^V may occupy. Note that the binary term, which we discuss below, provides additional information to localize $S_i^V \subseteq \tilde{S}_i^V$. The Θ_p term is defined by

$$\Theta_p(S_i^V) = \begin{cases} 0, & \text{distance}(p, S_i^U) < d \\ \infty, & \text{otherwise} \end{cases} \tag{1.2}$$

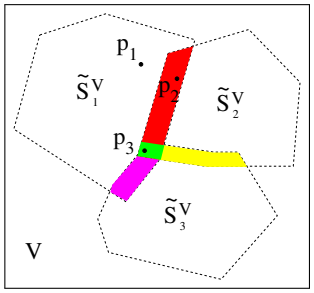
An example of defining Θ_p is shown in Figure 1.2, where the costs for p_1, p_2, and p_3 are shown for various assignments of S_1^V, S_2^V, and S_3^V. We additionally require

$$(\exists p \in S_i^V) \mid \Theta_p(S_i^V) = 0 \wedge \Theta_p(S_j^V) = \infty, j \neq i$$

which results in, at minimum, one $p \in S_i^V$ being given zero cost while having ∞ costs for all other $S_j^V, j \neq i$. We call this a *seed* point, and it is used to insure that no region may disappear entirely (fully enforcing Property (1)), which could otherwise occur if a small region S_i^U falls completely within one or more $\tilde{S}_j^V$ regions. In this paper, we select a constant distance parameter d, estimated from properties of the materials we segment, for each of our experiments. It is possible to select d based on other estimates, or a prior, but we found that the optimization allows for overestimates of d without overly adverse effects, as discussed later in Section 1.6. We also place seed pixels at the center of each $\tilde{S}_j^V$.



(a)



(b)

Θ	S_1^V	S_2^V	S_3^V
p_1	0	∞	∞
p_2	0	0	∞
p_3	0	0	0

(c)

Figure 1.2: An illustration of defining the unary term in the proposed approach. **(a)** Three adjacent S_i^U and associated $\tilde{S}_i^V$. **(b)** Three pixels which fall within various $\tilde{S}_i^V$. Specifically, p_1, p_2, and p_3, fall within one, two, and three bounding regions, respectively. **(c)** Unary term Θ defined for pixels p_1, p_2 and p_3 in (b).

Similarly, we define the binary term Φ_{pq} to insure that $\mathcal{A}^V$ is consistent with $\mathcal{A}^U$, modeling an approximation of Property (2); in particular, we model a weaker form:

Property (2b): $\{S_i^U, S_j^U\} \notin \mathcal{A}^U \Rightarrow \{S_i^V, S_j^V\} \notin \mathcal{A}^V$

where we solve only a single direction ($\Rightarrow$) of Property (2), which still disallows nonadjacent segments in S^U from becoming adjacent in S^V but does allow adjacent segments in S^U to become nonadjacent S^V. Because the unary term provides only a narrow band of d width in which boundaries may move, we find that this weaker

form provides similar robustness in practice. Given this, we set

$$\Phi_{pq}(S_i^V, S_j^V) = \begin{cases} 0, & i = j \\ \infty, & \{S_i^U, S_j^U\} \notin \mathcal{A}^U \\ g(p,q), & \{S_i^U, S_j^U\} \in \mathcal{A}^U \end{cases} \tag{1.3}$$

which introduces 0 cost for pixels assigned to the same segment, an ∞ cost for pixels assigned to segments that are not adjacent in S^U, and a cost functional g for pixels that are assigned to segments that are adjacent in S^U. Function g represents a probability that pixels p and q are along an image border, based on the intensities of p and q. As an example, if S^U is the segmentation shown in Figure 1.1, then Φ_{pq} would produce the following values for arbitrary pixels p, q such that $\{p, q\} \in \mathcal{P}_n$:

$$\Phi_{pq}(S_4^V, S_4^V) = 0$$
$$\Phi_{pq}(S_4^V, S_2^V) = \infty$$
$$\Phi_{pq}(S_4^V, S_7^V) = g(p,q)$$

For our method, we deal with two general types of images. The first, which we call *edge images*, show a higher intensity at the pixels along the boundaries of a segment than the pixels within each segment, which is achieved by by etching the metallic surface with an acid that attacks the grain boundaries preferentially [77, 3]. The second type, which we call *intensity images*, have segment boundary likeliness determined by the magnitude of the intensity gradient. For intensity images with pixels in the range $[0, 255]$, we define g as

$$g(p,q) = \exp\left(-\beta \left(V(p) - V(q)\right)^2\right) \tag{1.4}$$

where $V(p)$ is the intensity value of p in V, and we directly use the intensity values from the image itself as their likelihood to be on a segment boundary. For edge images, we define g as

$$g(p,q) = \exp\left(-\beta \max(V(p), V(q))^2\right). \tag{1.5}$$

In both, we use $\beta = (2\langle(V(p) - V(q))^2\rangle)^{-1}$ which is the expectation over all the slices in the volume [12].

Optimization Algorithm

In [38], it is shown that finding a global minimum of Equation (1.1) can be solved in polynomial time if there are only two segments (i.e., $|S| = 2$). However, [118, 11] proved that finding a general segmentation (often called a labeling) with $|S| > 2$ is an NP-hard problem, but a locally-optimal segmentation can be efficiently found with the α-β swap or α-expansion approaches, which use iterative applications of the min-cut max-flow algorithm [1] in a Markov Random Field (MRF)[55], run for each segment or pair of segments. We use the α-β swap approach, which produces a locally-optimal segmentation by finding an optimal assignment for each pair of segments. As discussed below, we use this approach because our application can use a constrained α-β swap, making it as efficient as α-expansion. We illustrate the graph formulation for a single pair, S_1 and S_2, below.

The graph formulation is shown in Figure 1.3, where nodes are constructed for each pixel (p-nodes) and for each segment (S-nodes). The binary terms are encoded into the edge weights between adjacent p-nodes, and the unary term is encoded in the edge weight from each p-node to both S-nodes. The α-β swap approach selects a pair of segments $S_i^V, S_j^V \in S^V$ and creates the above graph for pixels in $S_i^V \cup S_j^V$ and two S-nodes for the two considered segments. Running the min-cut max-flow algorithm on this graph has been shown to determine the optimal assignment of pixels between S_i^V and S_j^V satisfying Equation (1.1). It does so by reducing the problem of segmenting a set of pixels to the problem of partitioning a graph into two disjoint subgraphs by removing edges with minimum total edge weight. These subgraphs represent a new, optimal grouping of S_i^V and S_j^V. In the α-β swap approach, this process is repeated sequentially for each pair of segments in S^V to find a locally-optimal segmentation

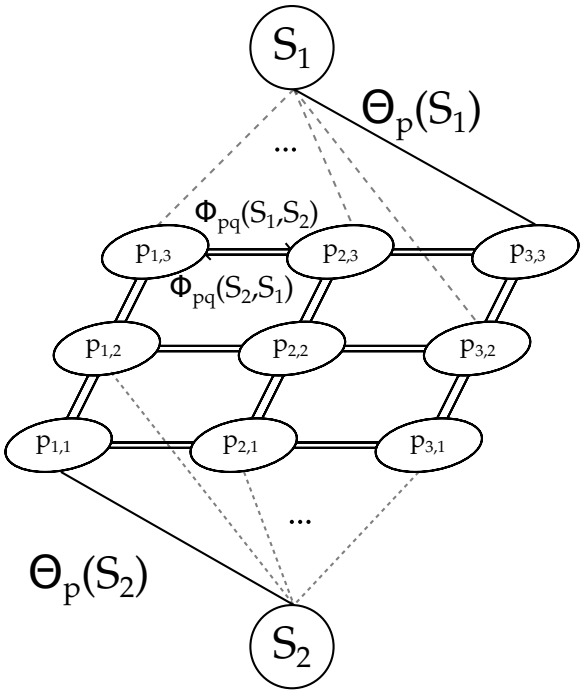


Figure 1.3: Graph formulation of the unary and binary terms for the min-cut max-flow algorithm for a pair of segments S_1 and S_2.

satisfying Equation (1.1).

By observation of Property (2) and the above definition of Equation (1.3), moving any pixel between S_i^V and S_j^V where $\{S_i^U, S_j^U\} \notin \mathcal{A}^U$ will clearly result in an ∞ penalty by Equation (1.3). Thus, we introduce a constrained α-β swap approach by only performing a swap (i.e., creating an MRF and updating the segments) between pairs of partitions whose analog partitions in S^U are in $\mathcal{A}$, i.e., $\{S_i^U, S_j^U\} \in \mathcal{A}^U$. Using S^U as an initialization allows our constrained α-β swap to arrive at exactly the same local minimum as the unconstrained α-β swap.

As a final note, we define topology only among different segments, not within a single segment itself (connectivity) [127], and the graph-cut method which minimizes Equation (1.1) cannot guarantee the connectivity of the individual segments. For the proposed method, the segmentation boundary is restricted to a narrow dilated band between segments with zero cost for the overlapping dilated regions. Thus the proposed method, when using such zero-cost dilated regions, will not obtain discon-

nected segments. When the Θ term is defined differently, such as Equation (1.8) later discussed in Section 1.5, we simply merge any resulting disconnected fragments into their containing segment. This completes the definition of a topologically-constrained graph cut-based segmentation propagation algorithm, which we will use and extend in the following sections, where we show how the proposed approach can be extended to incorporate important prior knowledge and properties that are relevant to 3D materials images.

1.3 Local Non-Homomorphism

The general approach proposed in Section 1.2, which we call the *global* method for the remainder of this work, enforces RAG homomorphism between slice segmentations when propagating from one slice to another. It is sufficient when the segments in the two slices have exactly the same adjacency relations, such as slices U_1 and U_2 in Figure 1.4. However, local non-homomorphism may occur between two slices when a new segment appears or an existing segment disappears when moving from slice U_k to U_{k+1}, as illustrated in Figure 1.4, moving from U_2 to U_3 and from U_3 to U_4, respectively.

We introduce a *local relabeling* strategy that can be run after the global method to identify local areas where local non-homomorphism (i.e., segment appearance or disappearance) may occur, and then rerun the energy minimization in these local areas to update the segmentation, accommodating possible non-homomorphism inconsistencies.

The local non-homomorphism strategy is summarized in Algorithm 1, which takes as input the global method results from Section 1.2. The first loop on line 3 accommodates the possible disappearance of an existing segment in U_{k+1} and the second loop on line 10 accommodates the possible appearance of a new segment in U_{k+1}. The mechanisms for identifying candidate local regions (line 2 and line 9) and identifying

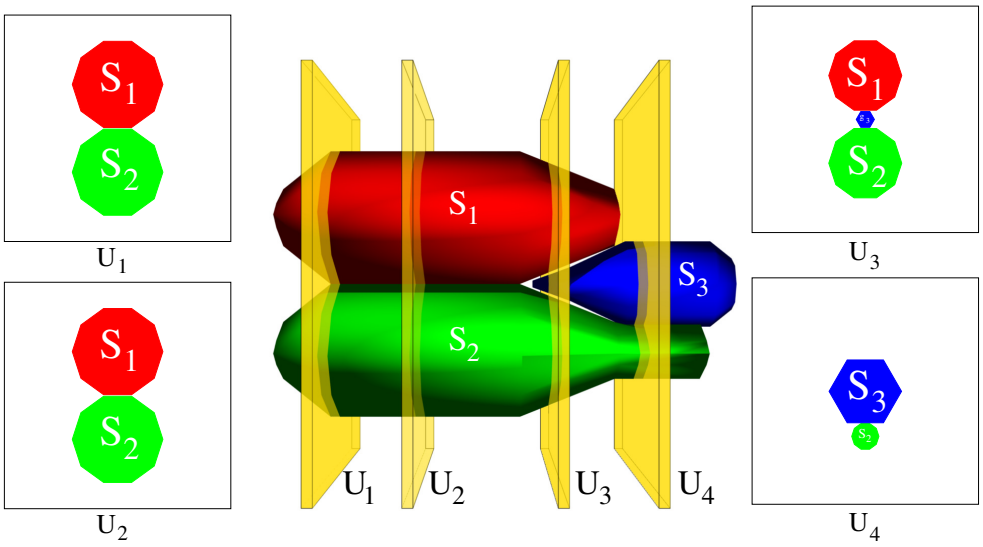


Figure 1.4: 2D structure consistency ($U_1 \leftrightarrow U_2$) and non-homomorphism ($U_2 \leftrightarrow U_3$ and $U_3 \leftrightarrow U_4$) across slices.

Algorithm 1 Algorithm for handling local non-homomorphism.

```
1:  function NON-HOMOMORPHIC(G)
2:      R_D ← DISAPPEARANCECANDIDATES(G)
3:      for each r ∈ R_D do
4:          Θ_p, Φ_pq ← defined as described in Section 1.2
5:          Remove ∞ penalty from Φ_pq
6:          G_r ← graph cut on local region r
7:          if VERIFY(G_r) then
8:              Update G with segments G_r
9:      R_A ← APPEARANCECANDIDATES(G)
10:     for each r ∈ R_A do
11:         s ← LOCATESEED
12:         Redefine Θ_p, Φ_pq by creating a new segment at s
13:         Remove ∞ penalty from Φ_pq
14:         G_r ← graph cut on local region r
15:         if VERIFY(G_r) then
16:             Update G with segments G_r
17:     return updated G
```

seeds for a new segment (line 11) will vary for different materials images. An example definition is given in our experiments in Section 1.6.

In line 5 and line 13, we remove the ∞ penalty from Φ_{pq} by setting

$$\Phi_{pq}(S_i^V, S_j^V) = \begin{cases} 0, & i = j \\ g(p,q), & \text{otherwise} \end{cases} \tag{1.6}$$

This allows all segments to be adjacent to each other without penalty in a local region r which might have prevented the disappearance of an existing segment, and allow the appearance of a new segment. The verification steps (line 7 and line 15) further verify the local non-homomorphism by using various image information and/or other structural knowledge; an example verification strategy will be discussed in our experiments in Section 1.6.

On line 12, we redefine the unary term such that $\Theta_p = 0$ when the seed pixels are given the new segment, and $\Theta_p = \infty$ when given any other segment. For other pixels, we define Θ_p to assign each possible segment by following a similar strategy as described in Section 1.2: $\Theta_p = 0$ when we are confident that a pixel must have a specific segment and $\Theta_p = \infty$ when we are confident that a pixel should not have a specific segment. The binary term can be defined by following a similar strategy as described in Section 1.2, but with the ∞ penalty removed (line 13). In this manner, the local non-homomorphism strategy will enforce the addition of new segments at seed pixels and the verification step (line 15) will finally determine whether we keep such new segments or not.

An example of the local non-homomorphism strategy used on a real material is shown in Figure 1.5, which illustrates both segment disappearance and appearance. The middle column is the global method, and shows example seed points (green dots) for possible segment appearance, which also show an associated segmentation (yellow dashed line) generated by the local graph cut. The verification step will only select segmentations that correspond to correct non-homomorphism, as shown in the last column of Figure 1.5, where the proposed strategy is able to remove erroneous segments, and introduce segmentations of segments that appear in the new slice U_{k+1}.

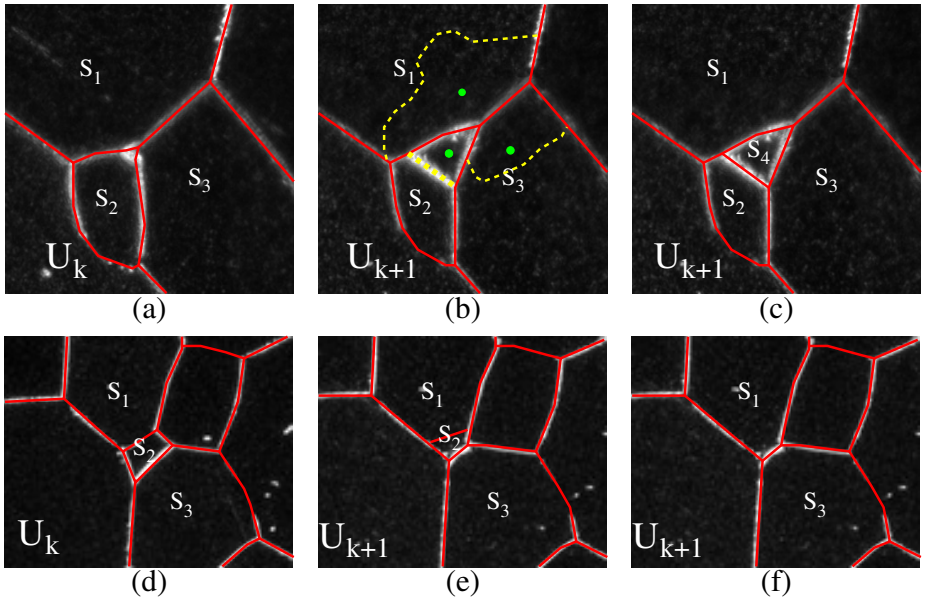


Figure 1.5: Illustration of non-homomorphism. **(a,d)** Segmentation U_k to be propagated to the neighboring U_{k+1} slice. **(b,e)** The homomorphic method result where non-homomorphisms are not handled. Seed points (green dots) and associated segmentations (yellow dashed lines) show how the proposed non-homomorphic strategy identifies and attempts to handle segment appearance. **(c,f)** The non-homomorphisms handled by the proposed method, where the proper segmentation was selected by the VERIFY function and erroneous segments are removed.

1.4 Shape Preservation

For some materials structures, the general shape of segments may be distinct and complex, such as the example shown in Figure 1.6 (a). The dilation strategy presented in Section 1.2 is unable to capture the small details of the structure's shape, as shown in Figure 1.6 (c) which is missing large portions of the boundary thereby reducing the quality of the segmentation and subsequent analysis of the material. Knowing that, often, the intra-slice resolution is high enough to have a number of slices containing the same segment, successive slices of the same structure usually exhibit consistent shapes [131], at least through several sections. In this section, we propose a strategy to preserve the 2D shape of selected segments when propagating a segmentation from U_k to U_{k+1}.

Specifically, we perform a skeletonization [58] in U_k for each segment for which

we desire shape consistency. An example is shown as shown in Figure 1.6 (b), where the skeletonization is performed for both the foreground segment (in red) and background (in blue). Note that only a single foreground and background are shown for illustrative purposes here. In the general case with multiple foreground objects (as in the experiment in Section 1.6), skeletons are extracted for each foreground object and the background object. In slice U_{k+1}, we enforce the desired segment of these skeleton pixels to be the same as their segments in U_k, e.g., red pixels have a desired segment of foreground segment and blue pixels have a desired segment of background. In practice, we may morphologically erode the skeletons constructed in U_k slightly when propagating them to U_{k+1} to account for changes of the segment's size. For a skeleton pixel p with a desired segment S_l^V, we further update its unary term in Equation (1.1) by a Dirac delta function

$$\Theta_p(S_i^V) = \begin{cases} 0, & i = l \\ \infty, & \text{otherwise} \end{cases} . \tag{1.7}$$

With this updated unary term, we run the graph cut algorithm to segment all the remaining pixels in U_{k+1} and obtain a segmentation of U_{k+1} that preserves the shape of the segments of interest. Note that many materials contain multiple segments and this shape-preserving strategy can be applied to all the segments, or a selected subset of segments, according to the requirements of different applications. If we do not want to preserve the shape of a certain segment, we simply use the dilation strategy in Section 1.2 for this segment and do not update the Θ_p values of its skeleton pixels. Figure 1.6 shows a comparison between our baseline approach using the unary term as in Section 1.2 vs. the method outlined in this section to propagate a manually created segmentation in U_k to a new slice U_{k+1}.

Since the incorporation of shape is a modification of the existing unary term, it can be easily implemented after the unary term from Section 1.2 is computed. Furthermore, the skeletonization can be computed very quickly for all the segments

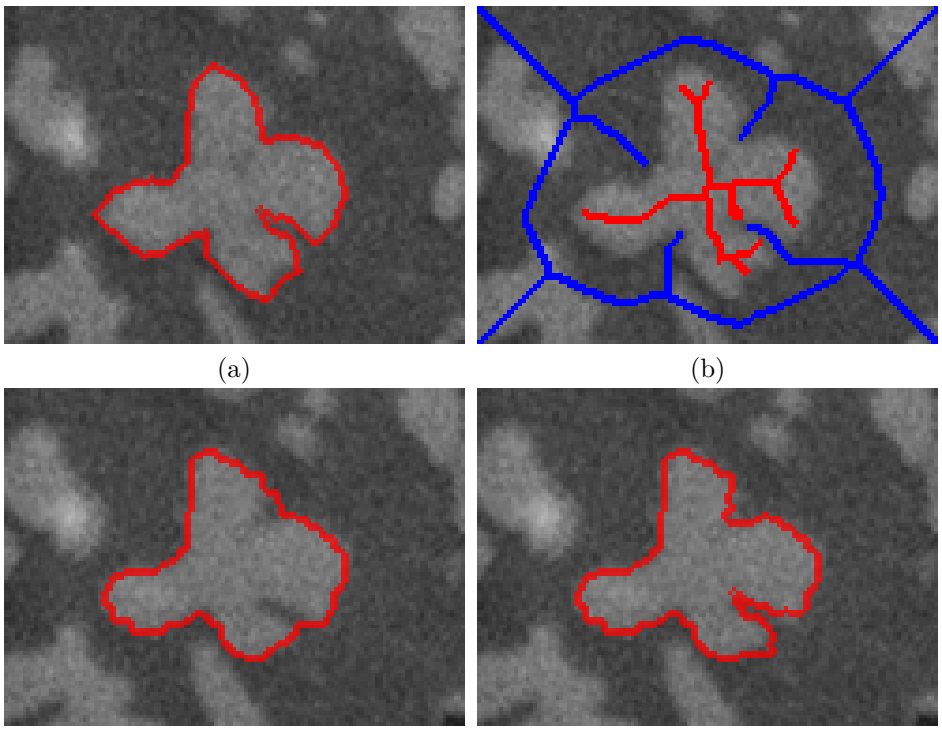


Figure 1.6: Dendritic γ' precipitates in Rene88DT. **(a)** Segmentation of slice U_k, created manually. **(b)** Skeletonizaton of the segmented segments in slice U_k (morphologically eroded slightly), showing the shape of the foreground segment (red) and background (blue). **(c)** Segmentation results on slice U_{k+1} using the global method in Section 1.2. **(d)** Segmentation result on slice U_{k+1} by incorporating the proposed shape-preservation strategy.

in a materials slice, and is viable for use on all of the (sometimes hundreds of) segments within a material, in combination with any of the other techniques discussed in this paper.

1.5 Intensity Preservation

For some materials images, the intensity within a segment can be another important property which may be desirable to preserve when propagating a segmentation. As shown in Figure 1.7, a segment might undergo changes in size or shape from one slice to the next while retaining a consistent intensity, separating it from surrounding segments or background, excepting image noise. Note that while the binary term

Φ_{pq} discussed in Section 1.2 does incorporate pairwise intensity difference between neighboring pixels to determine whether they are likely to have the same segment or different segments, it does not capture the *overall* intensity of a specific segment. In this section, we propose a strategy to incorporate this intensity as a prior by updating the unary term Θ_p in Equation (1.1) when propagating a segmentation. It can be easily extended to preserve other intensity features, such as color or texture.

Our key observation is that the overall intensity of many segments can be modeled as a simple Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ to account for image noise. By computing such a Gaussian for every segment in U_k, we can determine the probability $P(V(p)|\mu, \sigma^2)$ of a pixel p in slice U_{k+1} to be assigned a specific segment. For each segment $S_i^{U_k}$ from slice U_k and corresponding Gaussian $\mathcal{N}(\mu_i, \sigma_i^2)$, the unary term Θ_p can be defined by evaluating all the pixels in slice U_{k+1} against this distribution by using a negative log likelihood function

$$\Theta_p(S_i^V) = -\ln[P(V(p)|\mu_i, \sigma_i^2)]. \tag{1.8}$$

In practice, we limit this likelihood to the same dilated region to better enforce spatial coherence between slices. Figure 1.7 shows a comparison between our baseline approach using the unary term as in Section 1.2 vs. the method outlined in this section to propagate a manually created segmentation in U_k to a new slice U_{k+1}.

Similar to the shape-preservation strategy discussed in the previous section, it is possible, and fast, to apply the intensity-preservation strategy on all of the many segments or a selected subset of them in a slice. In addition, the proposed intensity preservation strategy does not require that the intensity of a segment remain consistent over all the slices where this segment is present. It only requires intensity consistency (including slow intensity changes) between neighboring slices; the intensity of a segment may change substantially after propagating a number of slices.

Furthermore, it is easy to combine the shape preservation strategy described in Section 1.4 and the intensity preservation strategy described in this section, by up-

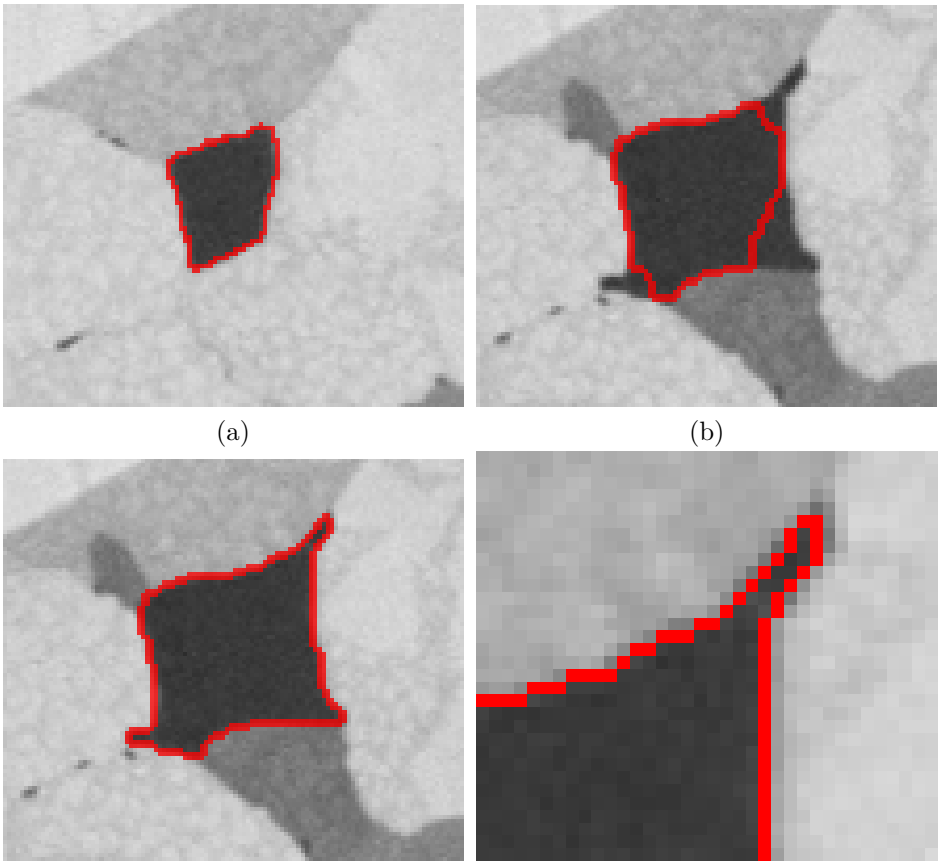
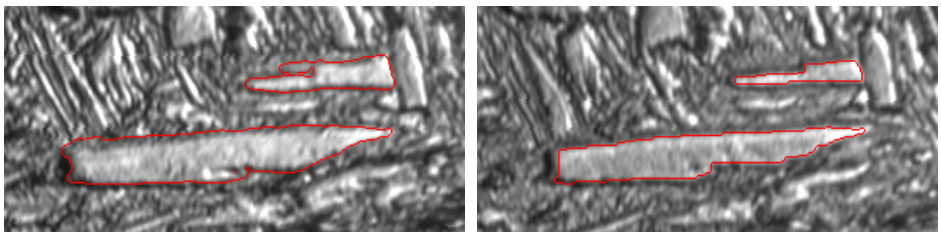


Figure 1.7: Grain structure of IN100 superalloy [39]. Image courtesy of Mike Groeber, AFRL. **(a)** Segmentation of slice U_k, created manually. **(b)** Segmentation result on slice U_{k+1} using the global method in Section 1.2. **(c)** Segmentation result on slice U_{k+1} using segment intensity preservation between U_k and U_{k+1}. **(d)** Zoomed view of upper-right corner of (c).

dating the unary term as described in Section 1.4 and as described in this section. A sample result is shown in Figure 1.8 where the propagation from Section 1.2 is augmented with both the shape preservation strategy and the intensity preservation strategy. In addition, these two strategies can also be easily combined with the local relabeling strategy described in Section 2.1 by applying them in identified local regions instead of the entire image.



(a) (b)

Figure 1.8: Martensite lath structure in steel. **(a)** Rough segmentation of slice U_k created manually. **(b)** Segmentation result on slice U_{k+1} using both shape preservation and intensity preservation strategies.

Table 1.1: Summary of datasets used in experiments, detailing the number of slices, image size, number of segments (n), type of material, and segmentation time.

#	Name Slices	Slice Size	n	Type	Time	
1	Ti-26 [89]	11	750×525	$\sim$120	Metal	15 min
2	Steel [88]	11	1600×1200	$\sim$7	Metal	$<$ 1 min
3	Cotyledon [126]	38	1024×1024	$\sim$700	Bio	40 min
4	Epoxy [52]	12	1288×957	$\sim$160	Polymer	10 min

Table 1.2: Overview of manually-constructed ground truth segmentation, showing the time needed for a human to segment each slice, the width ϵ of the boundaries (varies due to resolution) before they are thinned, and the average local non-homomorphism $\mathcal{NH}$ for each dataset.

#	Name	Manual Seg. Time	ϵ	$\mathcal{NH}$
1	Ti-26	17 minutes / slice	3px	0.053
2	Steel	5 minutes / slice	3px	0
3	Cotyledon	40 minutes / slice	2px	0.0095
4	Epoxy	28 minutes / slice	2px	0.0914

1.6 Experiments

In this section, we use four 3D materials images to evaluate the performance and illustrate the versatility of the proposed method. Each consists of a sequence of 2D

Table 1.3: Summary of methods and parameters used on each dataset.

#	Dataset	Methods	Φ_{pq}	Dilation
1	Ti-26	Sec. 1.2, 2.1	Eq. (1.3), (1.5)	d=20px
2	Steel	Sec. 1.2, 1.4	Eq. (1.3), (1.4)	d=10px
3	Cotyledon	Sec. 1.2	Eq. (1.12)	d=10px
4	Epoxy	Sec. 1.2, 1.5	Eq. (1.3), (1.4)	d=15px

image slices and has different sizes, imaging modalities, properties, and varying levels of homomorphism, as summarized in Table 1.1. Samples are shown in Figure 1.9. Specifically, to measure the local non-homomorphism of a sample, as shown by $\mathcal{NH}$ in Table 1.2, we find

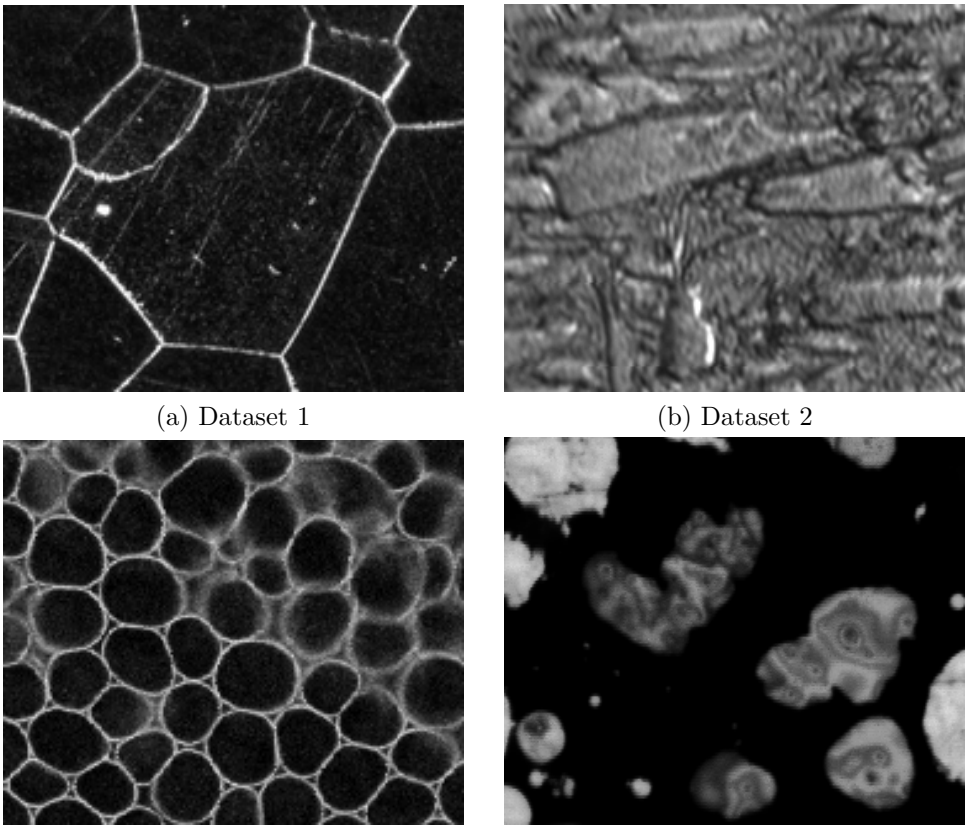
$$\mathcal{NH} = \frac{1}{n-1} \sum_{k=1}^{n-1} \frac{|\mathcal{A}^{U_k} \cap \mathcal{A}^{U_{k+1}}|}{|\mathcal{A}^{U_k} \cup \mathcal{A}^{U_{k+1}}|}$$

which is the average of the Jaccard similarity coefficient of the adjacency sets over all neighboring pairs of slices. Thus this measure shows how applicable the proposed method will be without the extensions proposed in Section 2.1. For all datasets, when running the proposed method, we start with a ground-truth segmentation on a single *initial* slice, propagating it to segment the neighboring slice(s). We then propagate the segmentations on the neighboring slice(s) to segment their neighboring slices. This process is repeated until all the slices are segmented.

Evaluation Metric

For performance evaluation, we have manual ground-truth segmentations of all slices, as summarized in Table 1.2, provided by materials scientists and/or constructed by trained image processing experts. It is important to note that materials segmentation often has very stringent requirements such that any missed or erroneous boundaries must be manually corrected. Thus, methods that reduce this error greatly reduce the need for manual intervention when doing modeling or analysis.

We evaluate a segmentation result by computing the coincidence between the detected boundaries and the ground truth segment boundaries. Specifically, we erode all boundaries (ground truth and evaluated boundaries) to a single pixel in width, counting true positives among the evaluated boundary pixels that are within a fixed distance of the ground truth, and treat all other pixels as false positives. Similarly, ground truth boundary pixels that are not within the same distance of the evaluated boundary pixels are taken as false positives. We use the associated unthinned



(a) Dataset 1 (b) Dataset 2

(c) Dataset 3 (d) Dataset 4

Figure 1.9: Cropped samples of a single slice from of each dataset. **(a)** β-Ti grains in Ti-21S [89], courtesy of Dave Rowenhorst, NRL. **(b)** Martensite lath structure in steel [88], also courtesy of Dave Rowenhorst, NRL. **(c)** Cotyledon plant embryo [126] provided by JC Palauqui, INRA. **(d)** NanoEnergetic Epoxy [52], obtained from [47].

boundary width for the evaluated dataset, shown by ϵ in Table 1.2, as this evaluation distance. Doing so makes this measurement independent of resolution.

For efficiency, a distance transform is pre-computed for both the ground truth edge pixels and the evaluated edge pixels to speed up the distance calculation. With these measures, we then calculate the precision and recall [78] on all propagated slices (excluding the initial slice). We also calculate a summary F-measure that integrates the precision and recall using the harmonic mean [66]. The x-axis for all figures shows the slice number.

With these measures, we then calculate the precision and recall, using the well-

known formulas [78],

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{1.9}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{1.10}$$

on all propagated slices. We also calculate a summary F-measure, using

$$\text{F} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.11}$$

that integrates the precision and recall using the harmonic mean [66].

For every dataset using proposed methods, we try every possible selection of initial slice to show the robustness of the proposed method. We then aggregate the performance resulting from different initial slices by finding an average and standard deviation for each slice and show an average and error bars in figures for experiments that are performed in this manner. The methods and parameters used for the proposed method are summarized in Table 1.3. For comparison methods that are not based on propagation (i.e., treat every image individually without an initial slice), we simply show the raw performance without error bars. In no way is the ground truth included in the performance scores for any of the methods directly.

We additionally introduce another measure to help verify that a segmentation is indeed reflective of a good segmentation for materials imaging purposes: the difference in number of segments, which we call the *cardinality difference*. Specifically, we measure $R = |G| - |S|$ where $|G|$ is the number of distinct segments in the ground truth, and $|S|$ is the number of segments in the evaluated segmentation. Thus, positive values indicate undersegmentation, while negative values indicates oversegmentation.

We conducted all our experiments on a single core of a 2GHz Linux workstation with 8GB of memory. The methods are implemented in a combination of C/C++ using the OpenCV [13] library, and Python using the SciPy [51] and NumPy libraries. No single propagation requires more than 40 minutes to complete with any

of the methods discussed in this paper. Specific runtimes are shown in Table 1.1. The proposed framework, particularly when identifying local non-homomorphism as discussed in Section 2.1, can benefit from a parallelized implementation.

Comparison Methods

To justify the proposed method, we compare its performance to three widely used 2D image segmentation methods: watershed, normalized cut, and thresholding (on Dataset 4 only). More specifically, we use a MATLAB [67] implementation of the watershed method [121] based on [69], and the normalized cut method [98], using a linear-time multiscale implementation based on [27]. For the thresholding method, we use Otsu's method [79] to determine the threshold. While these methods are developed to segment each 2D slice independently, we further extend the watershed method to propagate a segmentation from one slice to another for an apples-to-apples comparison with the proposed method.

For general-purpose materials image segmentation, there is no systematic study to show which method is the state-of-the-art. The comparison methods were selected because: (1) all are widely-used, (2) watershed has been shown to be particularly suitable for segmenting grain or cell-like structures, and can be adapted to consider 3D structure continuity between slices which we find to be particularly beneficial to 3D materials image segmentation, and (3) normalized cut is representative of a global image segmentation, which has attracted more attention in recent years.

For the watershed extension, which we call *propagated watershed*, *markers* for each non-background segment are identified and propagated to a new slice to enforce consistency between slices [18], starting from the same initial slice used in the proposed method. When propagating from S^U to V to obtain S^V, we erode each segment S_i^U by the same d value (used by the proposed method for dilation) and take this as the marker for the propagated watershed method. This actually restricts the segmenta-

tion boundary resulting from the propagated watershed method to be fully located within the same dilation region as in the proposed method. For small segments, such erosion may completely remove them. In this case, we either take the segment's center as a single-pixel marker, or omit the marker for this segment entirely. For each dataset we try both strategies, and report the better performance for propagated watershed. In addition, we tested 50 different minima suppression levels, selecting the best for each individual dataset.

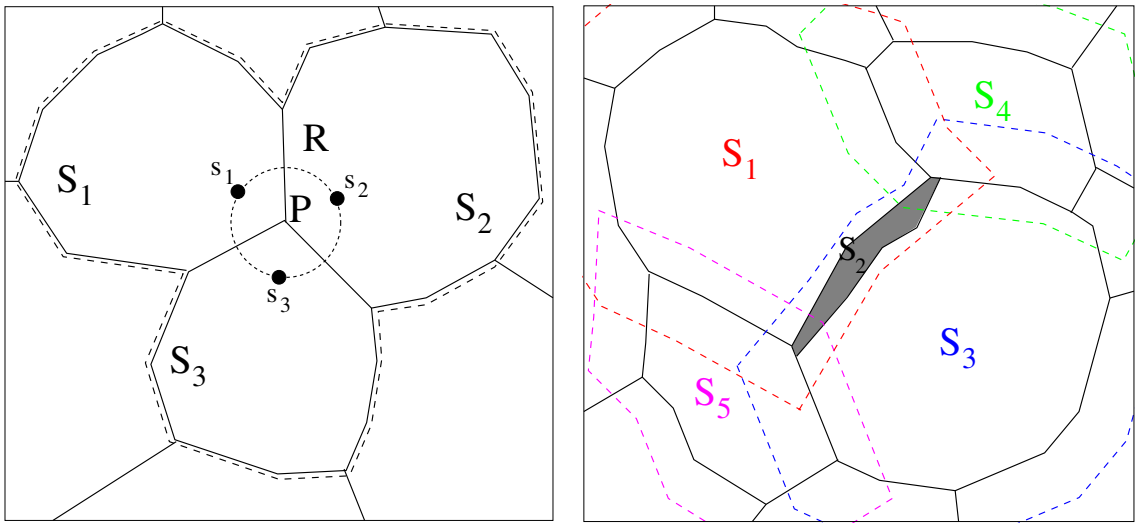
The normalized cut method requires the number of segments as input, so we provide the exact number of segments from the ground truth for each slice. Due to this, the cardinality difference (see Section 1.6) for normalized cut is always near zero. For edge-based images, we modify the normalized method accordingly.

Due to the limitations of the normalized cut implementation, we scale down high-resolution slices before applying the normalized cut algorithm, and then scale the resulting segmentation back to the high resolution (using nearest-neighbor approximation) before evaluation. Because of this, normalized cut's qualitative results may exhibit small "blocky" errors along the boundaries.

For all the comparison methods, we compute the same precision, recall and F-measure scores. For the propagated watershed method we evaluate it in the same manner as the proposed method by trying different initial slices and computing an average and standard deviation of the performance at each slice over different initial slices. Since the remaining comparison methods segment each slice independently and are not dependent on the selection of an initial slice, each slice has only a performance score, without a standard deviation.

Dataset 1: β-Ti grains in Ti-21S

In our experiments, we use a sequence of 11 microscopic titanium images to show the importance of incorporating local non-homomorphism as discussed in Section 2.1. As



(a) Triple Junction (b) Small grain

Figure 1.10: Illustration of seed point identification in Dataset 1. Seed points for each grain are shown as $s_i, \ldots, s_n$. **(a)** Seed points s_1, s_2, s_3 around triple junction P at distance R. **(b)** Disappearance candidate S_2 that will disappear after applying the proposed local non-homomorphic strategy because S_2 is located within the union of the dilated regions of its neighboring segments. The dilated region of each neighboring segment is enclosed by a dashed curve of the same color as its segment label.

edge images, each Ti-21S slice has a resolution of 750×525, and consists of ~ 120 β-Ti grains, which are the segments of interest, as shown in Figure 1.9 (a). These grains are all adjacent, meaning that there is no notion of a "background" in this material. The dilation size is set to 20 pixels for constructing the unary term in the homomorphic method.

Since this dataset may have local non-homomorphism, and all associated objects are adjacent to each other, we use the strategy presented in Section 2.1. Thus, we must define several functions: DisappearanceCandidates(G), AppearanceCandidates(G), LocateSeed, and Verify(G_r). For this material, we identify each segment (resulting from the global method) that is less than 200px in size as a candidate segment for disappearance. In DisappearanceCandidates(G), we take each such segment, combined with all its adjacent segments, as a local candidate region. For example, considering segment S_2, we construct the local candidate region consisting of

$S_1 \cup \ldots \cup S_5$ as shown in Figure 1.10 (b). For AppearanceCandidates(G), we consider the possible appearance of a new segment at each triple junctionTherefore, the local candidate region is made up of the segments around each triple junction, as shown by the regions surrounded by dashed curves in Figure 1.10 (a).

For segment appearance, we must additionally define LocateSeed such that we can identify where segments may emerge in the new slice U_{k+1}. Around triple junctions, we select seeds at a fixed radius R (30px for this material) from the center of each triple junction, as illustrated in Figure 1.10 (a). Finally, for Verify(G_r), we check each segment introduced in the local non-homomorphism strategy, keeping this segment if more than 66% of this new segment's boundary has an intensity that is higher than

$$\frac{3}{2|U_{k+1}|} \sum_{i=1}^{|U_{k+1}|} p_i,$$

which is $\frac{3}{2}\times$ the average intensity of all the pixels in slice U_{k+1}. This makes Verify adaptive to the overall brightness of a considered slice. For the case of segment disappearance, if an existing segment disappears in the local non-homomorphic strategy, Verify keeps this segmentation.

For this dataset, we conduct three additional experiments to better illustrate our method's contribution. First, we show the effect of different values of d with an experiment in Figure 1.11 where the evaluated performance of the proposed method, across multiple slices, becomes roughly uniform once the value of d is large enough to accommodate the variation from one slice to the next.

Second, we conduct another experiment that uses an adaptive selection of d. Specifically, this is done by finding a corresponding segment $\tilde{S}_i^V$ from the watershed segmentation result for each S_i^U and using the size change to estimate the value of d for each S_i^U. The performance of this method ("Proposed w/ Adaptive d") is roughly comparable to the fixed-d experiment ("Proposed w/ Fixed d"), as shown in Figure 1.13.

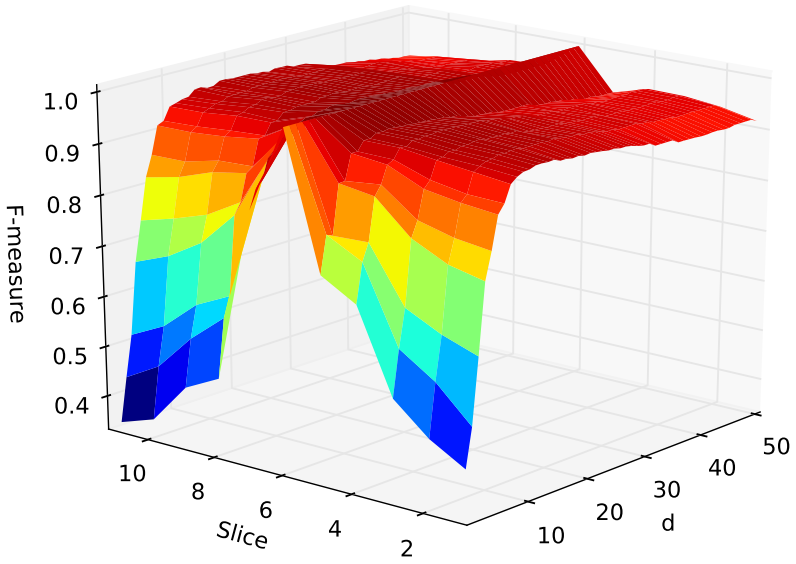


Figure 1.11: Segmentation performance on the 11 slices in Dataset 1 using different values for the dilation paramter d. The ground truth for slice 6 is used as the initial segmentation S^U for each run.

Finally, we conduct an experiment to show the necessity of considering homomorphism in materials segmentation. Instead of using explicit homomorphic and non-homomorphic methods, we conduct a single round of energy minimization, where the ∞ penalty in Equation (1.3) is removed to allow structures to disappear, and a dummy segment is introduced in the same dilated region used by the homomorphic method to capture possible appearing structures. From our experimental results, we also slightly penalize non-dummy segments in the Θ_p term to obtain better performance. If the resulting dummy segment contains multiple, disjoint regions, we take each of these disjoint regions as individual segments for further propagation. After trying many non-dummy segment penalties for Θ_p, we include the best possible performance of this dummy segment-based method ("Dummy Segment") in Figure 1.13. We can see that this dummy segment-based method does not perform as well as the proposed method. The major reason is that between neighboring slices, the topol-

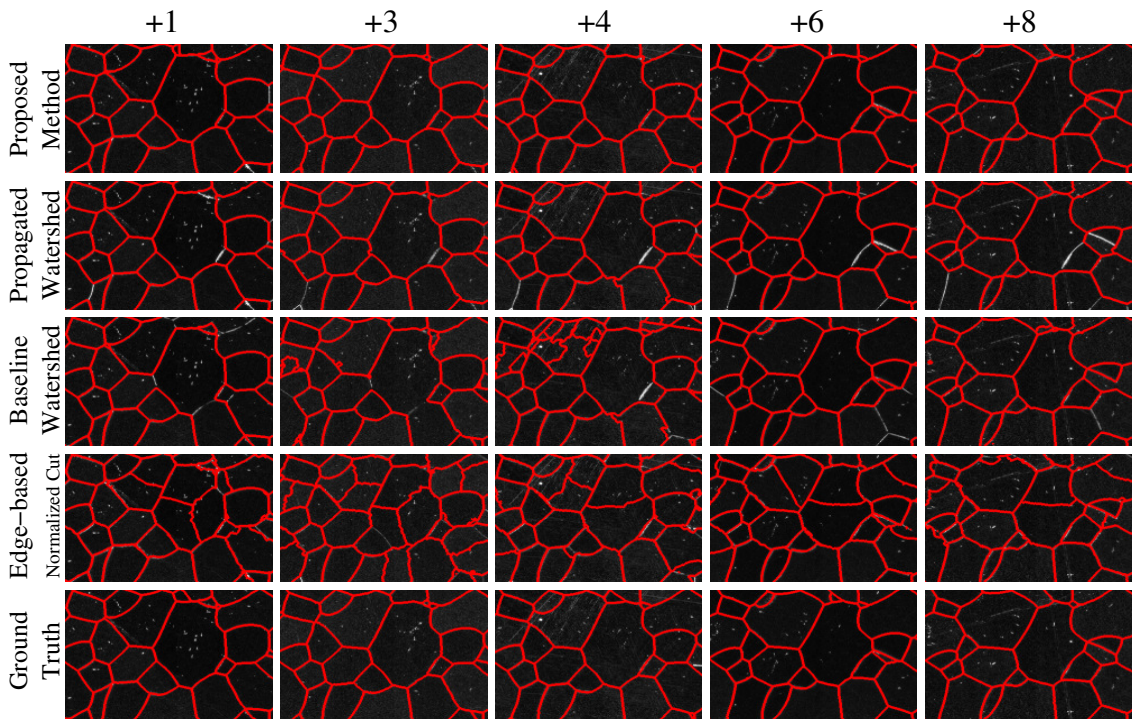
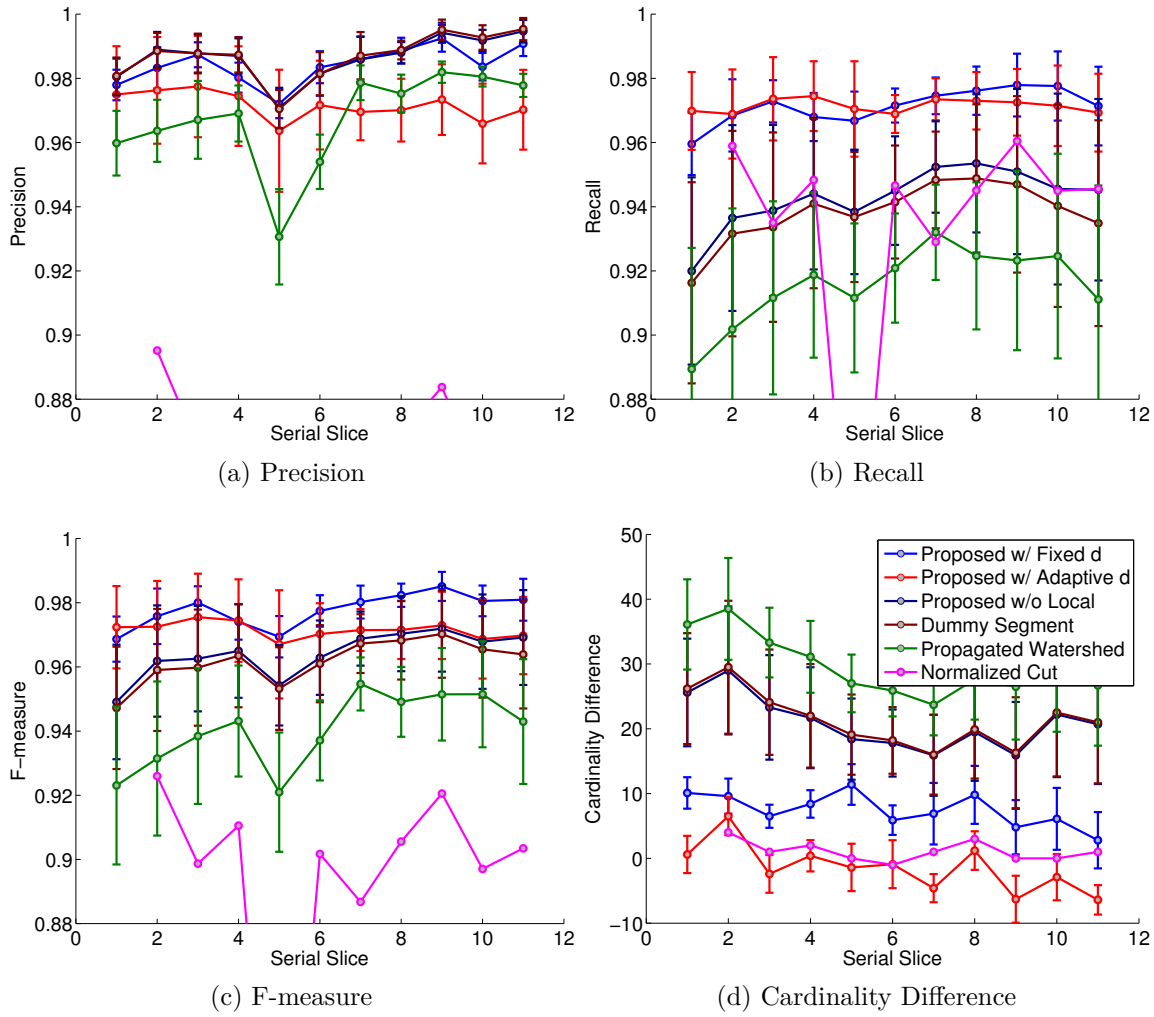


Figure 1.12: Zoomed view of segmentation results on selected slices using the proposed method, propagated watershed, and edge-based normalized cut, along with the ground truth on Dataset 1. Each column shows a slice of different distance from the initial slice.

ogy is largely homomorphic, with only some small, local non-homomorphism. The proposed method considers both of these two properties: starting with the homomorphism propagation in the whole image, followed by a local non-homomorphic method. The dummy segment-based method does not consider any homomorphism at all, and thus leads to poorer performance.

Although the baseline watershed method has been used to segment images with many grains or cells [63], its performance on this dataset is not satisfactory, largely due to oversegmentation, as shown in Figure 1.12. The propagated watershed method performs better than the baseline watershed method. However, the propagated watershed is not able to identify new grains as they appear when the series section plane moves to a new slice. This is evident by the small missed grain boundary segments in the last image of the second row of Figure 1.12. As a state-of-the-art



(a) Precision (b) Recall

(c) F-measure (d) Cardinality Difference

Figure 1.13: **(a-c)** The segmentation precision, recall, and F-measures for the proposed method, the watershed method, and normalized cut on the 11 tested slices of Dataset 1. **(d)** Cardinality difference measure for all evaluated methods. "Proposed w/ fixed d" is the proposed method with a fixed value for the dilation d term. "Proposed w/ adaptive d" is the proposed method where the value for d is computed from a rough watershed segmentation of the underlying image. "Proposed w/o local" is the proposed method without the local non-homomorphic step discussed in Section 2.1. Finally, the "Dummy Segment" shows the performance of replacing the non-homomorphic step with a single "dummy" label as discussed in the text.

image segmentation method, normalized cut produces a low F-measure. We found that normalized cut places many boundaries through grain centers, which is partly due to its preference to partition images to similar-area segments and the presence of scratches and other noise in some slices. Note that we use edge-based normalized

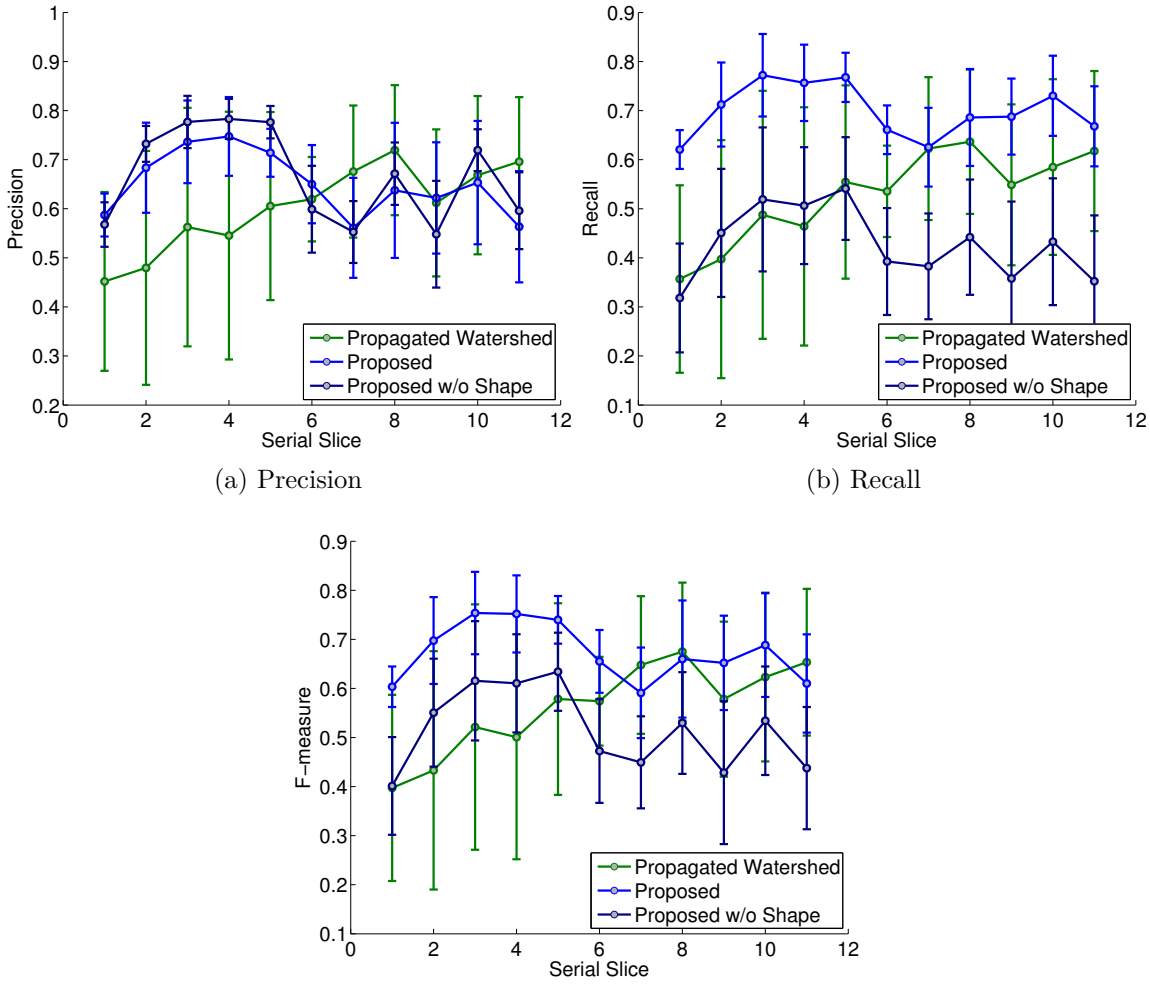
cut for this material since these images are edge images.

From Figure 1.13 (c), the proposed method scores better than all the comparison methods on almost every slice, regardless of selection of the initial slice. Similarly, the proposed method consistently obtains a close-to-accurate number of segments shown in Figure 1.13 (d), missing only few, typically very small, segments. Both watershed methods are less consistent in the number of missed (or erroneous) segments. Overall, the proposed method scores higher than 90% for both precision and recall, even after propagating to a slice that is 10 slices away from the initial slice, and shows much better performance than the propagated and baseline versions of watershed and edge-based normalized cut. Additionally, we show the proposed method with both the global method and local non-homomorphic strategy ("Proposed Global+Local") as well as only the global method ("Proposed Global") in Figure 1.13 (c), which shows the improvement gained by the local non-homomorphic strategy which, overall, increases the segmentation performance.

Without the local non-homomorphic strategy, a single propagation completes in roughly 5 minutes. With the local non-homomorphic strategy, the runtime for a single propagation is $\sim$15 minutes. In all cases, the run time is dominated by the graph-cut computation—all other computations we performed introduce negligible overhead (i.e., less than 15 seconds).

Dataset 2: Martensite Lath Structure in Steel

In this section, we segment 7 martensite lath segments (plus the background) in a sequence of 11, high-resolution (1600×1200) steel material slices [88] to evaluate the use of the shape preservation strategy introduced in Section 1.4. A subset of this material is shown in Figure 1.9 (b). The segments are characterized by consistent, often elongated shapes whose boundaries can, in some instances, be difficult even for a human segmenter to identify, as shown in Figure 1.15. The ground truth contains



(a) Precision

(b) Recall

(c) F-measure

Figure 1.14: **(a-c)** The segmentation precision, recall, and F-measures for the proposed method and the propagated watershed method on the 11 tested slices of Dataset 2. The curve labeled "Proposed w/o Shape" is the method discussed in Section 1.2 without the shape preservation strategy discussed in Section 1.4.

boundaries that are relatively wide (7 pixels) due to the resolution of the images and the uncertainty of some of the boundaries. Many segments are not salient or obscured by noise, and only the most salient segments—as identified by materials scientists—are included in the ground truth. Because non-propagated methods cannot discriminate among desired or undesired segments, we only evaluate the proposed method and the propagated watershed method. Note that this dataset is homomorphic ($\mathcal{NH} = 0$), and as such we do not display the cardinality difference measure,

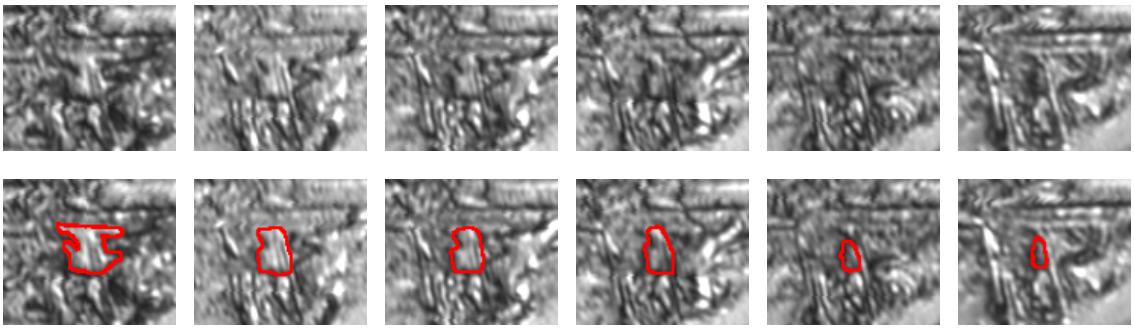


Figure 1.15: Illustration of the challenges faced in Dataset 2. Note specifically that the identified segment is not always clear in each slice, and the boundaries are often indistinct, even to a human segmenter (bottom row).

since both methods always obtain the same number of structures.

Since this dataset is made up of intensity images, we use Equation (1.4) in conjunction with Equation (1.3) as the binary term Φ_{pq} in the energy minimization for the proposed method. For the unary term Θ_p, we set the dilated size (see Section 1.2) to be 10px, and we additionally enforce skeleton propagation, eroding the skeleton by 5px, as discussed in Section 1.4. The background is treated as any other segment.

From the quantitative results shown in Figure 1.14, the proposed method without shape preservation from Section 1.4 ("Proposed w/o Shape") is competitive with the propagated watershed method. Further, the proposed method with the shape preservation from Section 1.4 ("Proposed") obtains the best performance on this dataset, in terms of both F-measure and cardinality difference as shown in Figure 1.14.

Figure 1.16 and Figure 1.17 show qualitative results from this dataset. Notice that the incorporation of shape is essential to segmenting this dataset—without the shape-preservation strategy, the proposed method cannot capture the cohesive shape details in the propagation. With only 7 segments to segment, the proposed method completes in seconds, with the runtime dominated by preprocessing steps rather than the graph-cut algorithm.

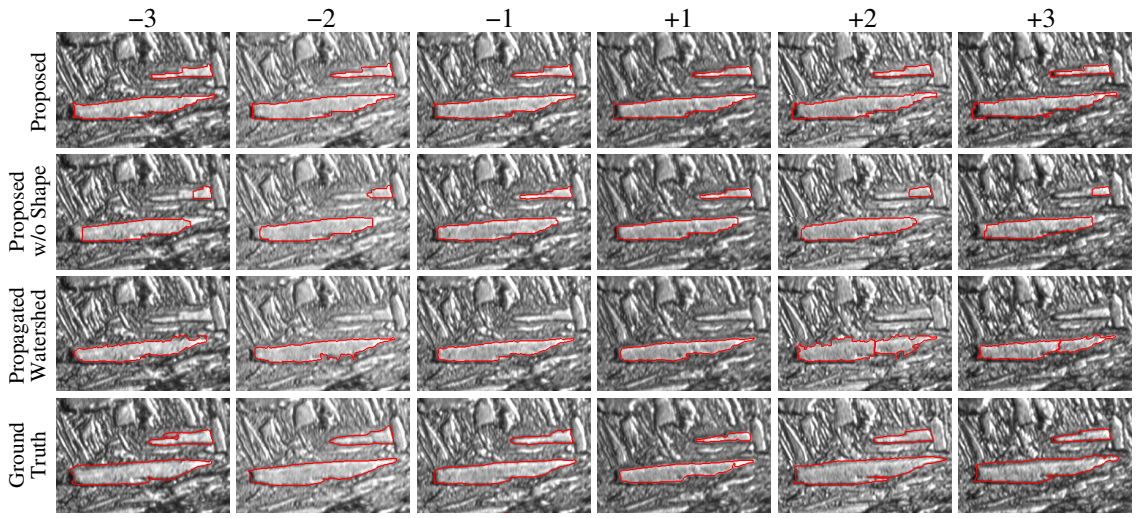


Figure 1.16: Segmentation results on two selected segments using the proposed method and propagated watershed, along with the ground truth on Dataset 2. Each column shows a slice of different distance from the initial slice.

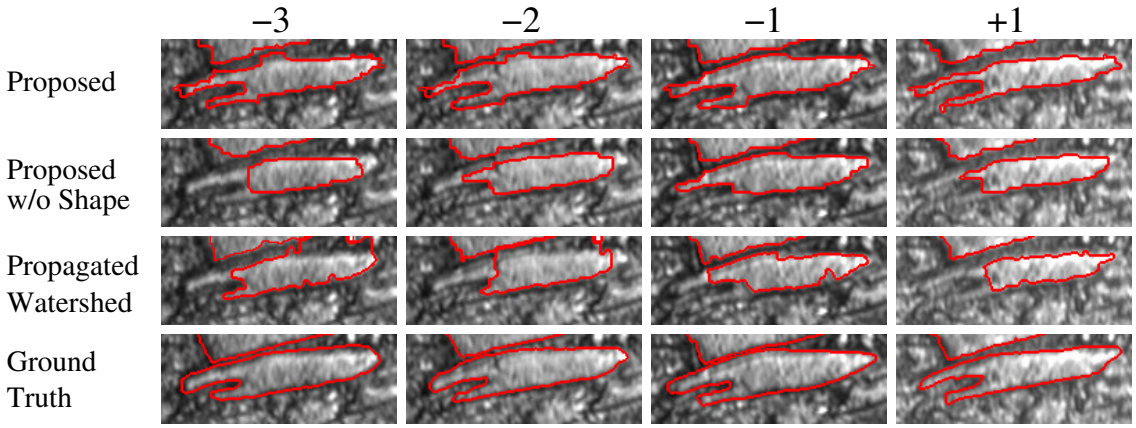
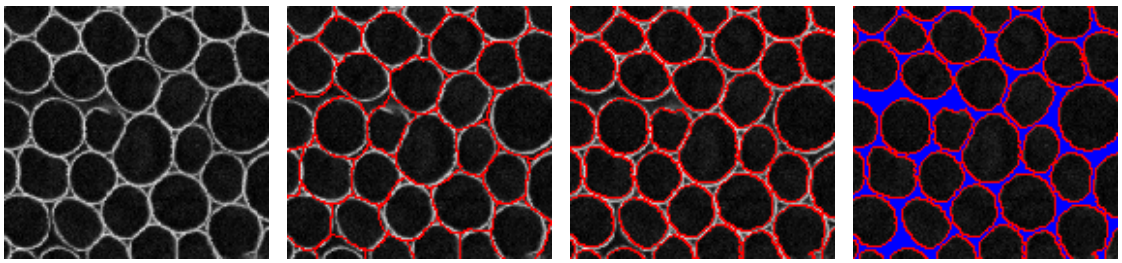


Figure 1.17: Segmentation results on a selected segment using the proposed method and propagated watershed, along with the ground truth on Dataset 2. Each column shows a slice of different distance from the initial slice.

Dataset 3: Cotyledon Plant Embryo

In this section, we test the proposed method on a sequence of 38 microscopic image slices with largely-consistent objects (i.e., few local non-homomorphic local regions) of cotylendon [112, 7, 126], imaged with confocal laser scanning microscopy, courtesy of JC Palauqui. Cotylendon is a plant embryo that is pivotal to the early development of many plants and the segments of interest in this dataset are cells, as shown



(a) (b) (c) (d)

Figure 1.18: **(a)** Magnified view of the cotylendon biomaterial in Dataset 3. **(b)** An unsatisfactory segmentation that incorrectly makes many cells adjacent to each other. **(c)** A correct segmentation of (a) where cells are not adjacent. **(d)** The background (blue).

in Figure 1.9 (c). We can see that, different from Dataset 1, cells in this dataset are generally not adjacent to each other and separated by a non-cell structure, which we call "background" (shown in blue in Figure 1.18 (d)) in this dataset.

As shown in Figure 1.18 (c), the desired segmentation of the cotylendon biomaterial has limited shared boundaries between cells, unlike the incorrect segmentation shown in Figure 1.18 (b). The human-supplied ground truth is very accurate (roughly 3px wide) to capture these minor details, but requires a significant amount of time to obtain per slice, as shown in Table 1.2. Note that this dataset represents a cohesive layer of cells extracted from a larger volume. The beginning and end of this layer exhibit strong confocal blurring, making it difficult to identify segments. Also, slice 23 exhibits an intensity variation that the surrounding slices do not, leading to poorer performance on this slice compared to others.

For the proposed method, we handle the background segment S_b^V separately. Specifically, we set Φ_{pq} as

$$\Phi_{pq}(S_i^V, S_j^V) = \begin{cases} 0, & i = j \\ \infty, & i \neq b \text{ or } j \neq b \\ g, & i = b \text{ and } j = b \end{cases} \tag{1.12}$$

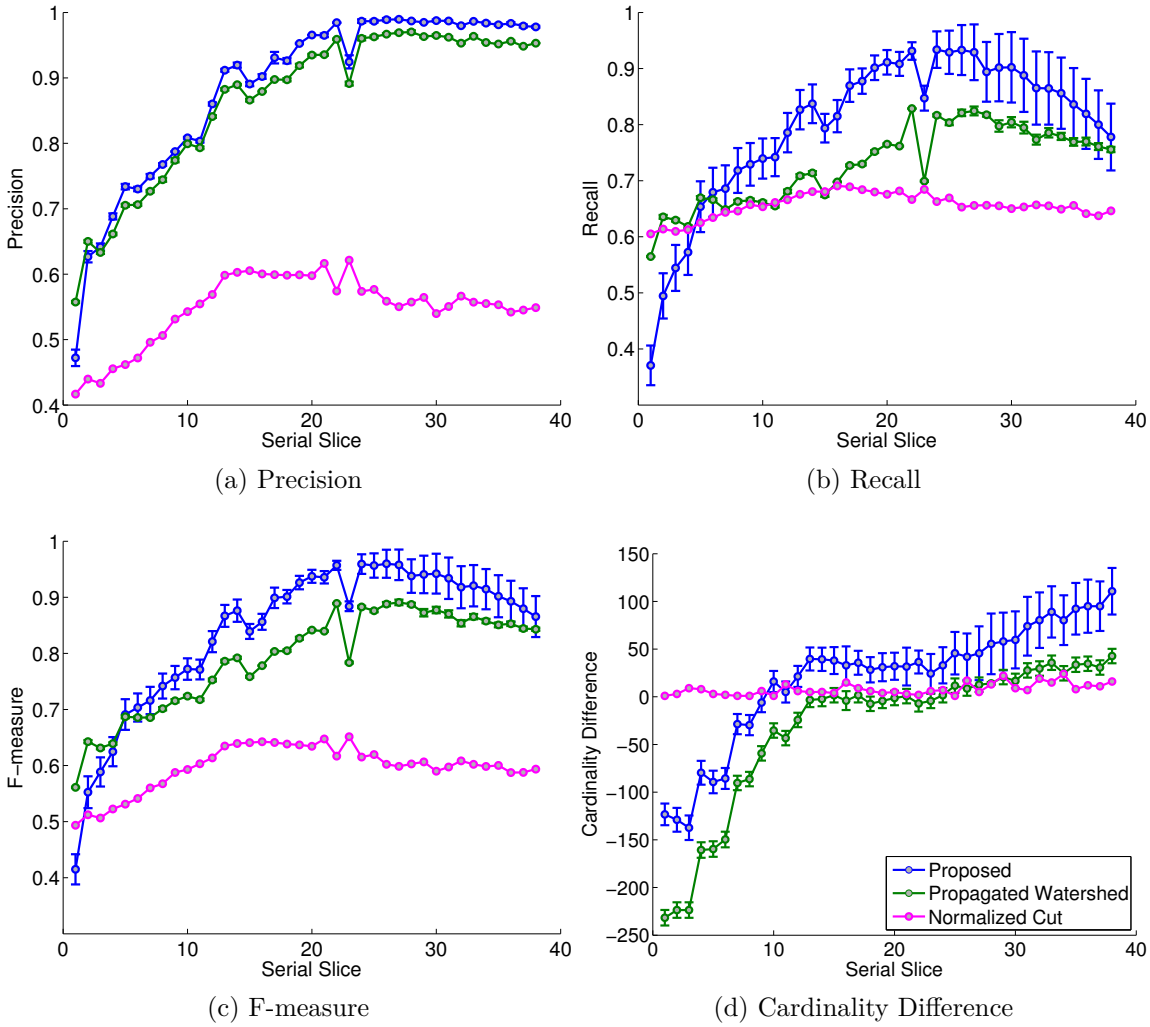
which is similar to Equation (1.3), but it specifically forces a fixed topology: all cells are adjacent to the background and no cells are adjacent to each other. For g, we use

Equation (1.5) since this dataset is an edge image. For the unary term Θ_p, we use the method outlined in Section 1.2, with a dilation of 10px.

For the comparison methods, it is nontrivial to enforce such prior-known adjacency relations. We found that the propagated watershed, matches the proposed method closely on precision but performs more poorly on recall. Inversely, the intensity-based normalized cut performs better on recall but more poorly on precision, as it is not a propagated method. As such, both comparison methods obtain a poorer F-measure compared to the proposed method, as shown in Figure 1.19. The performance of all methods falls off on the slices near the beginning and end of the sequence, as the cell boundaries become obscured by blurring. Qualitative results are shown in Figure 1.20, from which we can see that both comparison methods fail to capture the boundary details when two cells are near each other.

From Figure 1.19 (d), the proposed method consistently obtains close to the correct number of segments until it is propagated to the slices near the beginning of the sequence, where the uncertainty of the layer transition makes obtaining the exact number of segments difficult. Normalized cut–by design–obtains nearly the correct number of segments for all slices, as this is a required parameter of the algorithm, and as such we provide it with the ground truth number of slices. In contrast, the propagated watershed method routinely undersegments each slice; a phenomenon which is quite apparent in Figure 1.20.

Larger differences are apparent in the qualitative results in Figure 1.20, where the propagated watershed method tends to miss entire edges between cells when there are small gaps in the boundary's intensity. The proposed method tends to retain the separation between cells, while the propagated watershed method tends to merge these gaps with one of the adjacent cells. Overall, the proposed method matches the ground truth very well, while the comparison methods tend to miss the essential details between cells, as well as entire edges in certain instances.



(a) Precision (b) Recall

(c) F-measure (d) Cardinality Difference

Figure 1.19: **(a-c)** The segmentation precision, recall, and F-measures for the proposed method, the propagated watershed method, and normalized cut on Dataset 3. **(d)** Cardinality difference measure for all evaluated methods.

The proposed method requires less than an hour to propagate all ∼700 segments to a new slice, bounded by available system memory: propagating suc hlarge numbers of segments requires ∼8GB of memory per propagation.

Dataset 4: NanoEnergetic Epoxy

Finally, we apply our segmentation to a NanoEnergetic Epoxy volume consisting of ∼ 160 different segments across 12 high-resolution (1288 × 957) slices, as shown in

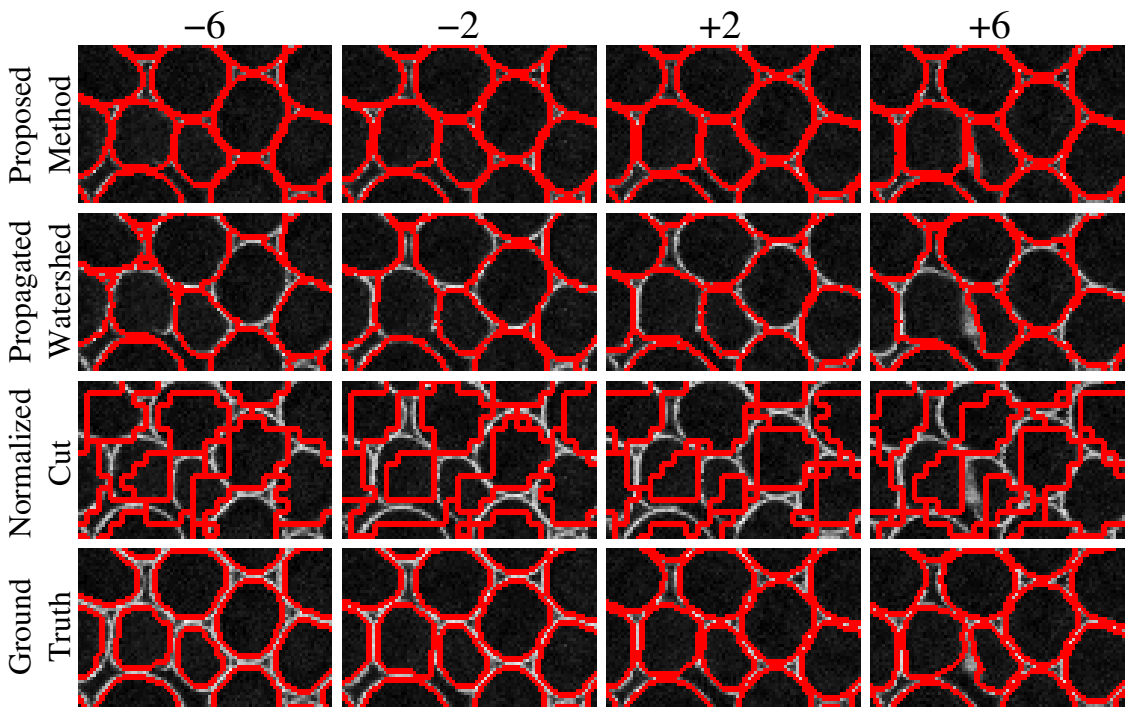
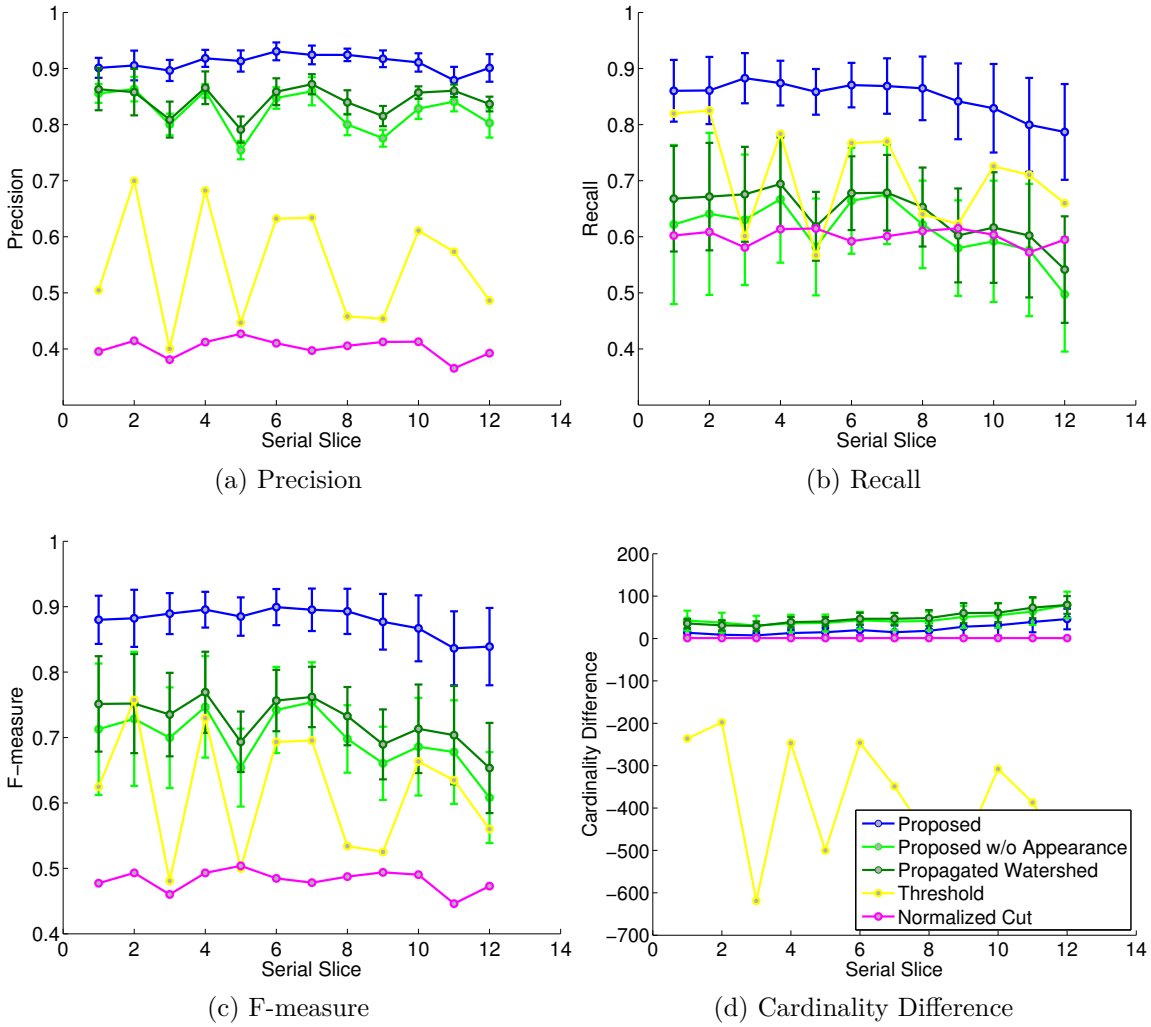


Figure 1.20: Zoomed view of segmentation results on selected slices using the proposed method, watershed, and normalized cut, along with the ground truth, on Dataset 3. Each column shows a slice of different distance from the initial slice.

Figure 1.9 (d). While there is local non-homomorphism, we show that the performance of our method without the strategy in Section 2.1 is still comparable to other methods. Intensity thresholding is typically applied to these images to separate segments of interest and the background. However, there are many very small segments that are not directly relevant to the segmentation process, as well as segments with less clear boundaries, which leaves room for improvement using the proposed method.

For comparison purposes, we use the proposed method, along with watershed, normalized cut, and image thresholding to segment this dataset. We use the method detailed in Section 1.5 where we use the negative log likelihood of the appearance for the Θ_p term. For the proposed method, as before, we use the method from Section 1.2, this time with a dilation size of 15px. In addition, we use the intensity preservation strategy detailed in Section 1.5 with a d_2 value of 50px around each segment and sensitivity of $\lambda = 2$.



(a) Precision (b) Recall

(c) F-measure (d) Cardinality Difference

Figure 1.21: **(a-c)** The segmentation precision, recall, and F-measures for the proposed method, the propagated watershed method, normalized cut, and image thresholding the 12 tested slices in Dataset 4. **(d)** Cardinality difference measure for all evaluated methods.

From Figure 1.21, the proposed method and propagated watershed are clearly competitive. The proposed method incorporating intensity from Section 1.5 ("Proposed") performs the best on average by a small margin. The proposed method using only the propagation from Section 1.2 ("Proposed w/o Intensity") also performs consistently, clearly competitive with propagated watershed. Notice that the performance of image thresholding is wildly inconsistent with regards to its F-measure from one slice to the next. From Figure 1.21 (d), we can also see that thresholding tends to

produces oversegmentation by including many tiny segments that are not segments of interest. Similarly, normalized cut performs better than image thresholding, but tends to cut through the background rather than isolating each individual segment. As discussed before, normalized cut achieves a near-to-zero cardinality difference because we provides it the ground truth number of segments. Sample qualitative results on this dataset are shown in Figure 1.22. In Figure 1.23, we showed a zoomed view of the segmentation to highlight the improvements produced by incorporating intensity preservation across slices.

The proposed method, with intensity preservation described in Section 1.5, requires an average of 10 minutes per frame for each propagation, with its runtime dominated by the graph-cut energy minimization algorithm.

Analysis

The proposed method is competitive on all included datasets. The incorporation of the additional strategies from Section 2.1, Section 1.4, and Section 1.5 each improve performance by varying degrees: specifically the local non-homomorphism strategy from Section 2.1 on Dataset 1, the shape consistency strategy from Section 1.4 on Dataset 2, and the appearance consistency strategy from Section 1.5 on Dataset 4. We also show the performance of the base global propagation method on Dataset 3.

Although the watershed method has been used to segment images with many grains or cells [63], the propagated watershed method's F-measure is lower than the proposed method on all datasets, as shown in Figure 1.13 (c), Figure 1.14, Figure 1.19 (c), and Figure 1.21 (c). We believe this is because the proposed method considers (non-)homomorphism in the propagation. Propagated watershed provides a low cardinality difference compared to the proposed method on Dataset 3, as shown in Figure 1.19 (d). The reason is that, on Dataset 3, we always keep markers for each segment when eroding a segment for marker construction (see Section 1.6), and the

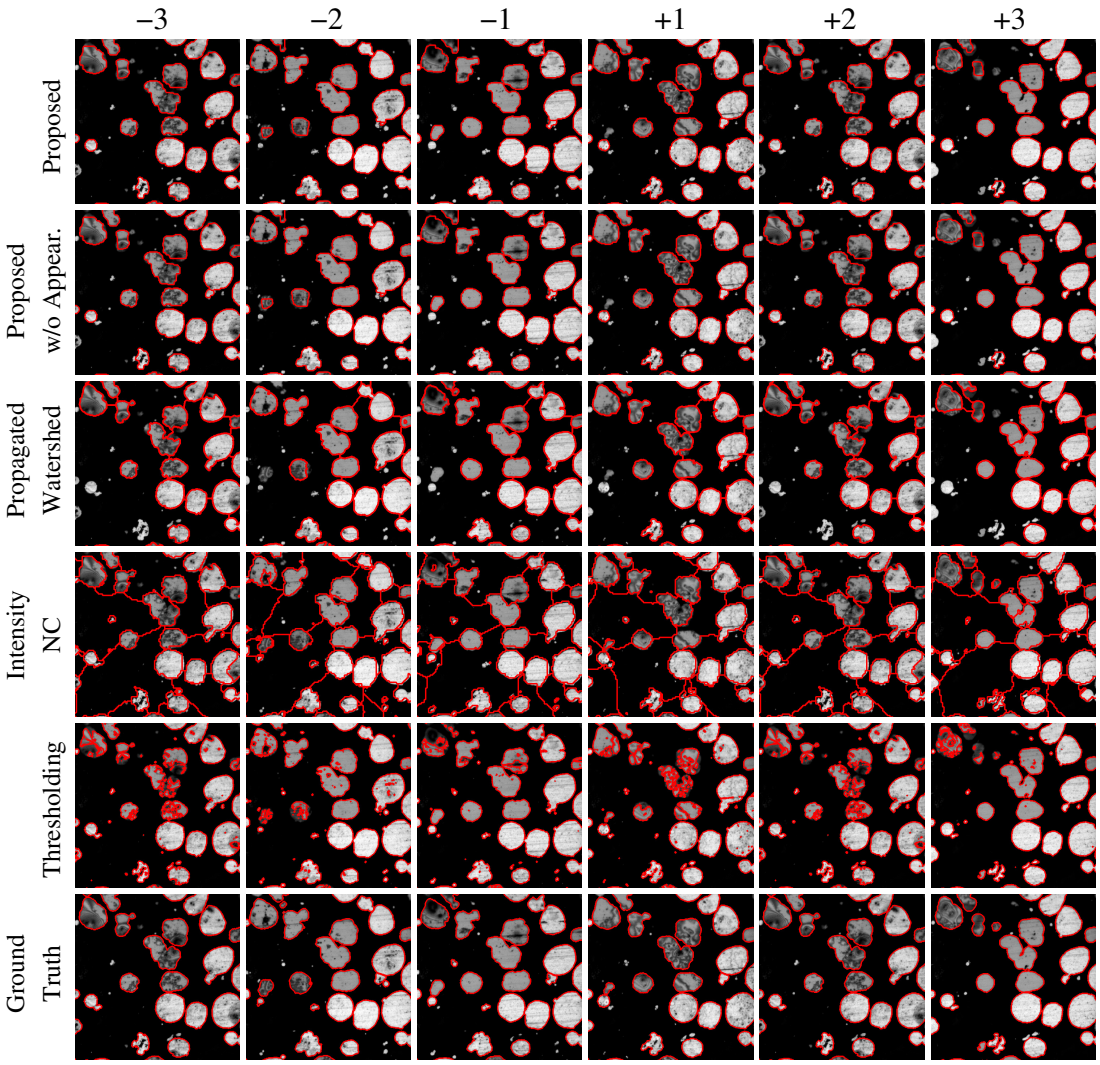


Figure 1.22: Segmentation results on selected segments using the proposed method, propagated watershed, intensity-based normalized cut, and image thresholding, along with the ground truth on Dataset 4. Each column shows a slice of different distance from the initial slice.

non-homomorphism score for this dataset is very low (0.0095).

Normalized cut performs even more poorly on all datesets. Though we supply normalized cut with the ground-truth number of segments, it fails to capture many of the more difficult boundaries, such as the gaps in Dataset 1, the dense boundaries of the nonadjacent cells in Dataset 2, and the background separating structures in Dataset 4, as shown in Figure 1.12, Figure 1.20, and Figure 1.22, respectively. Dovetailing with this, the thresholding method shown on Dataset 4 does achieve a

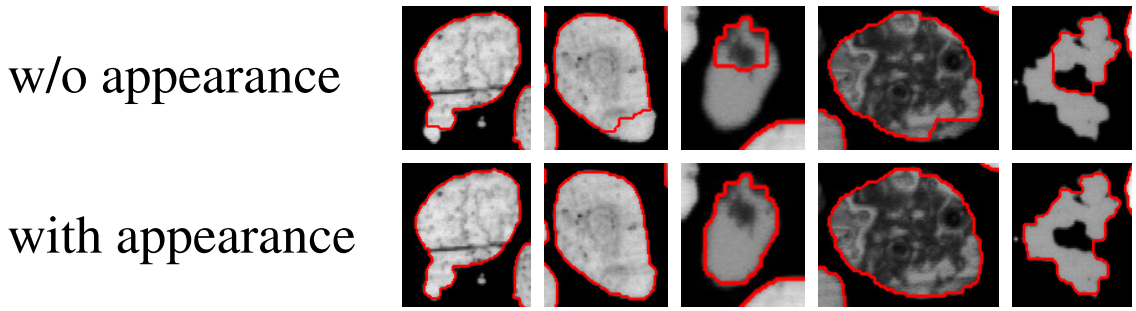


Figure 1.23: Comparison of the segmentation on Dataset 4 by using the proposed method without intensity preservation (Section 1.2 only) and the proposed method with intensity preservation (Section 1.2 and Section 1.5).

competitive F-measure (as do other methods on this dataset), however, it clearly oversegments the substructures, as shown in Figure 1.21 (d).

Overall, by integrating different unary and binary terms, a global and local non-homomorphic approach, and incorporating shape and appearance priors, the proposed approach is able to compete with a variety of different methods–both 2D and propagated–across a large variety of datasets.

1.7 Limitations and Future Work

From the formulation of Equation (1.3), the proposed method incorporates both the segmentation adjacency relationships from the previous slice ($\mathcal{A}^U$) and image information from the current slice ($g(p,q)$). In addition, we have shown that the data term Θ_p can incorporate spatial consistency (Section 1.2), shape (Section 1.4), and appearance (Section 1.5). Even with these priors, there are cases in specific datasets that are not fully handled by the proposed method.

An example from Dataset 1 is shown in Figure 1.24, where a small segment S_2 is not identified by the proposed method in Figure 1.24 (b) due to a lack of image information to suggest where this segment should be located. The desired segmentation is shown in the manually-created segmentation in Figure 1.24 (c), where physical knowledge and the locations of the surrounding segments require the presence of S_2.

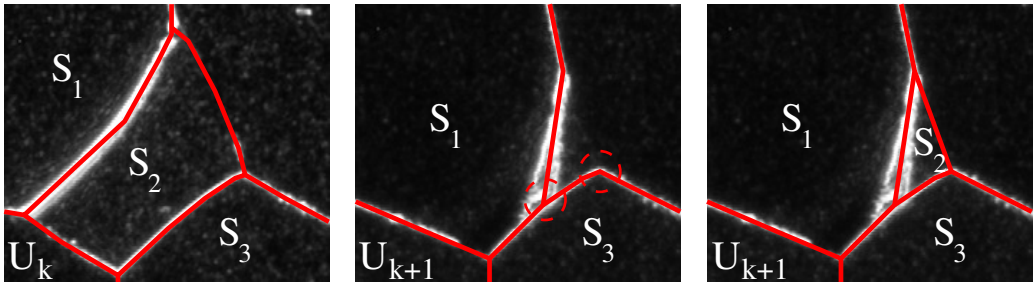


Figure 1.24: Example of the limitations of the proposed method. **(a)** Segmentation of slice U_k. **(b)** Segmentation of slice U_{k+1}, where segment S_2 is missed by the proposed method, due to limited image information to suggest its location. **(c)** Segmentation of U_{k+1} where segment S_2 is correctly located, as identified by a human.

The proposed method, however, does not know segment locations in the new slice prior to the propagation. Specifically, a true model of the structural energy for this particular dataset would not allow the local curvature shown in the dashed circles in Figure 1.24 (b). Such curvatures lead to there being a lower structural energy if segment S_2 were to remain. In our future work, we plan to investigate dataset-specific priors, either by building them into the energy functional or augmenting our local non-homomorphic approach from Section 2.1.

Another limitations of the proposed method is that the segmentation errors may accumulate after repeated propagation. For real applications, this can be addressed by using multiple initial slices distributed throughout a volume, as well as manual correction of each slice before further propagation. Our method also does not incorporate specific materials properties. In our future work, we plan to investigate dataset-specific priors, either by building them into the energy functional or augmenting our local non-homomorphic approach from Section 2.1.

Chapter 2

Topology-Preserving Propagation

2.1 Method

The method discussed in Chapter 1 is able to preserve topology non-adjacency, i.e.,

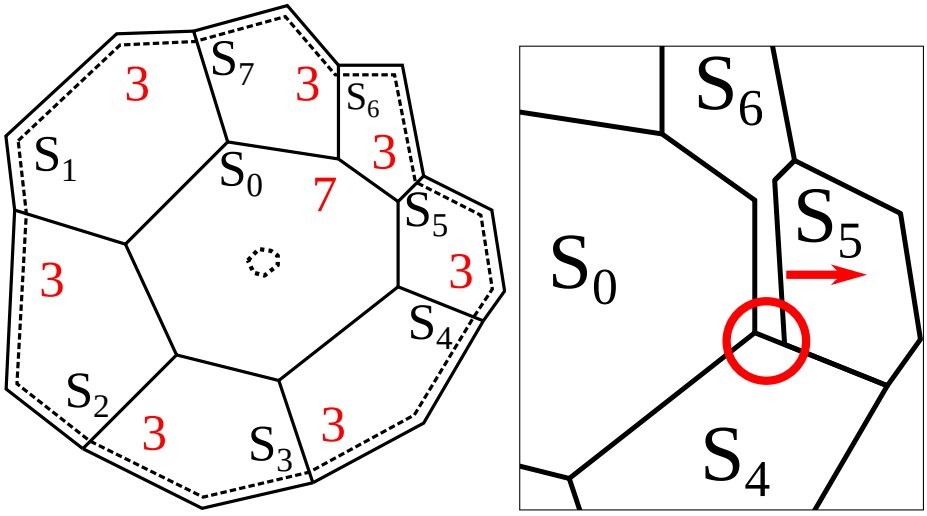
$$\{S_i^U, S_j^U\} \notin \mathcal{A}^U \Longrightarrow \{S_i^V, S_j^V\} \notin \mathcal{A}^V.$$

. However, it does not penalize the case where two adjacent segments in U become non-adjacent in V after propagation. In addition, it cannot guarantee the connectivity (intra-segment topology) of each segment in V. Therefore, it is not a segmentation propagation that enforces full topology preservation. In the following, we introduce a new approach to address this problem subject to the constraint that S^U shows local ring structures, as illustrated in Figure 0.4.

In this section, we propose an algorithm to propagate a segmentation by preserving full inter-segment topology, i.e., both adjacency and non-adjacency, which can be expressed as

$$\{S_i^U, S_j^U\} \in \mathcal{A}^U \iff \{S_i^V, S_j^V\} \in \mathcal{A}^V.$$

We achieve this by repeatedly updating the local ring structures sequentially. As illustrated in Figure 2.1 (a), using the segmentation S^U as the initial segmentation on V, we find a local ring structure that consists of one center segment and all segments adjacent to this center segment. The center segment is adjacent to every non-center segment in the ring, and is not adjacent to any segments outside of the ring. Additionally, in the general case, each non-center segment has a clockwise adjacent segment and a counterclockwise adjacent segment other than the center segment in



(a) (b)

Figure 2.1: Local ring structure example. **(a)** Preserving inter-segment topology by fixing the label of pixels along the ring boundary and in the center segment (dashed lines). Red numbers indicate the numbers of segments adjacent to the indicated segment. **(b)** Cropped view of (a) illustrating the preservation of inter-segment topology while updating the ring.

the ring. We also require the existence of at least one pair of non-adjacent segments in a ring to activate the infinity penalty as defined in Equation (1.3). This leads to the requirement that there be at least 4 non-center segments in a ring. If a ring contains only 2 or 3 non-center segments, as shown in Figure 2.2 (b)-(c), we can split one or two non-center segments along the radial direction, as shown in Figure 2.2 (d)-(e), to increase the number of non-center segments and introduce non-adjacency. This updates S^U which is then propagated to V, after which we merge such split segments together to obtain the final segmentation S^V. Another degenerate case is when there is a single non-center segment in the ring, as shown in Figure 2.2 (a), which reduces to the binary segmentation problem, and the method developed below can handle this degenerate case without splitting any segments.

From such a local ring structure, together with the image V on which this ring is embedded, we can define an MRF model (Equation (1.1)) specifically within this ring, which we minimize to obtain an updated segmentation. The primary issue is

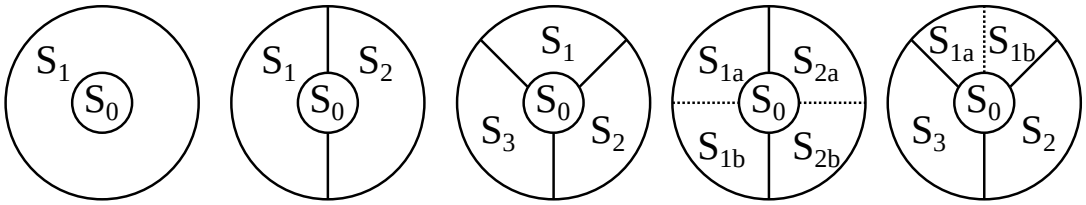


Figure 2.2: Illustration of the degenerate cases of ring structures. (a-c) Ring structures with 1, 2, and 3 non-center segments, respectively. (d-e) Non-center segment splitting to achieve four non-center segments for the rings in (b) and (c), respectively.

that we must preserve the topology of all the segments in S^U, not simply the topology inside this ring. Therefore, we fix the labels for all the pixels along the ring boundary, shown by the dashed contour in Figure 2.1 (a), e.g., the outer ring boundary shared with segment S_1 must still be labeled as S_1 after the update, etc. This can be easily achieved by assigning unary-term values for such pixels to be zero if their labels are the same as before and infinity otherwise. This way, we insure that the adjacency relations between any segment in this ring and any segment outside this ring will remain unchanged after the labeling update in this ring. To avoid the disappearance of the center segment, we also select the centroid pixel of the center segment and require its label to be unchanged (dashed lines in the center of Figure 2.1 (a)).

Based on this, we simply use the MRF model with the binary term as defined in Equation (1.3) within the image region defined by this ring to update its segmentation. As discussed above, this algorithm guarantees that non-adjacent segments remain non-adjacent. Which, together with the constraints defined on the ring boundary, will also guarantee that adjacent segments in this ring remain adjacent. This is indeed the case because, 1) the adjacency between non-center segments has been preserved by the label constraints on the ring boundary, and 2) the center segment is still adjacent to every non-center segment. For 2), it can be proved by contradiction, as shown in Figure 2.1 (b). If the center segment S_0 becomes non-adjacent from a non-center segment, S_5, a pair of non-adjacent segments, S_4 and S_6 (be reminded

that there are at least four non-center segments in a ring), must become adjacent to separate S_0 and S_5. However, the proposed algorithm has an infinity penalty term in Equation (1.3) specifically to prevent any non-adjacent segments from becoming adjacent.

It is well known that most algorithms, including graph cut algorithms, that are developed to optimize an MRF cost function may not guarantee the connectivity of each labeled segment. In this section, we show that this is not an issue for the proposed method. Our particular formulation, which uses 0 and ∞ for the unary term (see Equation (1.2)), guarantees connectivity of each labeled segment. This can be illustrated by the example shown in Figure 2.3. By using the Θ term as defined by Equation (1.2), a "band" region (bounded by the two dashed lines) of width $2d$ will be defined around the initial segmentation boundary (black line). Pixels in this band region can be labeled as segment 1 or 2, with 0 unary cost. After optimization (α-β swap), the new boundary (red line) separating these two labels must be located within this band region. Otherwise, this would introduce an ∞ unary cost. If any segment, say segment 2 in this example, is disconnected, such a disjoint fragment (red circle) must also be in this band region to avoid ∞ unary cost. However, such a fragment will not appear in practice since the α-β swap will automatically merge this fragment (red circle) into its containing segment 1 which leads to a smaller binary cost, and therefore a lower total MRF energy (unary cost is always 0 and does not change before or after the α-β swap).

With this formulation, we repeat the above steps for each local ring structure defined in S^U to update all segments. We may repeat the whole process for multiple rounds to continue updating the segment boundaries until they converge or until it reaches a pre-set maximum number of iterations. In practice we use a single iteration for all our experiments. Because we preserve the inter-segment topology and intra-segment topology of the entire segmentation while updating the segmentation in each

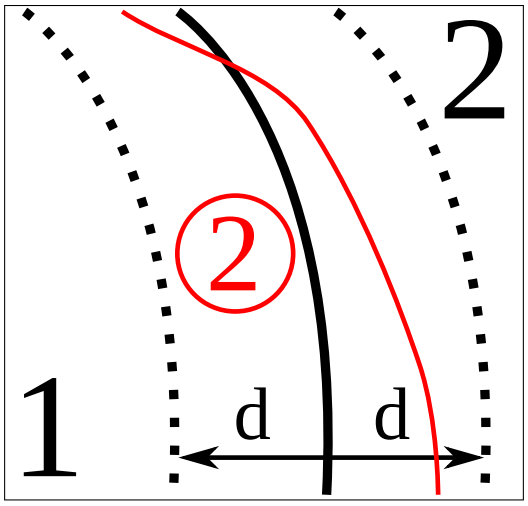
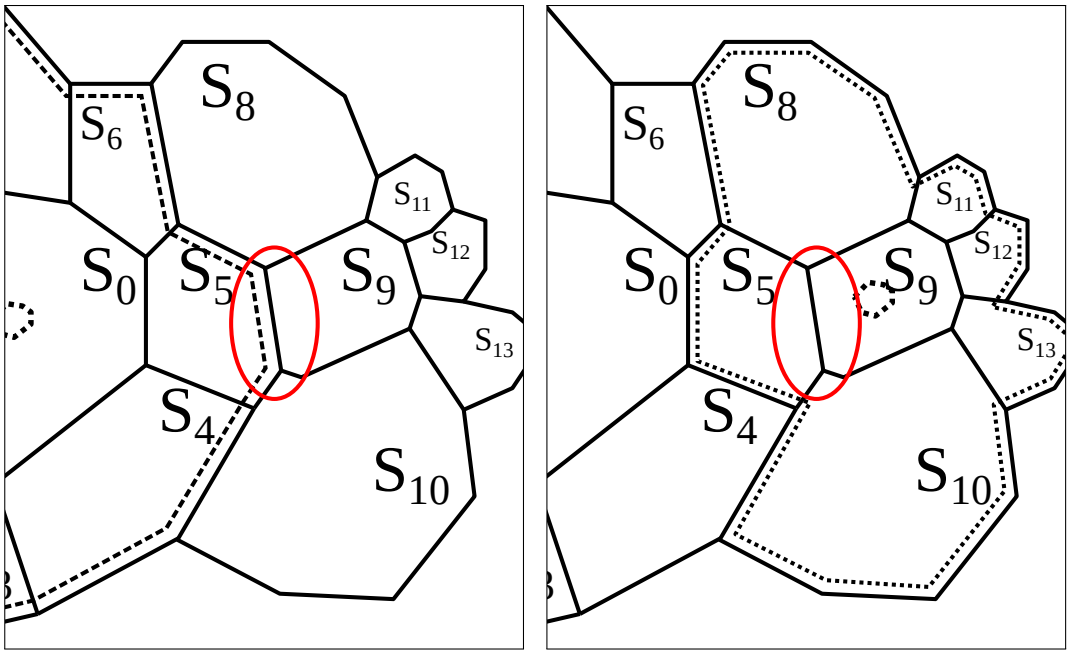


Figure 2.3: Illustration of segment connectivity in the proposed method.

local ring structure, it is easy to see that the complete iterative algorithm preserves both forms of topology at every step. As in the general multi-labeling problem, which is known to be NP-hard [118], our algorithm converges to a local minimum Note that although we fix the ring boundary and its pixel labels in each step of segmentation update, this ring boundary and its pixel labels will be updated when taking other segments as the center of the ring, as shown in Figure 2.4.

2.2 Experiments

In this section, we apply the proposed topology-preserving method to segment 3D electron microscopy metallic images to obtain their underlying grain structures, and show its application to interactive segmentation and stereo matching. For grain segmentation, we manually construct a grain segmentation on one slice as the initial template and then use the proposed method to propagate the segmentation to neighboring slices, repeating this propagation until all slices are segmented. In practice, the global grain topology is largely consistent from one slice to its neighbors, but local topology may change due the appearance of a new grain or disappearance of an existing grain. The previous chapter has shown that strictly enforcing the topol-



(a) (b)

Figure 2.4: Updating segment boundaries. **(a)** Highlighted edge (red) will be fixed when processing the ring centered at S_0. **(b)** The same edge will be updated when processing the ring centered at S_9.

ogy in segmentation, followed by a local relabeling to accommodate local topology changes, can combine both global topology consistency and local topology inconsistency, and leads to improved segmentation performance. As discussed previously, this cannot preserve the full intra-segment and inter-segment topology. In this experiment, we use the proposed method to propagate segmentation by enforcing full topology preservation, followed by the same local relabeling algorithm developed in Chapter 1, to segment the 3D metallic image volumes. We compare our performance with the algorithm developed in the previous chapter ("Waggoner 2012") and several other methods.

β-Ti grains in Ti-21S

First, we evaluate the proposed method on a sequence of 11 microscopic $750px \times 525px$ images consisting of β-Ti grains which are extracted from a Ti-21S titanium materials

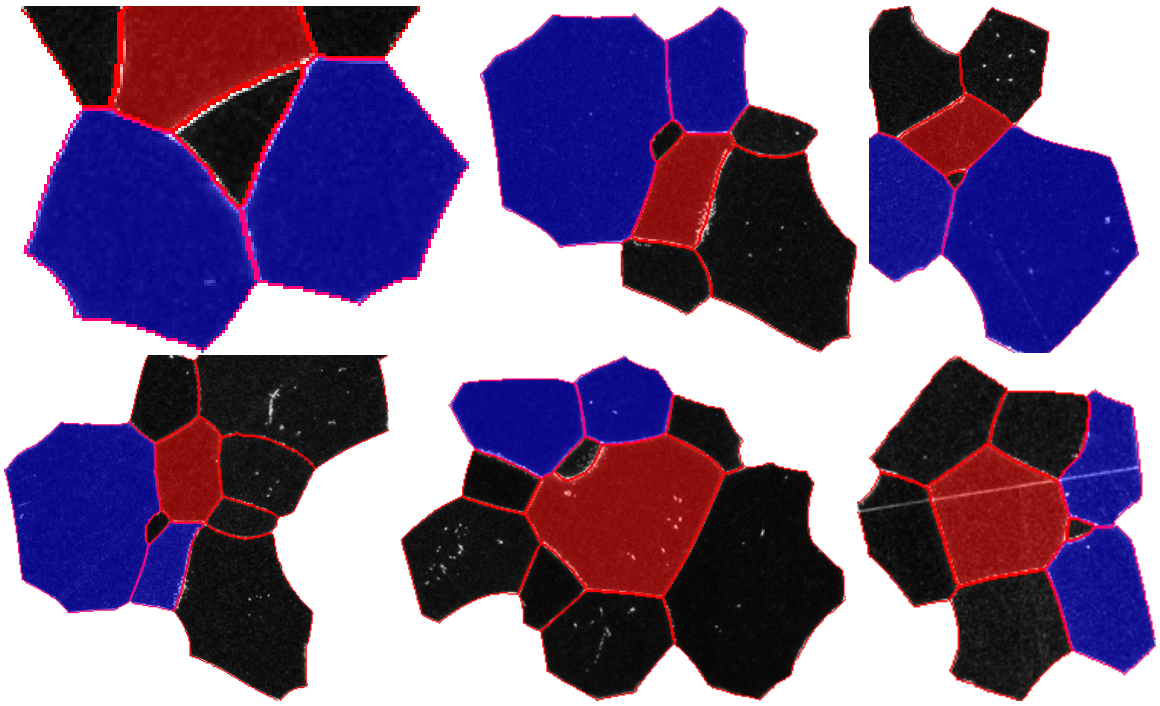


Figure 2.5: Sample non-ring local structures (center segment in red).

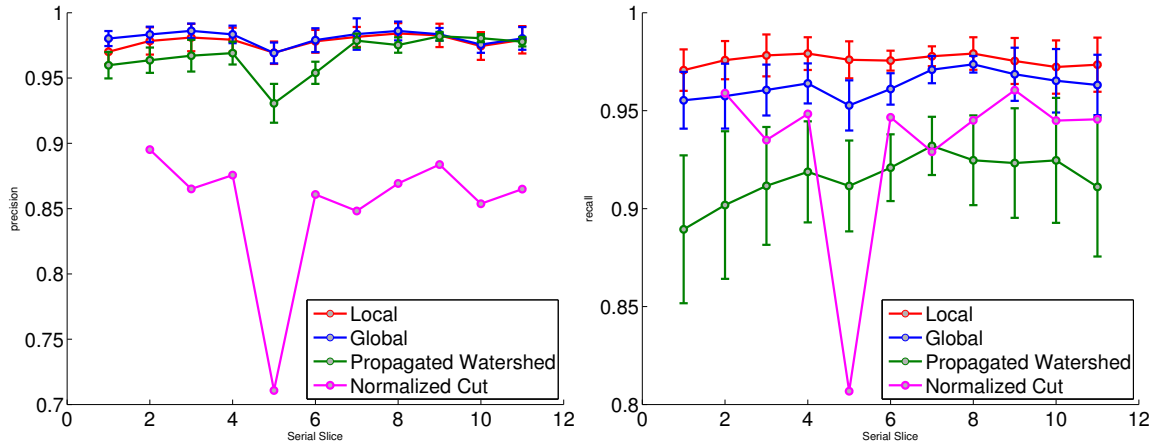
volume [89]. We additionally have a ground-truth for these slices, created by materials scientists, for objective performance evaluation.

We first validate that the 2D slices of such grain images contain many ring structures. On the ground-truth segmentation, we take each segment as the center, combined with all its adjacent segments, and check whether such a local structure is a ring. Experiments show that out of 5586 local structures across all slices, we have 5202 (93%) ring structures. Sample non-ring structures are shown in Figure 2.5. Such non-ring structures usually contain a very small non-center segment that does not reach the ring boundary and, for such non-ring structures, we simply apply the same algorithm described in Section 2.1 to update its segmentation. While the topology is not strictly preserved, the effect on the final performance is minimal since these cases are only a small percentage of the total updated local structures. Furthermore, these small structures are more likely to undergo local topology changes, and it is therefore less important or desirable to enforce a fixed topology in these instances.

We evaluate the proposed method by selecting one slice as the initial template and propagating its ground-truth segmentation to the other 10 slices repeatedly. In turn, we use each of the 11 slices as the initial template to segment the other 10 slices. This way, on each slice, we obtain 10 segmentation results (omitting the runs where the slice was used as the initial template). We compare each segmentation result with the corresponding ground-truth segmentation, find the edge pixel coincidence (within a $3px$ tolerance), and calculate the precision/recall and the combined F-measure. On each slice, we calculate the mean and standard deviation of the precision, recall, and F-measure over the 10 segmentation results, which are shown in Figure 2.6. We also measure the segment-number difference between the segmentation result and the ground truth, which we call the *cardinality difference*: positive values indicate undersegmentation, and negative values indicate oversegmentation. The cardinality difference on each slice (mean and standard deviation) are shown in Figure 2.6 (d). Finally, in Figure 2.7, we show the total MRF energy in the proposed algorithm converges, since the energy decreases after each ring structure is processed.

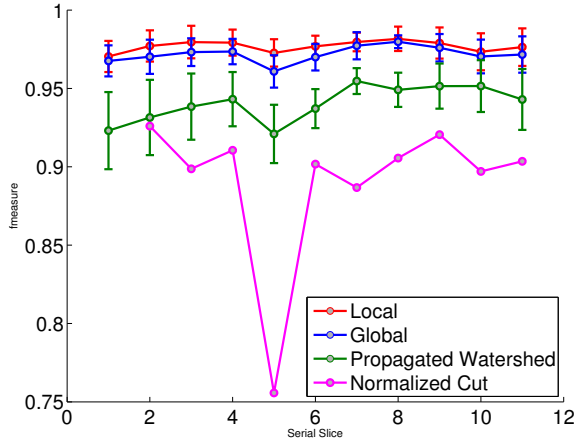
For comparison, we run the watershed method [121], using a Matlab implementation based on [69], and the normalized cut method [98], using a linear-time multiscale implementation based on [27], on the same 11 image slices. For watershed, we use a propagation strategy of the same form as in the proposed method, where we initialize the markers for watershed with an eroded version of the segmentation on the previous slice. For the normalized cut method, we provide the ground-truth number of segments for all slices. In addition, we compare with the method from the previous chapter, which attempts to preserve the topology by applying graph cut over the entire image. As mentioned above, for both proposed method and the previous method, we include an identical local-relabeling step to accommodate possible local topology changes resulting from newly appearing or disappearing grains between slices.

Figure 2.6 shows the results of this experiment. The proposed method is competi-

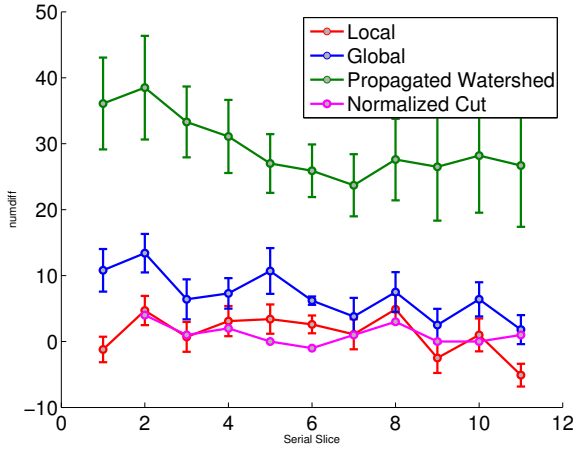


(a) Precision

(b) Recall



(c) F-measure



(d) Cardinality Difference

Figure 2.6: Performance of the proposed method, the previous topology-preserving image segmentation method, a propagated watershed method, and the normalized cut method, on the Ti-26S dataset.

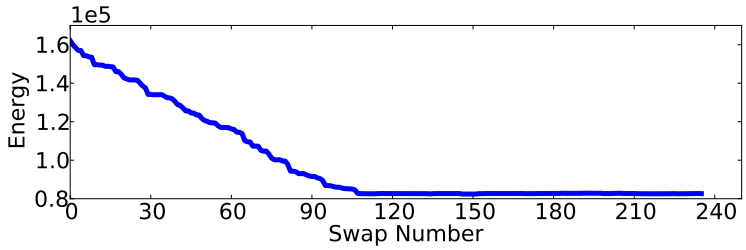


Figure 2.7: Total energy (y-axis) as each local ring structure is processed (x-axis) on a representative slice from the Ti-26S dataset.

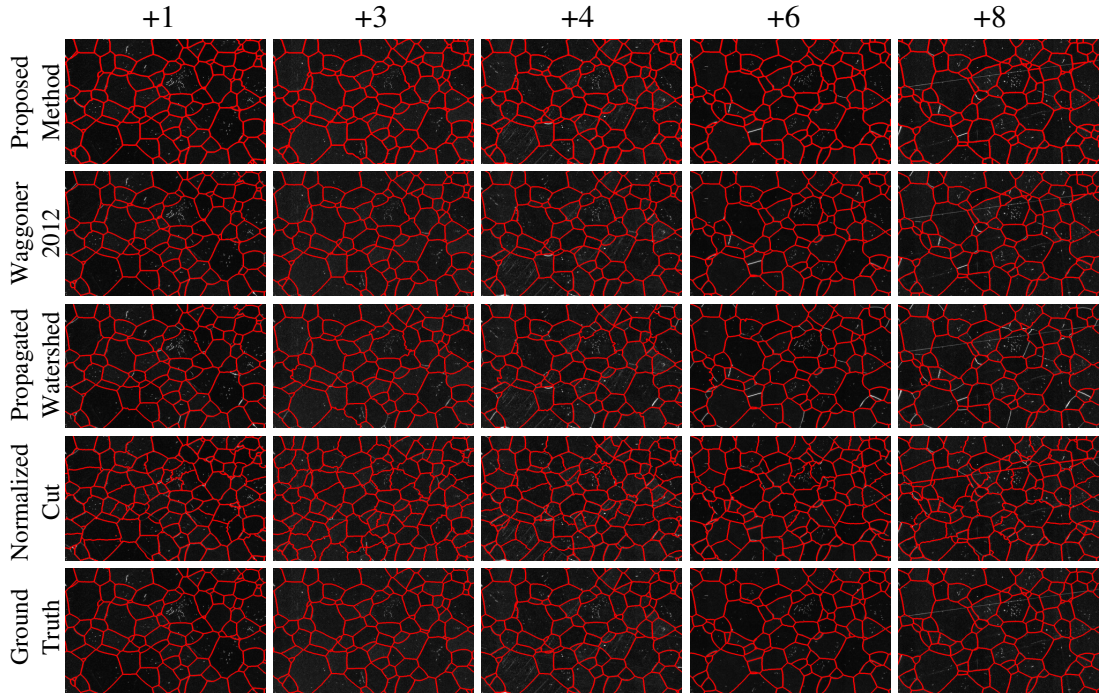
tive with the comparison methods as shown from the precision score in Figure 2.6 (a), and performs better than other methods in recall and the overall F-measure (Figure 2.6 (b) and Figure 2.6 (c)). Also of note, the proposed method obtains closer to the correct number of segments, shown by the cardinality difference measure in Figure 2.6 (d), where other competing methods tend to undersegment the slices. The normalized cut method is comparable to the proposed method for the cardinality difference measure, however, it is given the number of ground-truth slices as its input, so it is expected to obtain a near-ideal cardinality difference score.

Qualitative results are shown in Figure 2.8 (a). The proposed method is clearly competitive with the previous method and the watershed method. The watershed method tends to undersegment slightly, while normalized cut places the correct number of segments (since it is given this as input), but fails to find the correct segment boundaries in many instances. Because the proposed method and the previous method show similar performance in the high-level view in Figure 2.8 (a), we further show the more subtle differences between these two methods in Figure 2.8 (b). Notice that the proposed method correctly captures the grain boundaries more often, and captures them more accurately during the propagation, which leads to its better quantitative score in Figure 2.6.

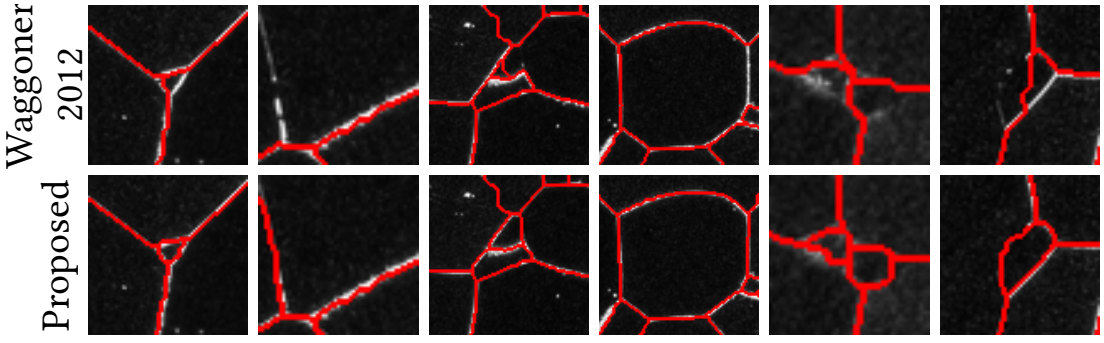
These experiments were conducted on a 2GHz Linux workstation with 8GB of memory. The proposed method is implemented in Python, with specific hotspots optimized in C. The proposed method requires less than 5 minutes per propagation, while the relabeling process from the previous chapter, implemented in C++, adds an additional 8 minutes.

DREAM3D Volumes

Since we have only a limited number of slices for the Ti-21S material, we expand our evaluation by including synthetic volumes created using DREAM3D [48]. DREAM3D

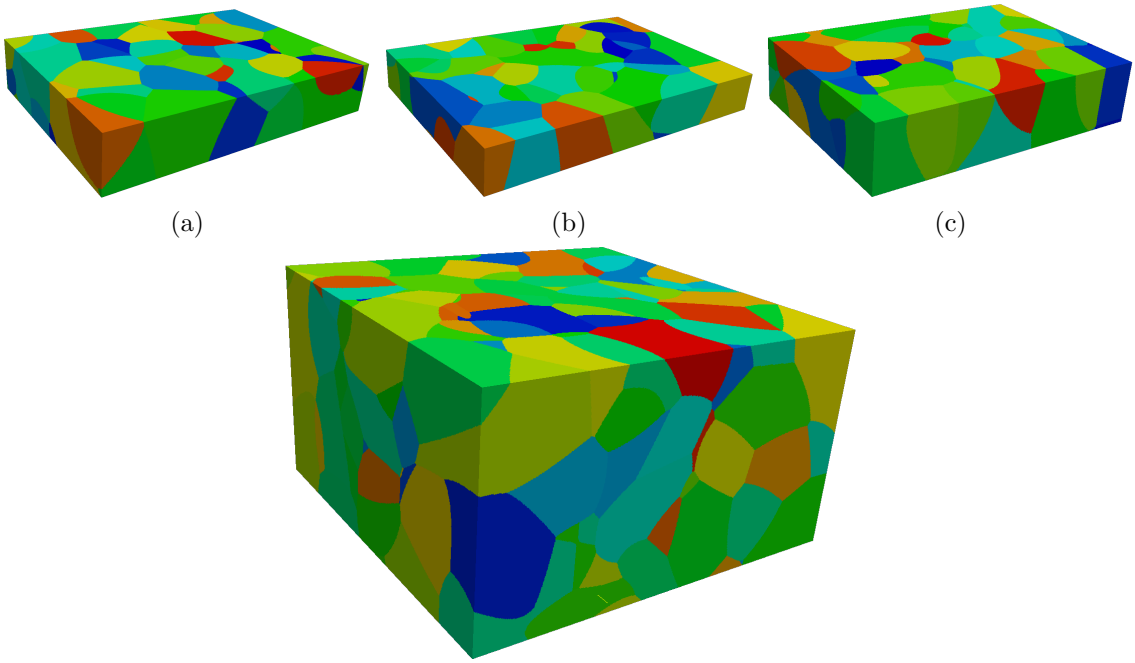


(a)



(b)

Figure 2.8: **(a)** Qualitative results for the Ti-26S dataset for the proposed method, the previous method, the propagated watershed method, and normalized cut. The distance from the initial template is shown by the numbers along the top. **(b)** The more subtle differences between the proposed method and the previous method.



(a) (b) (c)

(d)

Figure 2.9: Synthetic volumes generated by DREAM3D [48] to evaluate the proposed topology-preserving approach.

(Digital Representation Environment for Analyzing Microstructure in 3D) is a tool created by materials scientists specifically to aid in the creation, reconstruction, and analysis of 3D materials volumes, and provides a variety of tools, including a synthetic volume builder which we use to create realistic, physically-based synthetic materials volumes for our evaluation. The four synthetic volumes generated for our experiment are shown in Figure 2.9 with detailed information shown in Table 2.1.

Table 2.1: Dimensions of the synthetic datasets, and their imaged resolution on which the noise model is applied.

Vol	Dimensions	Imaged Size (voxels)
(a)	22.5 µm × 15.75 µm × 4 µm	750 × 525 × 50
(b)	22.5 µm × 15.75 µm × 3 µm	750 × 525 × 50
(c)	22.5 µm × 15.75 µm × 5 µm	750 × 525 × 50
(d)	37.5 µm × 36.75 µm × 10.5 µm	750 × 525 × 150

In generating these synthetic datasets, we add simulated random noise in the grain and along the grain boundary, and random scratches within the grain, and

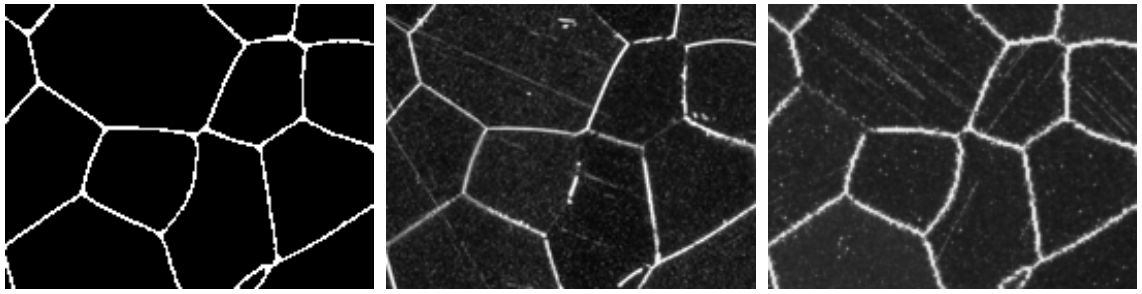
across entire slices, as shown in Figure 2.10. We generate the DREAM3D volumes with roughly the same inter-slice spacing as in the Ti-26S dataset so we can use all the same algorithm settings.

DREAM3D's synthetic volumes consist of discrete, noise-free evently-spaced labeled voxels, which we use as the ground truth for these volumes. For evaluation, we slice the 3D volume into discrete slices, extract a multivariate noise model from a real material (the Ti-21S material discussed in the previous section), and apply this noise model to the DREAM3D-generated slices. Specifically, we parameterize the

- grain interior intensity
- edge intensity
- grain interior noise
- edge noise
- scratches within grains
- scratches across the serial-section plane

gathered from the real material. We then sample from this noise model for each slice, grain, and edge. By doing this, we can create volumes that closely resemble real datasets, and imaging conditions that reflect the same imaging conditions in the Ti-26S dataset from the previous section. For comparision, we use the derived noise model to reconstruct a sample from the Ti-26S dataset using its own ground truth, which is shown in Figure 2.10. We generate the DREAM3D volumes with roughly the same spacing between serial section slices as the Ti-26S dataset so we can use all the same algorithm settings.

From Figure 2.11, the proposed method again performs better overall on the DREAM3D-created volumes compared to other comparison methods. Particularly, for image volume (d), a single template segmentation is propagated as far as 150 slices and the proposed method is still able to achieve an F-measure of 0.85.



(a) Ground Truth (b) Real Image (c) Applied Noise Model

Figure 2.10: Comparison of a real slice and the noise model applied to the ground truth.

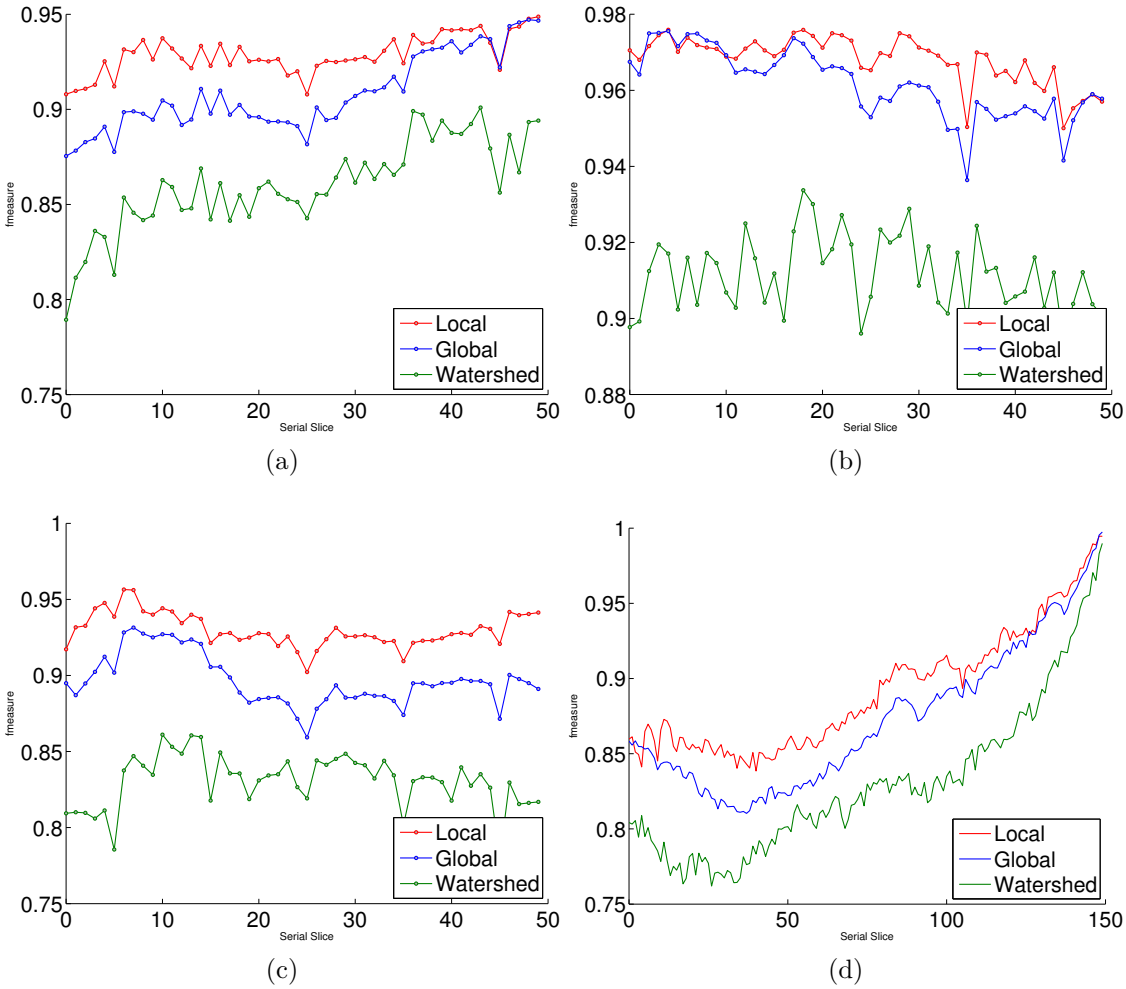


Figure 2.11: Performance on the DREAM3D generated datasets; (a)-(d) correspond with Table 2.1 and Figure 2.9, (a)-(c) show the average F-measure by taking different slices as the initial template, and (d) shows the F-measure from a single round of propagation starting with slice 150 as the initial template.

Interactive and Stereo Applications

To illustrate the general usefulness of this approach, we use the proposed method to segment a selection of natural and biological images. We use an interactive segmentation approach where the user annotates each object in the image with 1-2 clicks, and we infer the inter-segment topology from the Voronoi tessellation of the selected points, allowing the user to refine the points until the inferred topology is correct. This can be taken as the initialization for segmentation using the proposed method. Final segmentations are shown in the first three rows of Figure 2.12.

Similar to the propagation approach discussed previously, we can use the proposed method to propagate a segmentation between two images collected for stereo matching. As before, we provide a segmentation of one of the images, and use this as the initialization to segment the other image. Results are shown in the last two rows of Figure 2.12.

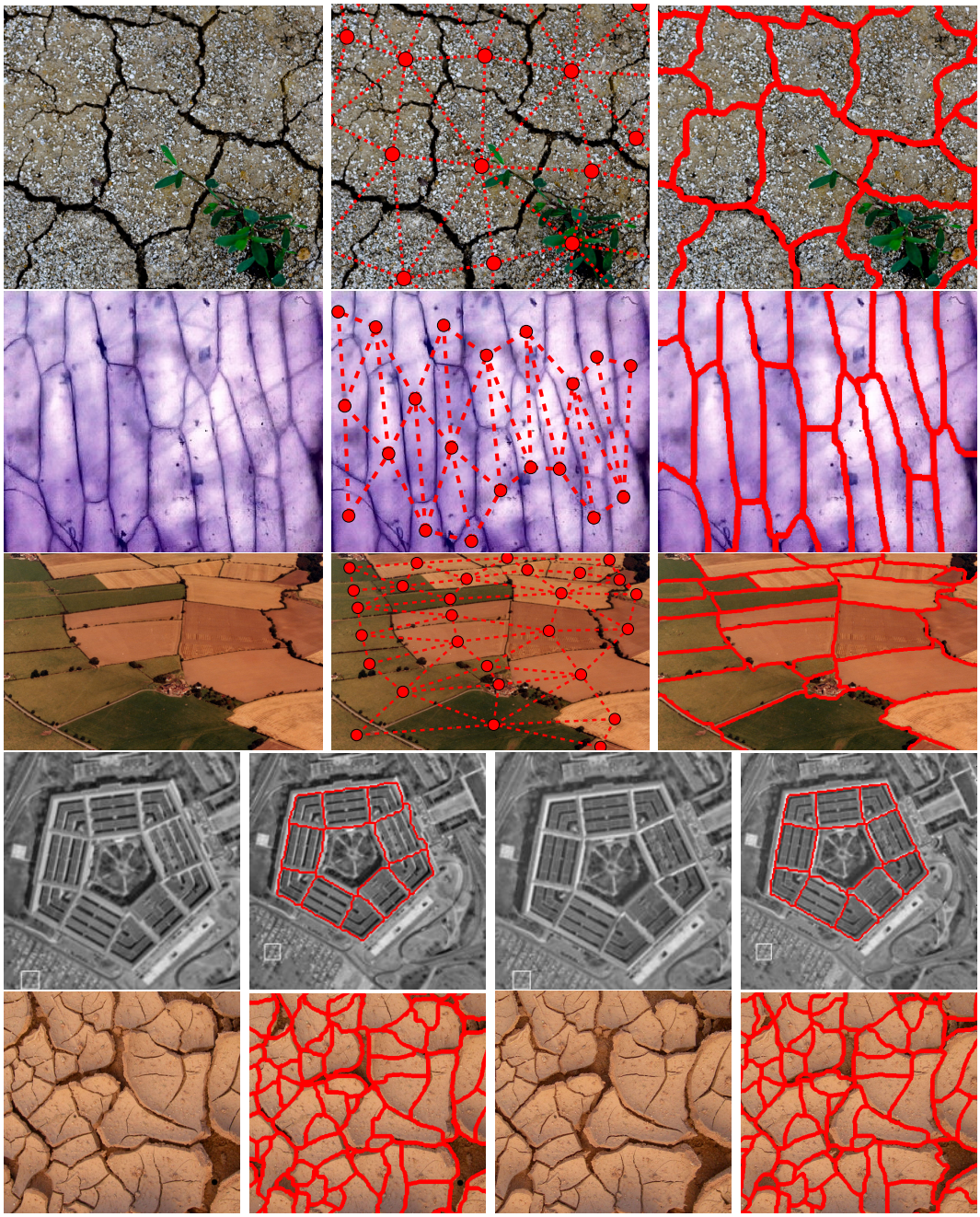


Figure 2.12: Qualitative segmentation results on natural and biological images. All images obtained from Flickr or Google Images. **(Rows 1-3)** Interactive segmentation on cracked dirt, onion cells, and aerial photography, respectively. The first column shows the original image, the second column displays the region adjacency graph derived from the Voronoi tessellation of the selected points, and the third column shows the resulting segmentation. **(Rows 4-5)** Segmentation propagation between two images collected for stereo matching, of aerial phtography and cracked ground tiles, respectively. The left two columns show the first image and its segmentation, and the right two columns show the second image and its resulting segmentation which is propagated from the first image.

Chapter 3

Interactive Materials Image Segmentation

3.1 Method

In Chapter 1 we developed a 3D materials science image segmentation method by propagating segmentation S^U of a slice U to a neighboring slice V, resulting in a segmentation S^V. This way, using an initial segmentation on one slice, we can repeatedly propagate this segmentation to the remaining slices in the volume to obtain a complete 3D segmentation. This propagation was done while preserving the topology (i.e., non-adjacency relations among 2D segments), which led to a better performance when compared with methods that did not incorporate topology as a prior. As before,

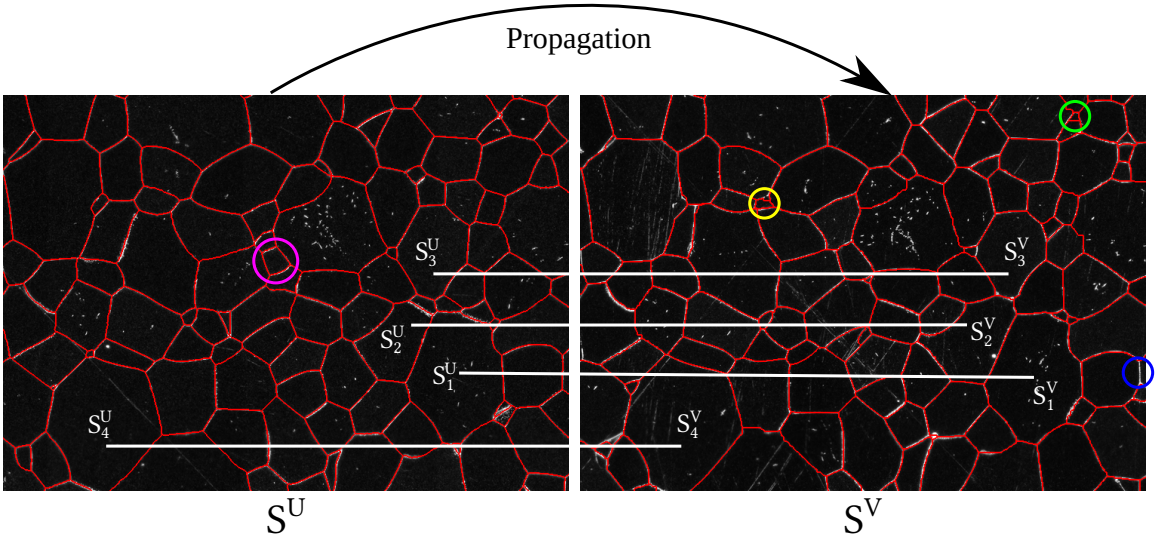


Figure 3.1: Example of segmentation propagation, highlighting different types of topology changes. Further discussion in the text.

we let

$$\Theta_p(S_i^V) = \begin{cases} 0, & \text{distance}(p, S_i^U) < d \\ \infty, & \text{otherwise} \end{cases}$$

where d is a dilation distance that reflects the maximum possible structural change between U and V, and

$$\Phi_{pq}(S_i^V, S_j^V) = \begin{cases} 0, & i = j \\ \infty, & \{S_i^U, S_j^U\} \notin \mathcal{A}^U \\ g(p,q), & \{S_i^U, S_j^U\} \in \mathcal{A}^U \end{cases} .$$

An example is shown in Figure 3.1, where S_1^V and S_2^V are allowed to be adjacent because S_1^U and S_2^U are adjacent in S^U. However, S_1^V and S_4^V are not allowed to be adjacent (have an infinity penalty) because S_1^U and S_4^U are not adjacent in S^U. This topology constraint was found to be particularly important for materials images, and our proposed method was able to outperform other methods that did not incorporate such a prior.

However, one phenomenon that was observed in Chapter 1 was that, during propagation, 2D structure topology between U and V might not always be fully consistent. For example, a new 3D structure with no intersection in slice U might appear in slice V, e.g., the structure in the yellow circle in Figure 3.1. Similarly, a 3D structure intersected by slice U might disappear in slice V, such as the structure circled in magenta in Figure 3.1. This breaks the topology constraints given in Equation (1.3) in some local regions. This may lead to spurious segments and missing structures, circled in green and blue respectively, in Figure 3.1.

Chapter 1 made use of a "non-homomorphic" brute-force automated search to locate such spurious and missing structures in V. However, particularly when the inter-slice distance is too large, it is not possible to examine every location for possible spurious or missing structures. In this chapter, our goal is to develop effective interactive tools to allow a user to conveniently specify the local areas that contain

spurious or missing structures, and incorporate such interactions to refine the segmentation S^V to a corrected $\tilde{S}^V$ on slice V, using the same energy minimization algorithm. More specifically, we propose to allow the user to correct these two types of segmentation errors within this framework by: a) annotating the location of a new segment to handle cases where a new structure appears in slice V, and b) annotation of existing segments that should no longer be present in segmentation S^V.

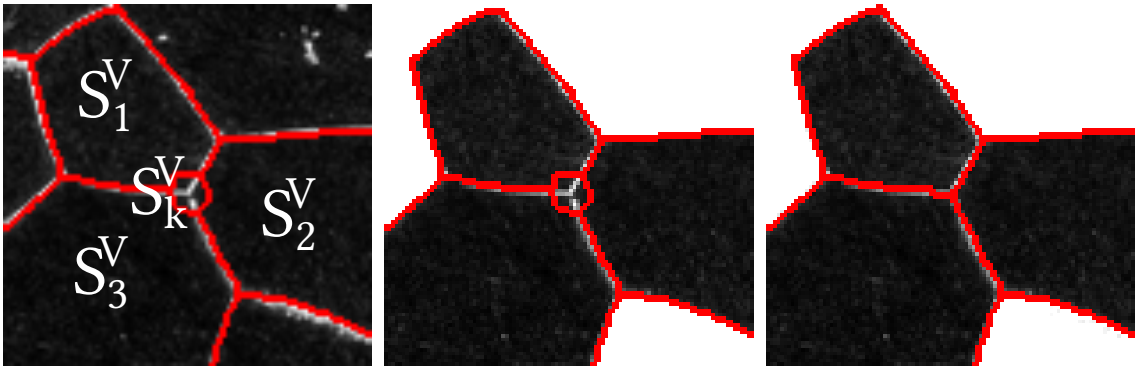
These interactions are inherently local because the 2D cross section of a 3D structure shows very small size before appearing or disappearing from a neighboring 2D slice. Therefore, correcting S^V to $\tilde{S}^V$ can be achieved by using the same energy minimization in local image areas around the interactive annotations. This is also important because interactive segmentation requires instantaneous user feedback. The previous propagation method segmented entire slices, which was more computationally intensive than is desirable in an interactive system. We will elaborate on these two interactions, and how we identify local regions for each, in the following subsections.

We further introduce three strategies aimed at making the proposed interactive tools more simple and efficient. Specifically, we discuss 1) a repropagation method in Section 3.2 that propagates annotations to neighboring slices to reduce the need to annotate every single slice, 2) a parameter estimation method in Section 3.3 that minimizes the need to tune parameters when adding new segments, and 3) a salient region detection approach in Section 3.4 that identifies key areas to be reviewed for errors to prevent laboriously inspecting large images to find areas that require interaction.

Removal of Spurious Segments

For this interaction, we allow the user to select a spurious segment S_k^V for removal by clicking the mouse on this segment in a visualized segmentation of S^V. Instead

of naively removing this segment by arbitrarily merging it into one of its neighbors, we use the same energy minimization discussed above to assign the individual pixels in S_k^V to potentially different neighboring segments. As discussed above, we identify a local region in which we update the segmentation. Specifically, this local region consists of the specified S_k^V and its neighboring segments, e.g., S_1^V, S_2^V, S_3^V surrounding the spurious segment S_k^V in Figure 3.2 (a), and re-run the energy minimization within this local region after modifying the Θ term to incorporate the interaction, resulting in an updated segmentation in this local region, as shown by the example in Figure 3.2 (b). For ease of notation, we use similar notation to the adjacency defi-



(a) (b) (c)

Figure 3.2: Example selection of a spurious segment S_k^V for removal. **(a)** Chosen S_k^V and surrounding segments. **(b)** Local region extracted around S_k^V. **(c)** The updated segmentation in the extracted local region.

nition in Equation (1.3) by using $\{\mathcal{A}^V\}_k$ to refer to the set of segments neighboring the segment S_k^V. This way, the local region for updating the segmentation is

$$\mathcal{L} = \{\mathcal{A}^V\}_k \bigcup S_k^V. \tag{3.1}$$

In this local region, we rerun the energy minimization of Equation (1.1) by modifying the Θ term. In particular, we do not allow any pixel to be assigned to S_k^V since this segment is to be removed. Instead, the pixels initially in S_k^V can be assigned to

Algorithm 2 Interactively specifying segment to remove.

function RemoveSegment(S^V, S_k^V)
  $\mathcal{L} \leftarrow \{\mathcal{A}^V\}_k \bigcup S_k^V$
  $\forall p \in \mathcal{L}$, build graph for energy minimization problem from Chapter 1
  $\Theta \leftarrow$ set from Equation (3.2)
  $\tilde{S}^V \leftarrow S^V$ incorporating the updated segmentation in $\mathcal{L}$
  return updated $\tilde{S}^V$

any of the segments in $\{\mathcal{A}^V\}_k$ with 0 cost for the Θ term, i.e.,

$$\begin{aligned} \forall p \in S_k^V, \quad \Theta_p(\tilde{S}_i^V) &= \begin{cases} 0, & S_i^V \in \{\mathcal{A}^V\}_k \\ \infty, & \text{otherwise} \end{cases} \\ \forall p \notin S_k^V, \quad \Theta_p(\tilde{S}_i^V) &= \Theta_p(S_i^V) \end{aligned} \tag{3.2}$$

By updating Θ in this fashion, we do not require the pixels previously in S_k^V merged into a single neighboring segment. Instead, these pixels may be assigned to more than one segment in $\{\mathcal{A}^V\}_k$, as shown in Figure 3.2 (b).

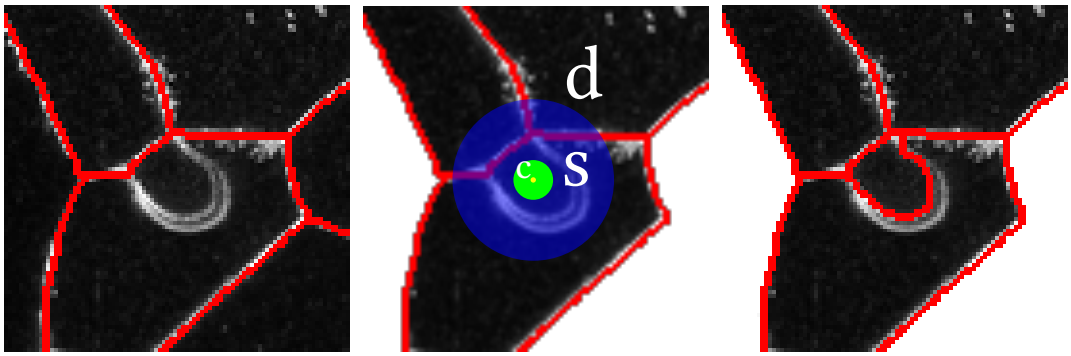
Note that this interaction is very simple and convenient, as it requires only a single click anywhere inside the spurious segment S_k^V. The full algorithm for removing spurious segments is summarized in Algorithm 2.

Addition of Missing Segments

Unlike removal, interactively annotating an additional structure cannot be solely formulated as a simple modification of the Θ term in the energy minimization formulation. This is because the multi-labeling problem used to optimize the energy minimization form in Equation (1.1) optimizes over a fixed set of segments, and cannot introduce new segments. Thus, for each missing segment, we must explicitly create a new segment at the location interactively specified by the user.

Based on the initial segmentation $S^V = \{S_1^V, S_2^V, \ldots, S_n^V\}$, we take as input from the user an annotation specifying the center location c of the new segment $\tilde{S}_{n+1}^V$. In addition to this, we also accept two parameters from the user: 1) the *seed* radius

s specifying a circular region around c such that this circular region is completely contained within the missing structure; 2) a *dilation* radius d, which is similar to the dilation parameter used in Chapter 1, such that the circular region with this dilation radius d centered at c completely covers the missing structure to be segmented. We explicitly enforce that $d \geq s$ for any choice of s. We call pixels within the seed radius s of c "seed pixels" and pixels within the dilation radius d of c "dilation pixels." In this interaction, seed pixels are *guaranteed* to be part of the missing segment that is added, as shown by the green circle in Figure 3.3 (b), and dilation pixels, excluding seed pixels, are *potentially* part of the missing segment that is added, as shown by the blue area in Figure 3.3 (b). This makes the selection of s and d conceptually simple for the user to tune. In Section 3.3, we discuss how to automate the selection of s and d to further reduce the user's burden when interactively segmenting a materials volume.



(a) (b) (c)

Figure 3.3: Annotating the addition of a missing segment. **(a)** Segmentation S^V with a missing segment near the center of the image. **(b)** Annotation of a center point c, along with a seed radius s and a dilation radius d, and the identified local region for updating the segmentation. **(c)** The updated segmentation of the local region shown in (b).

Similar to the removal interaction in Section 3.1, we define a local region around the specified c to update the segmentation of S^V. Specifically, we define this region by taking all segments in S^V that contain one or more seed or dilation pixels. In this local region we modify the Θ term of the energy minimization in Equation (1.1) to

Algorithm 3 Interactively specifying segment to add.

1: **function** AddSegment(S^V, c, s, d)
2: $\mathcal{L} \leftarrow$ union of all segments that contain a seed pixel or dilation pixel
3: $\forall p \in \mathcal{L}$, build graph for energy minimization problem from Chapter 1
4: $\Theta \leftarrow$ set from Equation (3.3) and Equation (3.4)
5: $\tilde{S}^V \leftarrow S^V$ incorporating the updated segmentation in $\mathcal{L}$
6: **return** updated $\tilde{S}^V$

obtain an updated segmentation. Specifically, we allow all seed and dilation pixels to be reassigned to the new segment $\tilde{S}^V_{n+1}$ by setting

$$\Theta_p(\tilde{S}^V_{n+1}) = \begin{cases} 0, & \|p - c\| \leq d \\ \infty, & \text{otherwise} \end{cases} \tag{3.3}$$

where $\|\cdot\|$ is the euclidean distance between pixels p and c. Furthermore, to insure that the seed pixels are always guaranteed to be part of $\tilde{S}^V_{n+1}$ we set an infinity penalty for seed pixels assigned to any segment other than $\tilde{S}^V_{n+1}$,

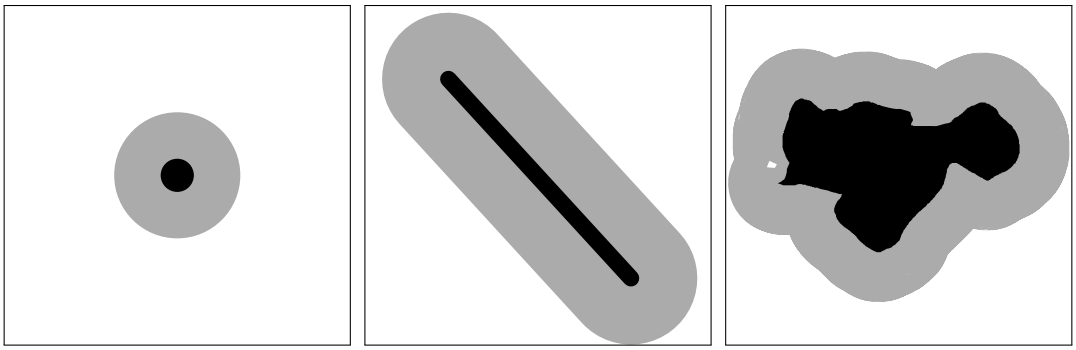
$$\Theta_p(\tilde{S}^V_i) = \begin{cases} \infty, & \|p - c\| \leq s \text{ and } i \neq n+1 \\ \Theta_p(S^V_i), & \text{otherwise.} \end{cases} \tag{3.4}$$

The full algorithm for adding a missing segment is summarized in Algorithm 3.

Note that annotations need not be constrained to a single point, as shown in Figure 3.4 (a). Line-based annotations, as shown in Figure 3.4 (b) can be defined by setting seed pixels to be all those pixels within a distance s of any position along an annotated line. Dilation pixels can be defined similarly. Further, any "scribble"-like annotation, provided it remains connected and without holes, can be morphologically dilated by s or d to obtain seed or dilation pixels, respectively, as shown in Figure 3.4 (c).

3.2 Annotation Repropagation

While annotations defining addition or removal of segments can be made on a single slice, they may provide valuable information for adjacent slices and, as such, can be



(a) (b) (c)

Figure 3.4: Alternate annotation shapes for addition. **(a)** Standard point-annotation. **(b)** Line annotation. **(c)** "Scribble" annotation. The grey area covers where seed pixels would be selected.

propagated (which we refer to as "repropagation") to these adjacent slices, similar to the segmentation propagation discussed in Chapter 1. For removal, we simply locate the same segment in all adjacent slices (if present), and repeat the removal operation summarized in Algorithm 2. For addition, illustrated in Figure 3.5, we create the new segment as summarized in Algorithm 3, after which we repropagate the segmentation, including the newly-created segment, to the adjacent slices in the same manner as discussed in Chapter 1. This repropagation can be done within the local region $\mathcal{L}$ on the adjacent slices for efficiency, and terminates if the new segment is no longer present during the repropagation (determined by the energy minimization discussed in Chapter 1).

When errors are small, such annotation repropagation may not be particularly beneficial. However, when there are gross errors, annotation repropagation can reduce the number of needed annotations significantly.

3.3 Parameter Estimation

When interactively adding a new segment, as discussed in Section 3.1, the seed radius s and dilation radius d are required to be specified by the user. This results in additional burden on the part of the user. In this section, we develop a parameter

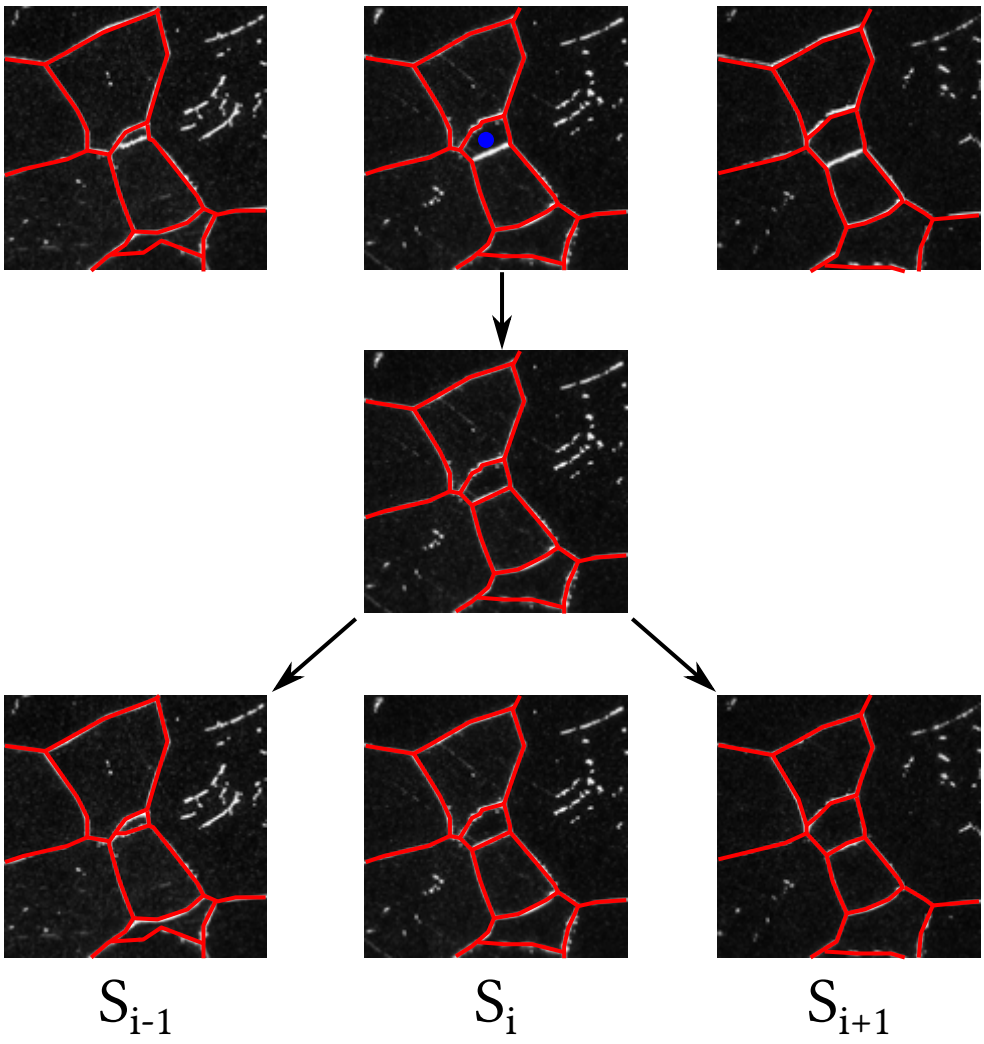
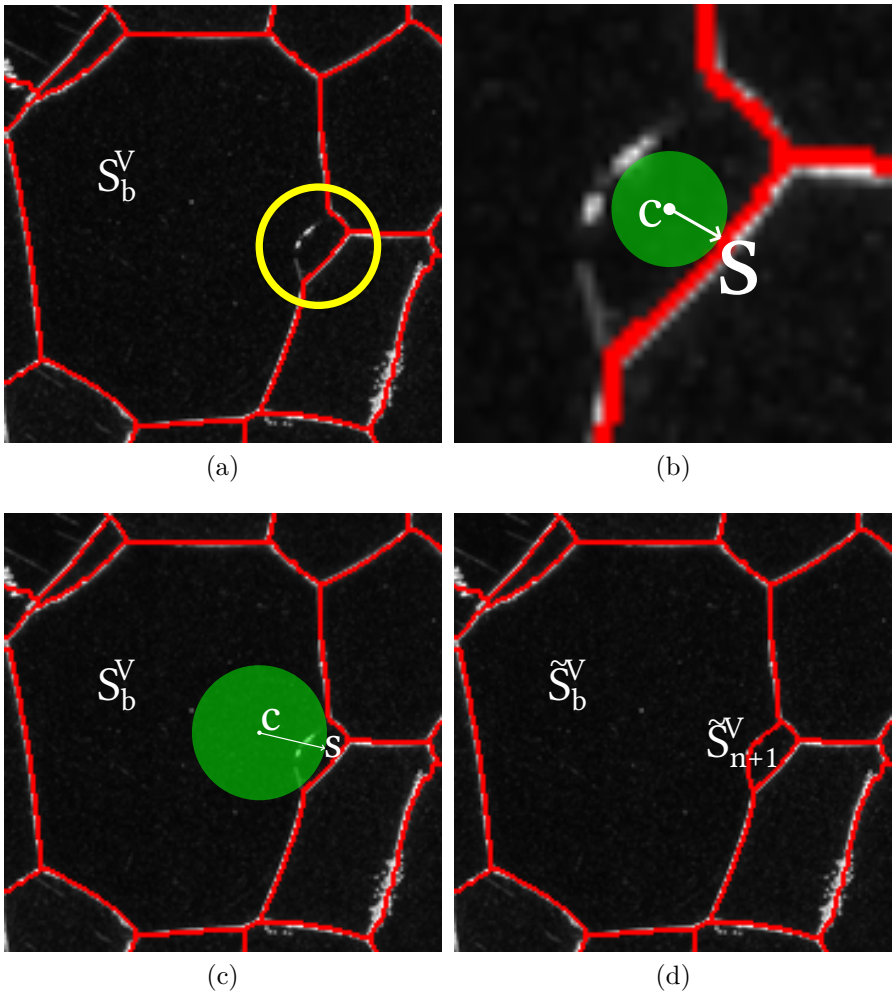


Figure 3.5: Annotation repropagation for addition. The blue annotation on slice S_i (row 1) yields an updated segmentation of slice S_i (row 2). This new segmentation can be repropagated to the neighboring slices S_{i-1} and S_{i+1} (row 3) using the same propagation approach introduced in Chapter 1.

estimation approach to automatically select these two parameters so the user need only override them in very rare cases, or not at all.

We do this by leveraging information about the center c the user provided relative to the initial segment in which it resides. Generally a missing segment occurs when 2D cross-section intersects a new 3D structure in V. Given a small inter-slice distance, we expect that these missing segments are often small compared with its neighboring segments in slice V. An example is shown in Figure 3.6 (a), where a small segment is missing (indicated by the yellow circle) in the segmentation S^V: this missing segment is mistakenly merged into a large neighboring segment S_b^V. Intuitively, placing c near the boundary of S_b^V likely indicates the missing segment is small, as shown by Figure 3.6 (b). Conversely, placing c closer to the center of S_b^V likely indicates the resulting missing segment is large as shown in Figure 3.6 (c). We make a simplifying assumption that we do not allow the missing segment to spill over the boundary of S_b^V. For example, the selection of c and s in Figure 3.6 (b) is able to generate the updated segmentation shown in Figure 3.6 (d).

To obtain an estimation of s we start by setting $s = 0$, and we then increase s by a small ϵ amount until the circle centered at c with radius s is within ϵ distance of the boundary of the containing segment S_b^V, as shown by the arrow in Figure 3.6 (b-c). In materials images, the majority of newly-appearing structures when moving from one slice to another are usually near the boundary of an existing structure S_b^V (near a "Y"-junction boundary between structures). This automatic approach is ideally suited for these cases. When the user specifies a c that falls directly on a segment boundary in S^V, we default to requiring user-supplied s in these less-common cases. For estimating d, it is scaled according to the value of s. Specifically, we set $d = 2 \cdot s$. As shown in Section 3.5, this approach saves both time and effort.



(a) (b) (c) (d)

Figure 3.6: Automatic selection of seed radius s and dilation radius d. **(a)** A missing segment located within a large segment in S_b^V. **(b-c)** Selections of c at varying distances from the boundary of S_b^V, resulting in different estimations of s. **(d)** Updated $\tilde{S}^V$ by adding a missing segment using the c shown (b) and the proposed parameter estimation method to determine s.

3.4 Salient Region Detection

Because materials images can be very large and complex, it can take a significant amount of time for a human annotator to review the segmentation of such a large image to determine where it may require additional interaction. In this section, we introduce a salient region detection approach that identifies candidate regions highlighting the areas most likely to need additional interaction. In this work, we define a salient region as a subset of the complete image that is most likely to require inter-

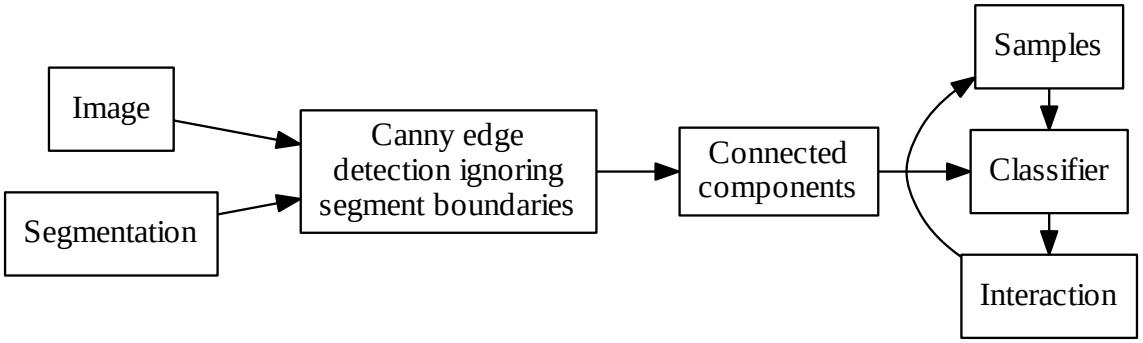


Figure 3.7: Salient region detection pipeline. Edges are detected and classified as salient or non-salient. Those classified as salient are then on shown in a bounding box displayed on the image to guide the interactive segmentation. Based on the interaction, positive and negative samples can be extracted to enhance the classifier for the next image/segmentation pair.

activity, i.e. where the segmentation shows large uncertainty. Since the addition of missing segments is the most time-consuming annotation task, we focus on detecting the edges in the image that are unsegmented, indicating a missing segment, for this salient region detection. As such, we identify prominent edges that are not segmented during the propagation as candidates, and use a SVM classifier [25] to learn which candidates are truly salient edges, and which are noise that can be ignored by the human annotator. These salient edges are later enclosed in a bounding box to form a salient region.

More specifically, we use the online learning [95] system outlined in Figure 3.7. We first use the Canny edge detector [15] in areas that are a fixed distance from the segmentation boundaries, preventing edges in the image which already fall on segmentation boundaries from being considered candidates, effectively leaving "residual" edges, which may correspond to either edges of missed structures or noise. The output of the edge detector is separated into connected components which are then classified using an initial pool of 300 positive and negative samples to determine which should be presented as salient regions. Finally, when the user is presented with the salient regions highlighted on the segmentation, any subsequent interactive

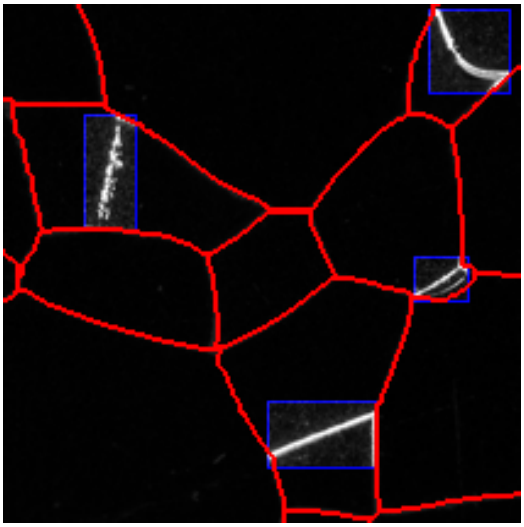


Figure 3.8: Qualitative results of salient region detection. Salient regions are surrounded a bounding box (blue), highlighting locations indicating where segments (bounded by red lines) may be missing.

annotations are converted to positive samples, and remaining unannotated salient regions are converted to negative samples. This process is then repeated for the next image/segmentation pair, with updated samples integrated into the classifier.

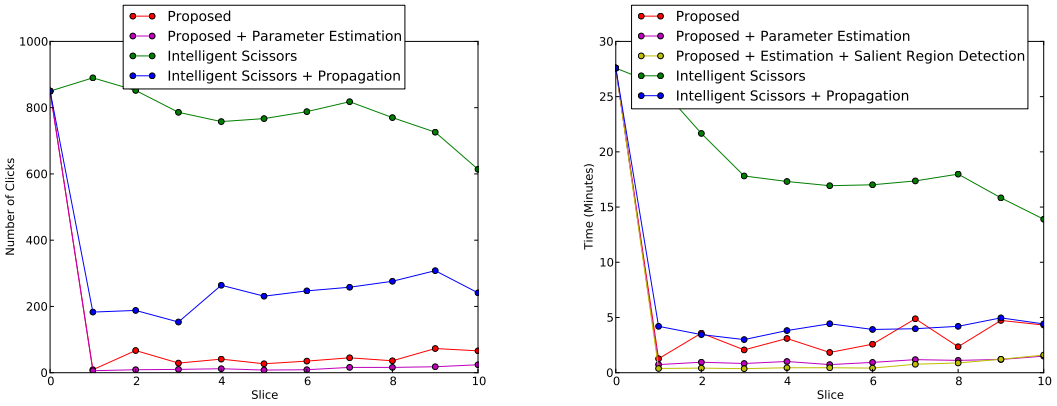
To classify each edge, we extract a feature vector consisting of multiple shape and intensity properties, including the total area occupied by the edge, the minor and major axis length of the ellipse fit to the edge, the maximum intensity along the edge, and the mean intensity along the edge. These properties can be computed quickly, which fits well with the real-time requirements of the interactive segmentation problem. This feature vector is then used to classify the edges as centers of potential salient regions. For the classifier, we use a SVM with a RBF kernel, which can be retrained while a new image is being annotated. Salient regions, as identified by the classified edges, are visualized in a bounding box to enhance their visibility. Qualitative results are shown in Figure 3.8, where salient missing edges are highlighted and surrounded by bounding boxes, but non-salient regions (noise in the interiors of the grains) are suppressed.

β-Ti grains in Ti-21S

To evaluate the proposed interactive segmentation method, we use it to segment a sequence of 11 (indexed from 0 to 10) microscopic titanium images [89] provided by Dave Rowenhorst, NRL. We measure the effort (i.e., number of clicks) used to segment each slice in the dataset, as well as the overall time expended by the user to segment a slice. The previous segmentation propagation approach in Chapter 1 requires a complete segmentation on one slice as an initialization. We count the manual segmentation on this initial slice into the effort and time required. We present the proposed method both with and without using the automatic parameter estimation discussed in Section 3.3. The human annotator was guided by the salient region detection approach discussed in Section 3.4 for all variants of the proposed method.

For comparison, we use the readily-available "Intelligent Scissors" interactive segmentation method [70]. Using the intelligent scissors tool, we independently segment all 11 slices from the same dataset, evaluating both effort (number of clicks) and time. In addition, we produce a hybrid of our previous automatic method in Chapter 1 and the intelligent scissors method, which we call "Intelligent Scissors + Propagation" in Figure 3.9. This approach uses the method from Chapter 1 to propagate a segmentation from an initial slice to the remaining slices, but it uses the intelligent scissors tool [70] to carry out the interactive component instead of the interaction proposed in this work.

The results of this comparative experiment are shown in Figure 3.9. Note that, in propagated methods (all "Proposed" methods and "Intelligent Scissors + Propagation"), the first slice is used as the initial slice U, so it requires significantly more effort and time to segment compared with the remaining slices. From Figure 3.9, we can see that the method proposed in this chapter ("Proposed") allows much more rapid

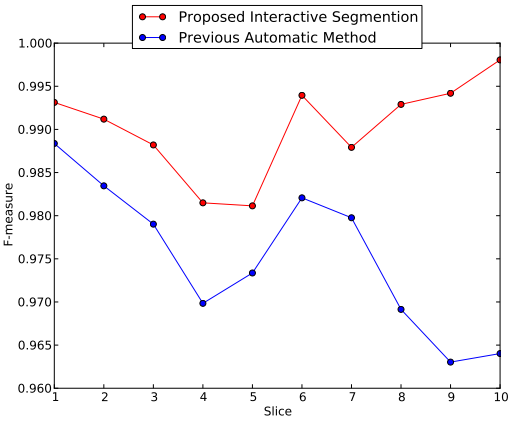


(a) Effort (b) Time

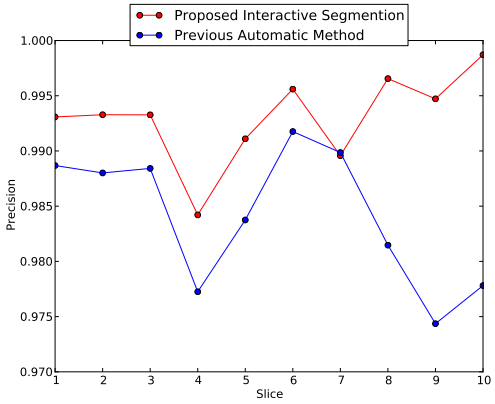
Figure 3.9: Evaluation of **(a)** the amount of effort (number of clicks) and **(b)** time taken for a user to interactively segment the 11 sample slices. Smaller values are better for both figures.

segmentation time (< 5 minutes in most cases) and with much less effort (< 100 clicks in most cases) compared with the unpropagated intelligent scissors method. The intelligent scissors method ("Intelligent Scissors"), without the benefit of propagation, requires significantly more time and effort. The hybrid method ("Intelligent Scissors + Propagation") fares better than the unpropagated intelligent scissors method, but it still requires greater effort than the proposed method. The proposed parameter estimation method ("Proposed + Parameter Estimation") can further reduce both the time and effort required by the proposed method. Finally, the salient region detection method does not change the effort required, but it is able to reduce the amount of time needed to segment this dataset, as shown in Figure 3.9 (b) ("Proposed + Estimation + Salient Region Detection").

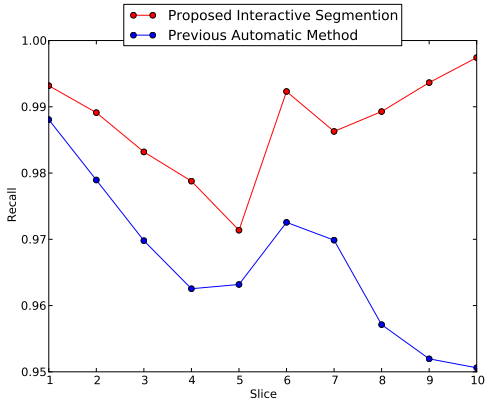
In Figure 3.10, we show that the proposed interactive method is able to increase the segmentation accuracy of our state-of-the-art materials image segmentation method in Chapter 1. As before, we use the precision, recall, and F-measure, which is the harmonic mean of the precision and recall [66], to show the segment boundary coincidence with the manually-constructed ground truth segmentation. For both the proposed and previous automatic methods, we propagate from an initial slice 0



(a) F-measure



(b) Precision



(c) Recall

Figure 3.10: Performance of the proposed interactive segmentation method compared with our previous automated method in Chapter 1 measured by the boundary coincidence with the ground truth segmentation.

to the remaining 10 slices, and the proposed interaction-enhanced method increases performance for all slices. Finally, qualitative segmentation results using the proposed interactive method are shown in Figure 3.11 where we show the automatic segmentation results with spurious or missing segments, the human annotation, and the updated segmentation.

DREAM3D Synthetic Volume

We further evaluate the proposed interactive segmentation approach by generating a synthetic grain volume using DREAM3D [48], discussed earlier in Chapter 2. As

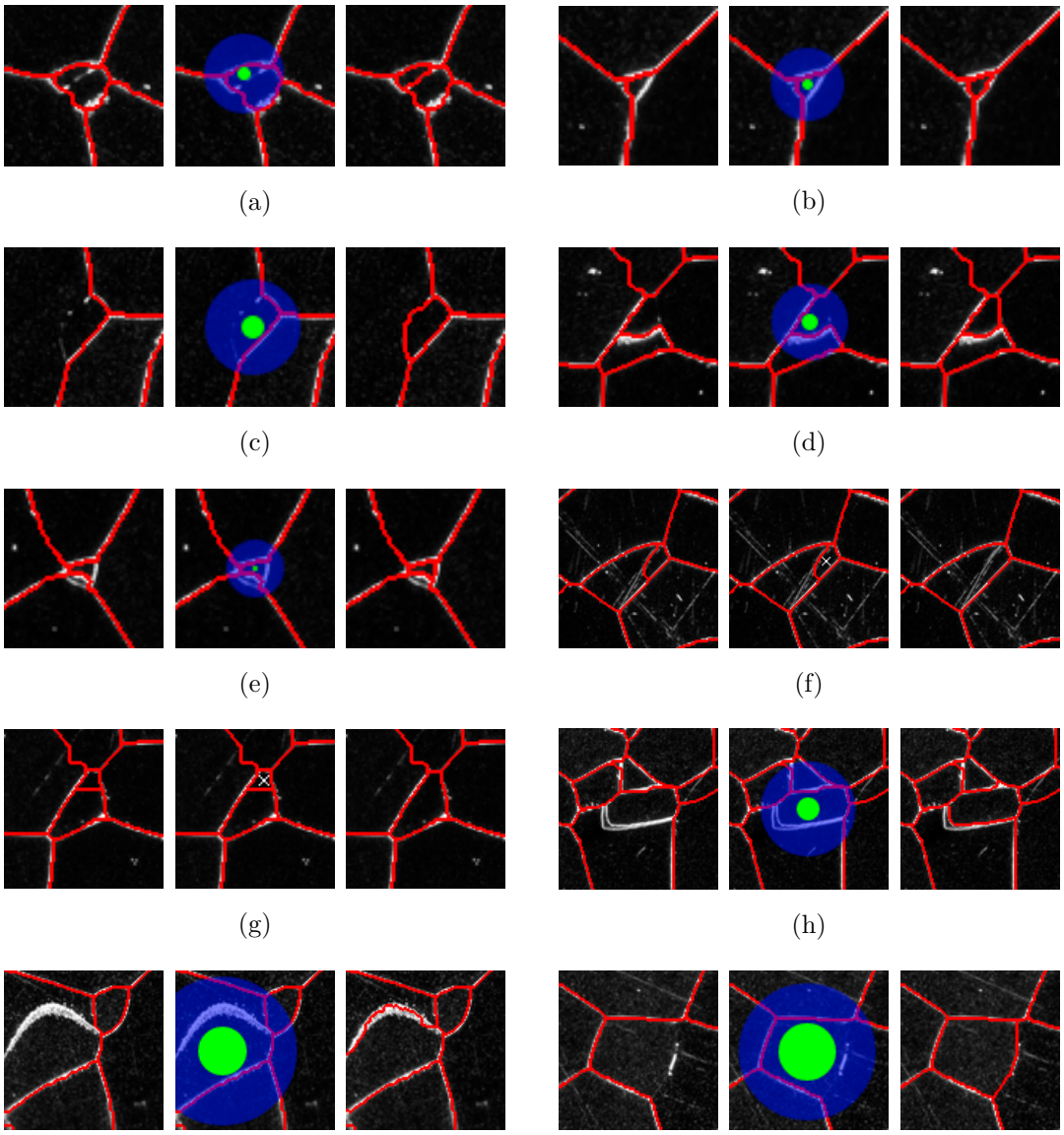


Figure 3.11: Qualitative results where each subfigure shows the initial automatic segmentation S^V (left); the human annotation (middle) with the seed pixels in green and dilation pixels in blue, and "X"s indicating spurious segments to be removed; and the updated segmentation $\tilde{S}^V$ (right). Note that (f) and (g) illustrate removal annotation and the remaining illustrate addition annotation.

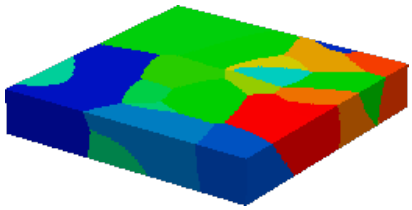
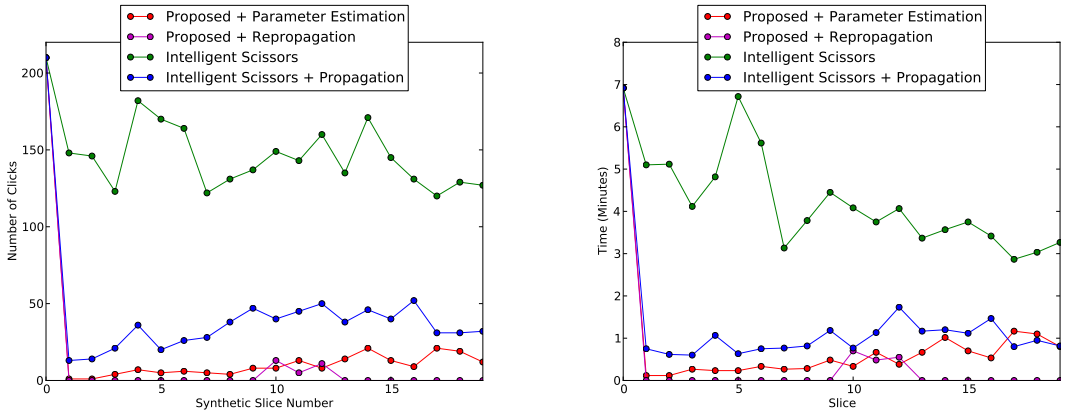


Figure 3.12: Synthetic volume generated by DREAM3D [48] to evaluate the proposed interactive segmentation.

shown in Figure 3.12, we generate a 12 µm × 12 µm × 2 µm volume where each slice is 0.06 µm × 0.06 µm × 0.1 µm in size. This volume is sampled as a 200 × 200 × 20 image, sliced along the z-axis into 20 images of size 200 × 200 each. We use these 20 synthetic slices, along with a noise model extracted from the above Ti-21S material (see Chapter 2 for more details), as additional data to evaluate the proposed interactive segmentation. As before, we evaluate interactive methods that start from an initial segmentation, and one method that does not. For methods starting from an initial segmentation, we use the results of propagating the ground truth for the first slice to all the remaining slices using the approach discussed in Chapter 1.

Quantitative results of this evaluation are shown in Figure 3.13. Included in this evaluation is the proposed method with the paramter estimation discussed in Section 3.3 ("Proposed + Parameter Estimation"), the proposed method using the repropagation method discussed in Section 3.2 ("Proposed + Repropagation"), along with the intelligent scissors method used to completely segment every slice ("Intelligent Scissors") and used only to correct the propagated results ("Intelligent Scissors + Propagation"). As shown in Figure 3.13 (a), the proposed method with parameter estimation requires less effort compared with both intelligent scissors methods. With the addition of the repropagation method, interactions are only needed on three slices, since the corrections are then repropagated to the rest of the volume, resulting in significantly less effort than all other methods. Results for the time taken by the evaluated methods are shown in Figure 3.13 (b), where the intelligent scissors meth-



(a) Effort (b) Time

Figure 3.13: Evaluation on the DREAM3D synthetic dataset showing **(a)** the amount of effort (number of clicks) and **(b)** time taken for a user to interactively segment the 20 sample slices. Smaller values are better for both figures.

ods all require more time, whereas the proposed method with repropagation performs better than the other evaluated methods.

Conclusion

In this work, we developed a segmentation propagation approach for segmenting 3D materials images. In this approach, 2D segmentation is propagated repeatedly from a slice to its neighbors and we developed new strategies to consider shape, intensity, and topological constraints during this propagation. To handle topology inconsistency, we developed a strategy to identify and accommodate such inconsistencies in local areas. To preserve the shape of segments in the propagation, we extracted the skeleton of a considered segment of interest and passed both its location and assigned segmentation to the new slice. To incorporate intensity of segments, we modeled the intensity distribution of the segment of interest, and used this to help identify the same segment in the new slice. We tested the proposed method on different materials datasets and achieved promising performance that is superior to the well-known watershed and normalized cut methods.

We then presented a new method for propagating a multi-label segmentation from one image to another by preserving both inter-segment topology and intra-segment topology. The former requires the specified segment adjacency relations and the latter requires the connectivity of each segment. We showed that this can be accomplished when each segment and its adjacent segments constitute a ring structure. We applied this new topology-preserving method to segment 3D metallic images for underlying grain structures, and achieved better performance than several comparison methods.

Finally, we adapt the propagation approach with an interactive segmentation component. By exploiting the inherit grain-like structure of some materials images, we are able provide minimal and simplistic human tools that lead to improved segmentation

performance. Specifically, we provide annotations that allow the addition of missing segments and the removal of spurious segments. Each annotation is localized and the graph-cut algorithm is used to quickly modify the underlying image segmentation. Comparisions on both effort ("number of clicks") and time show that the proposed approach is more efficient than comparision methods.

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: Theory, algorithms, & applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter, *Molecular biology of the cell*, 4 ed., Garland Science Taylor & Francis Group, 2002.

[3] C. Andres, F. Caballero, C. Capdevila, and D. Martin, *Revealing austenite grain boundaries by thermal etching: advantages and disadvantages*, Materials Characterization **49** (2002), no. 2, 121–127.

[4] Melih S. Aslan, Asem M. Ali, Dongqing Chen, Ben Arnold, Aly A. Farag, and Ping Xiang, *3D vertebrae segmentation using graph cuts with shape prior constraints.*, IEEE International Conference on Image Processing, IEEE, 2010, pp. 2193–2196.

[5] D. Basanta, M. A. Miodownik, P. J. Bentley, and E. A. Holm, *Evolving and growing microstructures of materials using biologically inspired CA*, NASA/DoD Conference on Evolvable Hardware, 2004, p. 275.

[6] D. Batra, A. Kowdle, D. Parikh, Jiebo Luo, and Tsuhan Chen, *iCoseg: Interactive co-segmentation with intelligent scribble guidance*, IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3169–3176.

[7] Hélène Bauby, Fanchon Divol, Elisabeth Truernit, Olivier Grandjean, and Jean-Christophe Palauqui, *Protophloem differentiation in early arabidopsis thaliana development*, Plant and Cell Physiology **48** (2007), no. 1, 97–109.

[8] J. M. Berg, J. L. Tymoczko, and L. Stryer, *Biochemistry.*, 6th ed., W. H. Freeman & Co Ltd, 2007.

[9] Yuri Boykov and Gareth Funka-Lea, *Graph cuts and efficient N-D image segmentation*, International Journal of Computer Vision **70** (2006), 109–131.

[10] Yuri Boykov and Marie-Pierre Jolly, *Interactive organ segmentation using graph cuts*, Medical Image Computing and Computer-Assisted Intervention MICCAI 2000 (Scott Delp, Anthony DiGoia, and Branislav Jaramaz, eds.), Lecture Notes in Computer Science, vol. 1935, Springer Berlin / Heidelberg, 2000, pp. 147–175.

[11] Yuri Boykov, Olga Veksler, and Ramin Zabih, *Fast approximate energy minimization via graph cuts*, IEEE TPAMI **23** (2001), no. 11, 1222–1239.

[12] Y.Y. Boykov and M.P. Jolly, *Interactive graph cuts for optimal boundary & region segmentation of objects in nd images*, IEEE International Conference on Computer Vision, vol. 1, IEEE, 2001, pp. 105–112.

[13] G. Bradski, *The OpenCV Library*, Dr. Dobb's Journal of Software Tools (2000).

[14] A. Brahme, M.H. Alvi, D. Saylor, J. Fridy, and A. D. Rollett, *3D reconstruction of microstructure in a commercial purity aluminum*, Scripta Materialia **55** (2006), no. 1, 75–80.

[15] J. Canny, *A computational approach to edge detection*, IEEE TPAMI **8** (1986), no. 6, 679–698.

[16] 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, 2011, pp. 17–24.

[17] C. Carson, S. Belongie, H. Greenspan, and J. Malik, *Blobworld: Color- and texture-based image segmentation using EM and its application to image querying and classification*, IEEE TPAMI **24** (2002), no. 8, 1026–1038.

[18] Yong-su Chae and Desok Kim, *Automatic marker-driven three dimensional watershed transform for tumor volume measurement*, International Conference on Advances in Hybrid Information Technology, 2007, pp. 149–158.

[19] Chao Chen, Daniel Freedman, and Christoph H. Lampert, *Enforcing topological constraints in random field image segmentation*, IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[20] H. Chuang, L. Huffman, M. Comer, J. Simmons, and I. Pollak, *An automated segmentation for nickel-based superalloy*, IEEE International Conference on Image Processing, 2008, pp. 2280–2283.

[21] D. Comaniciu and P. Meer, *Mean shift: A robust approach toward feature space analysis*, IEEE TPAMI **24** (2002), no. 5, 603–619.

[22] Mary Comer, Charles Bouman, Marc De Graef, and Jeff Simmons, *Bayesian methods for image segmentation*, JOM Journal of the Minerals, Metals and Materials Society **63** (2011), 55–57.

[23] M.L. Comer and E.J. Delp, *Parameter estimation and segmentation of noisy or textured images using the EM algorithm and MPM estimation*, IEEE International Conference on Image Processing, vol. 2, IEEE, 1994, pp. 650–654.

[24] _____, *The EM/MPM algorithm for segmentation of textured images: Analysis and further experimental results*, Image Processing, IEEE Transactions on **9** (2000), no. 10, 1731–1744.

[25] Corinna Cortes and Vladimir Vapnik, *Support-vector networks*, Machine Learning **20** (1995), no. 3, 273–297.

[26] C. Couprie, L. Grady, L. Najman, and H. Talbot, *Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest*, IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 731–738.

[27] T. Cour, F. Benezit, and J. Shi, *Spectral segmentation with multiscale graph decomposition*, IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 1124–1131.

[28] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, *Watershed cuts: Thinnings, shortest path forests, and topological watersheds*, IEEE TPAMI **32** (2010), no. 5, 925 –939.

[29] I.J. Cox, S. B. Rao, and Y. Zhong, *Ratio regions: A technique for image segmentation*, International Conference on Pattern Recognition, 1996, pp. 557–564.

[30] M. de Bruijne, B. van Ginneken, M. A. Viergever, and W. J. Niessen, *Adapting active shape models for 3D segmentation of tubular structures in medical images*, Inf Process Med Imaging **18** (2003), 136–147.

[31] R S Dilmaghani, A Ahmadian, M Ghavami, and A H Aghvami, *3D brain tumor segmentation using level-sets method and meshes simplification from volumetric MR images*, Signal Processing Letters IEEE **11** (2004), no. 10, 127–131.

[32] J. H. Elder, A. Krupnik, and L. A. Johnston, *Contour grouping with prior models*, IEEE TPAMI **25** (2003), no. 6, 661–674.

[33] J. C. Gamio, S. Belongie, and S. Majumdar, *Normalized cuts in 3D for spinal MRI segmentation*, IEEE TPAMI **26** (2004), no. 1, 36–44.

[34] Y. Gdalyahu, D. Weinshall, and M. Werman, *Stochastic image segmentation by typical cuts*, IEEE Conference on Computer Vision and Pattern Recognition, 1999, pp. 596–601.

[35] Stuart Geman and Donald Geman, *Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images*, IEEE TPAMI **6** (1984), no. 6, 721–741.

[36] R. C. Gonzalez and R. E. Woods, *Digital image processing (3rd edition)*, Prentice Hall, 2008.

[37] L. Grady, *Random walks for image segmentation*, IEEE TPAMI **28** (2006), no. 11, 1768–1783.

[38] D. M. Greig, B. T. Porteous, and A. H. Seheult, *Exact maximum a posteriori estimation for binary images*, Journal of the Royal Statistical Society Series B **51** (1989), 271–279.

[39] Michael Groeber, Somnath Ghosh, Michael Uchic, and Dennis Dimiduk, *Developing a robust 3-D characterization-representation framework for modeling polycrystalline materials*, JOM Journal of the Minerals, Metals and Materials Society **59** (2007), 32–36, 10.1007/s11837-007-0113-5.

[40] X. Han, C. Xu, and J. L. Prince, *A topology preserving deformable model using level set*, IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. 765–770.

[41] G.T. Herman, *Fundamentals of computerized tomography: Image reconstruction from projections*, Advances in Pattern Recognition, Springer, 2009.

[42] E. A. Holm, C. C. Battaile, H.E. Fang, T. E. Buchheit, and G. W. Wellman, *Making the connection between microstructure and mechanics*, 2008.

[43] L. Huffman, J. Simmons, and I. Pollak, *Segmentation of digital microscopy data for the analysis of defect structures in materials using nonlinear diffusion*,

Proceedings of SPIE: Computational Imaging VI (C. Bouman, E. Miller, and I. Pollak, eds.), 2008.

[44] Landis M. Huffman, Jeff P. Simmons, Marc De Graef, and Ilya Pollak, *Shape priors for map segmentation of alloy micrographs using graph cuts*, IEEE Workshop on Statistical Signal Processing, 2011, pp. 28–30.

[45] I. A. Ibrahim, F. A. Mohamed, and E. J. Lavernia, *Particulate reinforced metal matrix composites-a review*, Journal of Materials Science **26** (1991), 1137–1156.

[46] K Ishihara, *Bioinspired phospholipid polymer biomaterials for making high performance artificial organs*, Science and Technology of Advanced Materials **1** (2000), no. 3, 131–138.

[47] M. Jackson, *Bluequartz software*, 2007–.

[48] Michael Jackson and Michael Groeber, *DREAM3D*, **h**ttp://dream3d.bluequartz.net , 2012.

[49] Viren Jain, Benjamin Bollmann, Mark Richardson, et al., *Boundary learning by optimization with topological constraints*, IEEE Conference on Computer Vision and Pattern Recognition, June 2010, pp. 2488–2495.

[50] K. Janssens, D. Olmsted, E.A. Holm, S.M. Foiles, S.J. Plimpton, and P.M. Derlet, *Computing the mobility of grain boundaries*, Nature Materials **5** (2006), 124–127.

[51] Eric Jones, Travis Oliphant, Pearu Peterson, et al., *SciPy: Open source scientific tools for Python*, 2001–.

[52] Jennifer L. Jordan, D. W. Richards, Jonathan E. Spowart, Bradley White, and Naresh N. Thadhani, *Microstructural design & optimization of highly filled epoxy based composites*, Tech. Report ADA511295, Air Force Research Lab Eglin AFB FL Munitions Directorate, 2009.

[53] A.C. Kak and M. Slaney, *Principles of computerized tomographic imaging*, Classics in applied mathematics, Society for Industrial and Applied Mathematics, 2001.

[54] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, International Journal of Computer Vision **1** (1988), no. 4, 321–331.

[55] R. Kindermann, J.L. Snell, and American Mathematical Society, *Markov random fields and their applications*, Contemporary mathematics, American Mathematical Society, 1980.

[56] Zhanghui Kuang, Dirk Schnieders, Hao Zhou, Kwan-Yee Wong, Yizhou Yu, and Bo Peng, *Learning image-specific parameters for interactive segmentation*, IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 590–597.

[57] Sarang Lakare, *3D segmentation techniques for medical volumes*, Neural Networks **65** (2000), 1–23.

[58] Louisa Lam, Seong-Whan Lee, and Ching Y. Suen, *Thinning methodologies-a comprehensive survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992), no. 9, 869–885.

[59] Claude Leclerc, Claudia Brose, ClàďèàdŁmence NouzàďèàdŁ, Fransisca Leonard, Laleh Majlessi, Sybille Becker, Hagen von Briesen, and Richard Lo-Man, *Immobilized cytokines as biomaterials for manufacturing immune cell based vaccines*, Journal of Biomedical Materials Research Part A **86A** (2008), no. 4, 1033–1040.

[60] David Letscher and Jason Fritts, *Image segmentation using topological persistence*, Computer Analysis of Images and Patterns, 2007, pp. 587–595.

[61] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D. Fox, *Level set evolution without re-initialization: A new variational formulation*, IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2005, pp. 430–436.

[62] Dongsheng Li, Anthony D. Rollett, Greg Vialle, and Hamid Garmestani, *Multiproperty microstructure and property design of magnetic materials*, Journal of Engineering Materials and Technology **130** (2008), no. 2, 21–23.

[63] Qingwu Li, Xue Ni, and Guogao Liu, *Ceramic image processing using the second curvelet transform and watershed algorithm*, IEEE International Conference on Robotics and Biomimetics, 2007, pp. 2037–2042.

[64] M. Maire, P. Arbelaez, C.C. Fowlkes, and J. Malik, *Using contours to detect and localize junctions in natural images*, IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[65] J. Marroquin, S. Mitter, and T. Poggio, *Probabilistic solution of ill-posed problems in computational vision*, Journal of the American Statistical Association (1987), 76–89.

[66] 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, vol. 2, 2001, pp. 416–423.

[67] MATLAB, *version 7.7.0.471 (r2008b)*, The MathWorks Inc., Natick, Massachusetts, 2008.

[68] T. Mcinemey and D. Terzopoulos, *Topology adaptive deformable surfaces for medical image volume segmentation*, IEEE Transactions on Medical Imaging **18** (1999), no. 10, 840–850.

[69] Fernand Meyer, *Topographic distance and watershed lines*, Signal Processing **38** (1994), no. 1, 113–125.

[70] Eric N. Mortensen and William A. Barrett, *Intelligent scissors for image composition*, Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (New York, NY, USA), SIGGRAPH '95, ACM, 1995, pp. 191–198.

[71] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Communications on Pure and Applied Mathematics **42** (1989), no. 5, 577–685.

[72] Ken Museth, David E. Breen, Leonid Zhukov, and Ross T. Whitaker, *Level set segmentation from multiple non-uniform volume datasets*, IEEE Visualization, 2002, pp. 179–186.

[73] Sabine Neuss, Christian Apel, Patricia Buttler, Bernd Denecke, Anandhan Dhanasingh, Xiaolei Ding, Dirk Grafahrend, Andreas Groger, Karsten Hemmrich, Alexander Herr, Willi Jahnen-Dechent, Svetlana Mastitskaya, Alberto Perez-Bouza, Stephanie Rosewick, Jochen Salber, Michael WàdĕàdŸltje, and Martin Zenke, *Assessment of stem cell/biomaterial combinations for stem cell-based tissue engineering*, Biomaterials **29** (2008), no. 3, 302–313.

[74] R. Nock and F. Nielsen, *Statistical region merging*, IEEE TPAMI **26** (2004), no. 11, 1452–1458.

[75] Jorge Novo, Noelia Barreira, Manuel González Penedo, and José Santos, *Topological active volume 3D segmentation model optimized with genetic approaches*, Natural Computing **11** (2012), no. 1, 161–174.

[76] Sebastian Nowozin and Christoph H. Lampert, *Global connectivity potentials for random field models*, IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 818–825.

[77] T. Ogura, A. Makino, and T. Masumoto, *A grain boundary etching method for the analysis of intergranular p-segregation in iron-based alloys*, Metallurgical and Materials Transactions A **15** (1984), no. 8, 1563–1570.

[78] David L. Olson and Dursun Delen, *Advanced data mining techniques*, 1st ed., Springer Publishing Company, 2008.

[79] N Otsu, *A threshold selection method from gray-level histograms*, IEEE Transactions On Systems, Man, And Cybernetics **9** (1979), no. 1, 62–66.

[80] D. Huttenlocher P. Felzenszwalb, *Efficient graph-based image segmentation*, International Journal of Computer Vision **59** (2004), no. 2, 167–181.

[81] J.B. Park and J.D. Bronzino, *Biomaterials: Principles and applications*, CRC Press, 2003.

[82] P. Perona, T. Shiota, and J. Malik, *Anisotropic diffusion*, Geometry-Driven Diffusion in Computer Vision (B. M. H. Romeny, ed.), Dordrecht & Boston: Kluwer Academic Publishers, 1994, pp. 73–92.

[83] Sai T. Reddy, Melody A. Swartz, and Jeffrey A. Hubbell, *Targeting dendritic cells with biomaterials: developing the next generation of vaccines*, Trends in Immunology **27** (2006), no. 12, 573–579.

[84] R.C. Reed, *The superalloys: Fundamentals and applications*, Cambridge University Press, 2006.

[85] C.G. Roberts, S.L. Semiatin, and A.D. Rollett, *Particle-associated misorientation distribution in a nickel-base superalloy*, Scripta Materialia **56** (2007), no. 10, 899–902.

[86] A.D. Rollett, G. Gottstein, L. Shvindlerman, and D. Molodov, *Grain boundary mobility: a brief review*, Z. Metallkunde **95** (2004), 226–229.

[87] C. Rother, V. Kolmogorov, and A. Blake, *Grabcut: Interactive foreground extraction using iterated graph cuts*, ACM Transactions on Graphics (Proc. of SIGGRAPH) **23** (2004), no. 3, 309–314.

[88] D.J. Rowenhorst, A. Gupta, C.R. Feng, and G. Spanos, *3D crystallographic and morphological analysis of coarse martensite: Combining EBSD and serial sectioning*, Scripta Materialia **55** (2006), no. 1, 11–16.

[89] D.J. Rowenhorst, A.C. Lewis, and G. Spanos, *Three-dimensional analysis of grain topology and interface curvature in a β-titanium alloy*, Acta Materialia **58** (2010), 5511–5519.

[90] D. Rueckert and P. Burger, *Geometrically deformable templates for shape-based segmentation and tracking in cardiac MR image*, International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 1997, pp. 83–98.

[91] Jakob Santner, Thomas Pock, and Horst Bischof, *Interactive multi-label segmentation*, 2011, pp. 397–410.

[92] Rebekka K. Schneider, Julia Anraths, Rafael Kramann, JàdĕàdŸrg Bornemann, Manfred Bovi, Ruth Knàdĕàdđchel, and Sabine Neuss, *The role of biomaterials in the direction of mesenchymal stem cell properties and extracellular matrix remodelling in dermal tissue engineering*, Biomaterials **31** (2010), no. 31, 7948–7959.

[93] B. Settles, *Active learning literature survey*, Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[94] M. Sezgin and B. Sankur, *Survey over image thresholding techniques and quantitative performance evaluation*, Journal of Electronic Imaging **13** (2004), no. 1, 146–165.

[95] Shai Shalev-Shwartz, *Online Learning: Theory, Algorithms, and Applications*, Ph.D. thesis, The Hebrew University of Jerusalem, July 2007.

[96] L. G. Shapiro and G. C. Stockman, *Computer vision*, Upper Saddle River, NJ: Prentice Hall, 2001.

[97] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 731–717.

[98] _____, *Normalized cuts and image segmentation*, IEEE TPAMI **22** (2000), no. 8, 888–905.

[99] Jianbo Shi and J. Malik, *Motion segmentation and tracking using normalized cuts*, IEEE International Conference on Computer Vision, 2008, pp. 1154–1160.

[100] J. P. Simmons, P. Chuang, M. Comer, J. E. Spowart, M. D. Uchic, and De M. Graef, *Application and further development of advanced image processing algorithms for automated analysis of serial section image data*, Modelling and Simulation in Materials Science and Engineering **17** (2009), no. 2, 025002.

[101] JP Simmons, B. Bartha, M. De Graef, and M. Comer, *Development of bayesian segmentation techniques for automated segmentation of titanium alloy images*, Microscopy and Microanalysis **14** (2008), no. S2, 602–603.

[102] Anirudha Singh and Jennifer Elisseeff, *Biomaterials for stem cell differentiation*, J. Mater. Chem. **20** (2010), 8832–8847.

[103] J. S. Stahl and S. Wang, *Convex grouping combining boundary and region information*, IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 946–953.

[104] _____, *Globally optimal grouping for symmetric boundaries*, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 1030–1037.

[105] _____, *Salient boundary detection combining boundary, corner, and region information*, Technical Report, http://www.cse.sc.edu/∼songwang/document/techreport.pdf, Department of Computer Science and Engineering, University of South Carolina, 2006.

[106] C. Straehle, U. Koethe, G. Knott, K. Briggman, W. Denk, and F.A. Hamprecht, *Seeded watershed cut uncertainty estimators for guided interactive segmentation*, IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 765–772.

[107] C. N. Straehle, U. Köthe, G. Knott, and F. A. Hamprecht, *Carving: scalable interactive segmentation of neural volume electron microscopy images*, 2011, pp. 653–660.

[108] T. P. Swiler and E. A. Holm, *Diffusion in polycrystalline microstructures*, Annual Meeting of the American Ceramic Society, 1995.

[109] R. Szeliski, D. Tonnesen, and D. Terzopoulos, *Modeling surfaces of arbitrary topology with dynamic particles*, IEEE Conference on Computer Vision and Pattern Recognition, 1993, pp. 140–152.

[110] Jian Tan and W.Mark Saltzman, *Biomaterials with hierarchically defined micro- and nanoscale structure*, Biomaterials **25** (2004), no. 17, 3593–3601.

[111] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh, *Active learning for interactive 3D image segmentation*, Proceedings of the 14th international conference on Medical image computing and computer-assisted intervention, 2011, pp. 603–610.

[112] E. Truernit, H. Bauby, B. Dubreucq, O. Grandjean, J. Runions, J. Barthelemy, and J. C. Palauqui, *High-resolution whole-mount imaging of three-dimensional tissue organization and gene expression enables the study of Phloem development and structure in Arabidopsis*, Plant Cell **20** (2008), no. 6, 1494–1503.

[113] E. Truernit and J. C. Palauqui, *Looking deeper: whole-mount confocal imaging of plant tissue for the accurate study of inner tissue layers*, Plant Signal Behav **4** (2009), no. 2, 151–152.

[114] Zhuowen Tu, Xiang (Sean) Zhou, Dorin Comaniciu, and Luca Bogoni, *A learning based approach for 3D segmentation and colon detagging*, European Conference on Computer Vision (Berlin, Heidelberg), ECCV'06, Springer-Verlag, 2006, pp. 436–448.

[115] M. D. Uchic, M. Groeber, D. M. Dimiduk, and J.P. Simmons, *3D microstructural characterization of nickel superalloys via serial-sectioning using a dual beam FIB-SEM*, Scripta Materialia **55** (2006), 23–28.

[116] Markus Unger, Thomas Pock, Werner Trobin, Daniel Cremers, and Horst Bischof, *TVSeg—interactive total variation based image segmentation*, British Machine Vision Conference 2008, 2008, pp. 40.1–40.10.

[117] O. Veksler, *Star shape prior for graph-cut image segmentation*, European Conference on Computer Vision, 2008, pp. 454–467.

[118] Olga Veksler, *Efficient graph-based energy minimization methods in computer vision*, Ph.D. thesis, Cornell University, Ithaca, NY, USA, 1999.

[119] V. Vezhnevets and V. Konouchine, *Grow-Cut—interactive multi-label N-D image segmentation*, Graphicon, 2005, pp. 150–156.

[120] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother, *Graph cut based image segmentation with connectivity priors*, IEEE Conference on Computer Vision and Pattern Recognition **0** (2008), 1–8.

[121] L. Vincent and P. Soille, *Watersheds in digital spaces: An efficient algorithm based on immersion simulations*, IEEE TPAMI **13** (1991), no. 6, 583–598.

[122] Niels Volkmann, *A novel three-dimensional variant of the watershed transform for segmentation of electron density maps*, Journal of Structural Biology **138** (2002), no. 1-2, 123–129.

[123] S. Wang and J. M. Siskind, *Image segmentation with ratio cut*, IEEE TPAMI **25** (2003), no. 6, 675–690.

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

[125] L. Wojnar, *Image analysis: Applications in materials engineering*, Materials Science and Technology, CRC Press, 1999.

[126] N. Wuyts, J. C. Palauqui, G. Conejero, J. L. Verdeil, C. Granier, and C. Massonnet, *High-contrast three-dimensional imaging of the Arabidopsis leaf enables the analysis of cell dimensions in the epidermis and mesophyll*, Plant Methods **6** (2010), 17.

[127] Yun Zeng, Dimitris Samaras, Wei Chen, and Qunsheng Peng, *Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in N-D images*, Computer Vision and Image Understanding **112** (2008), no. 1, 81–90.

[128] Yimeng Zhang and Tsuhan Chen, *Efficient inference for fully-connected CRFs with stationarity*, IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 582–589.

[129] Y. Zhong, A. K. Jain, and M. P. Dubuisson-Jolly, *Object tracking using deformable template*, IEEE TPAMI **22** (2000), no. 5, 544–549.

[130] S. C. Zhu and A. Yuille, *Region competition: Unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation*, IEEE TPAMI **18** (1996), no. 9, 884–900.

[131] Stephan Zimmermann and Klaus Landes, *A particle image shape imaging (PSI) investigation of particles in a plasma jet*, 2nd International Conference on Spray

Deposition and Melt Atomization and the 5th International Conference on Spray Forming **383** (2004), no. 1, 153–157.