Graph Cut Approaches for Materials Segmentation Preserving Shape, Appearance, and Topology

Jarrell W. Waggoner[1], Jeff Simmons[2], Marc De Graef[3], Song Wang[1]

[1]University of South Carolina, Department of Engineering and Computing,
315 Main St.; Columbia, SC, 29208, USA
[2]Materials and Manufacturing Directorate, Air Force Research Labs;
Dayton, OH, 45433, USA
[3]Carnegie Mellon University, Department of Materials Science and Engineering,
5000 Forbes Avenue; Pittsburgh, PA, 15213, USA

Keywords: segmentation, propagation, shape, appearance, topology, graph cut

Abstract

Segmenting material images into underlying objects is an important but challenging problem given object complexity and image noise. Consistency of shape, appearance, and topology among the underlying objects are critical properties of materials images and can be considered as criteria to improve segmentation. For example, some materials may have objects 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 inter-object topology which constrains their neighboring relations. In this paper, we develop new graph-cut based approaches for materials science image segmentation. Specifically, these approaches segment image volumes by repeatedly propagating a 2D segmentation from one slice to another. We introduce different terms into the graph-cut cost function to enforce desirable shape, appearance, and topology consistency. We justify the effectiveness of the proposed approaches by using them to segment sequences of serial-section images of different materials.

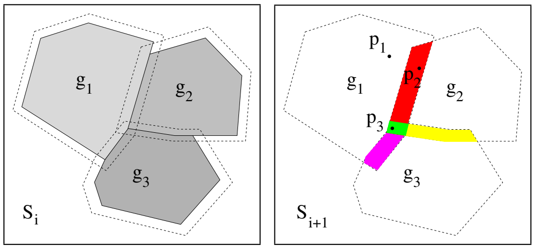
Segmentation via Labeling

In this paper we use the multi-labeling framework outlined in [1, 2, 3, 4] to *propagate* a 2D segmentation from one segmented slice S_i to another S_{i+1}. A labeling algorithm simply assigns an integer (label) to each pixel in S_{i+1}, in which we use labels to represent the phases in a material (e.g. specific crystal structures, or different material states). A label function in this framework f is formulated as

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N}} V_{pq}(f_p, f_q), \tag{1}$$

where $\mathcal{P}$ is the set of image pixels in S_{i+1}, f_p is the label (phase) of pixel $p \in \mathcal{P}$, and $\mathcal{N}$ is the set of pixel pairs that are neighbors in S_{i+1}. In this paper, pixels directly above, below, left, or right of a pixel are considered neighbors.

The *phase term*, $D_p(f_p)$, represents a cost for assigning a particular phase (label f_p) to a particular pixel p; it is propagated from the available segmentation of S_i [4]. In



(a) (b)

D	g_1	g_2	g_3
p_1	0	∞	∞
p_2	0	0	∞
p_3	0	0	0

(c)

Figure 1: An example of the phase term $D_p(f_p)$. **(a)** Objects g_1, g_2, and g_3 in slice S_i and the regions near the phase boundaries (dashed lines). **(b)** Phases propagated to slice S_{i+1}, where the colored regions are near phase boundaries. **(c)** Phase term D_p defined for three pixels in (b) in terms of three phase labels, g_1, g_2, and g_3.

particular, to reflect the phase continuity between S_i and S_{i+1}, we can set all the pixels in S_{i+1} to be the same phase as S_i (e.g. pixel p_1 in Fig. 1(b)), except for pixels near the phase boundaries (e.g. pixels p_2 and p_3 in Fig. 1(b)), where the pixels have non-infinity cost for all nearby phases, as shown in Fig. 1(b-c).

The *interface term*, $V_{pq}(f_p, f_q)$, represents the interfacial energy between phases (analogous to the Potts model for multiphase mixtures). If $f_p = f_q$ at two neighboring pixels p and q, then the interfacial energy is zero; otherwise there will be a penalty representing the energy between the different phases. In [4], if $f_p \neq f_q$, we define this penalty as

$$V_{pq}(f_p, f_q) = \begin{cases} \infty, & f_p \nLeftrightarrow f_q \\ \frac{255}{\max(S_{i+1}(p), S_{i+1}(q))}, & f_p \Leftrightarrow f_q \end{cases} . \qquad (2)$$

where $S_{i+1}(p)$ and $S_{i+1}(q)$ are the image intensities of pixels p and q in S_{i+1}, respectively. There is an infinity energy if the phases of p and q are different and these phases were not adjacent in S_i ($f_p \nLeftrightarrow f_q$). This prevents non-adjacent phases from being labeled as adjacent. There is a non-zero finite energy related to image information if the phases of p and q are different and these phases are adjacent in S_i ($f_p \Leftrightarrow f_q$). As detailed in [4], this drives phase boundaries toward desired image edges.

In [5], it is shown that finding a globally optimal labeling that minimizes Eq. (1) is an NP-hard problem, but a locally-optimal labeling can be efficiently found with the minimum graph-cut algorithm. In this algorithm, a graph model is constructed to represent pixels, and the phase terms and interface terms are encoded into the edge weights of the graph. Finding the local optimal labeling is reduced to the problem of partitioning the graph into two disjoint subgraphs by removing edges with minimum total edge weight. Furthermore, this problem can be solved by the well-known max-flow min-cut algorithm [6].

Preserving Shape

According to normal practice, the spacing between serial sections of a material are made frequently enough that at least roughly 10 slices are made through each object of interest (e.g. grain, precipitate, etc.). With this high density sampling, successive slices of the same object usually exhibit consistent shapes, at least through several sections, as shown in Fig. 2(a-b).

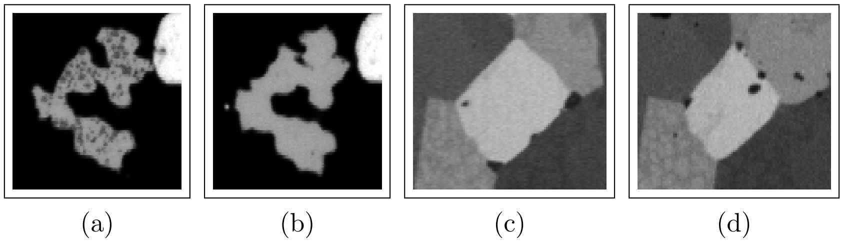


Figure 2: **(a-b)** Gamma′ precipitates in Rene88DT [7]. **(c-d)** Grains in IN100 superalloy [8], courtesy of Mike Groeber. **(a)** Slice S_i. **(b)** Slice S_{i+1}, where the object shows similar shape to the object in S_i. **(c)** Slice S_i. **(d)** Slice S_{i+1}, where the objects show similar intensity distribution to the objects in S_i.

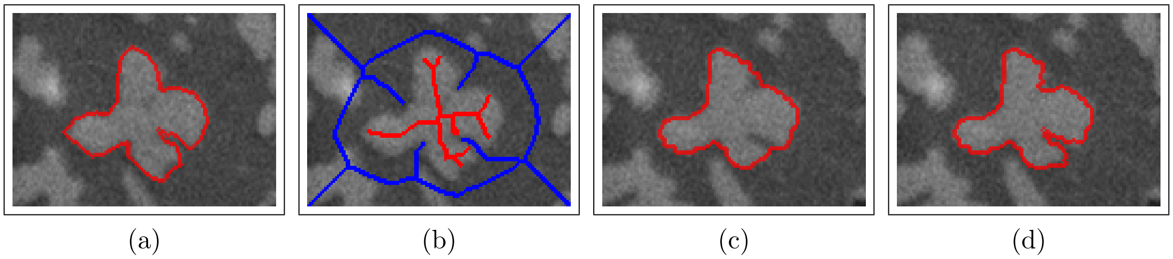


Figure 3: Dendritic gamma′ precipitates in Rene88DT. **(a)** Segmentation of slice S_i, created manually. **(b)** Skeletonizaton of the segmented object in slice S_i (morphologically eroded slightly), showing the shape of the object (red) and background (blue). **(c)** Segmentation results on slice S_{i+1} using our baseline method in [4]. **(d)** Segmentation result on slice S_{i+1} preserving shape between slices by assigning a desired phase to the skeleton pixels.

For the phase term in Eq. (1) we can define a Dirac delta function

$$D_p(f_p) = \begin{cases} 0, & f_p = d_p \\ \infty, & f_p \neq d_p \end{cases} . \tag{3}$$

if d_p is the desired phase for pixel p, and all other phases are known to be undesired. In the following, we use this approach to preserve an object shape when propagating a segmentation from S_i to S_{i+1}.

Specifically, we perform a skeletonization [9] of each object of interest in S_i, as shown in Fig. 3(b), and assign a desired phase to the skeleton pixels in S_{i+1}. The desired phase of a skeleton pixel in S_{i+1} is defined to be the phase of the same pixel in S_i. This helps preserve the shape of each object of interest when propagating the segmentation from S_i to S_{i+1}. The skeleton is morphologically eroded slightly to account for changes in the object's size. Note that for the pixels in S_{i+1} without desired phases, we still determine their phases by minimizing Eq. (1). Figure 3 shows a comparison between our baseline approach using the phase term as in [4] vs. the method outlined in this section to propagate a manually created segmentation in S_i to a new slice S_{i+1}.

Preserving Appearance

The intensity within an object can be another important property which may be desirable to preserve when propagating a segmentation. As shown in Fig. 2(c-d), an object might undergo changes in size or shape from one slice to the next while retaining a consistent intensity, aside from image noise. While the interface term $V_{pq}(f_p, f_q)$ in [4] incorporates

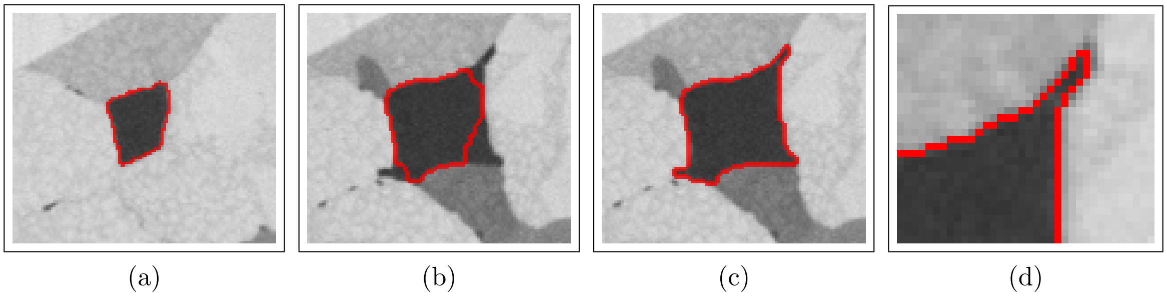


Figure 4: Grain structure of IN100 superalloy [8]. Image courtesy of Mike Groeber, AFRL. **(a)** Segmentation of slice S_i, created manually. **(b)** Segmentation result on slice S_{i+1} using our baseline method in [4]. **(c)** Segmentation result on slice S_{i+1} considering object intensity consistency between S_i and S_{i+1}. **(d)** Zoomed view of upper-right corner of (c).

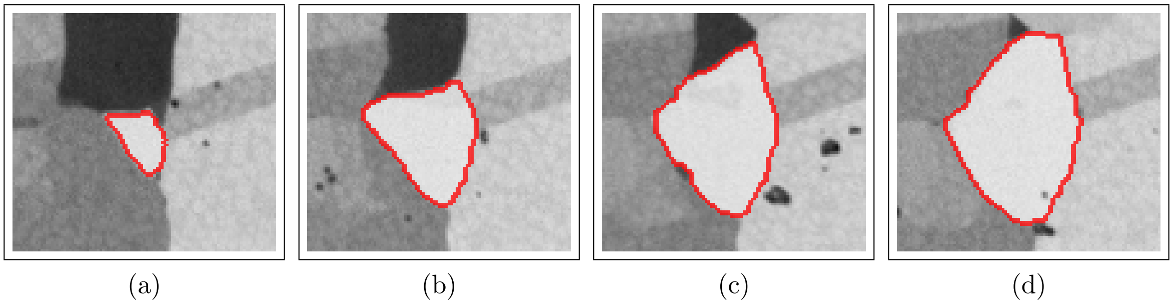


Figure 5: Same IN100 [8] as in Fig. 4. **(a)** Segmentation of slice S_i, created manually. **(b-d)** Segmentation results of iteratively propagating the segmentation in (a) to subsequent slices by using the intensity-preserving technique.

pairwise intensity difference between neighboring pixels to determine whether they have the same phase or different phases, it does not capture the *overall* intensity of a specific object. As such, we present a method to incorporate the intensity of an object into the phase term $D_p(f_p)$ in Eq. (1).

Our key observation is that the overall intensity of many objects 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 phase in S_i, we can determine the probability of a pixel in slice S_{i+1} belonging to a specific object. For each phase f from slice S_i and corresponding Gaussian $\mathcal{N}(\mu_f, \sigma_f^2)$, the phase term $D_p(f_p)$ can be defined by evaluating all the pixels in slice S_{i+1} against this distribution by using a Dirac delta function

$$D_p(f_p) = \begin{cases} 0, & (\mu_f + \lambda\sigma_f) > S_{i+1}(p) > (\mu_f - \lambda\sigma_f) \\ \infty, & \qquad\qquad otherwise \end{cases} \tag{4}$$

where $S_{i+1}(p)$ is the intensity of pixel p in slice S_{i+1} and $\lambda > 0$ is a sensitivity parameter. As in [4], we add an additional constraint that only pixels within a certain distance of the original object in S_i are considered. Figure 4 shows a comparison between our baseline approach using the phase term as in [4] vs. the method outlined in this section to propagate a manually created segmentation in S_i to a new slice S_{i+1}. Figure 5 shows a sample segmentation result after iteratively propagating for several slices.

It is possible to combine the techniques discussed in this paper by first applying the intensity-based technique in this section, followed by the shape-based technique presented in the previous section. A sample result is shown in Fig. 6.

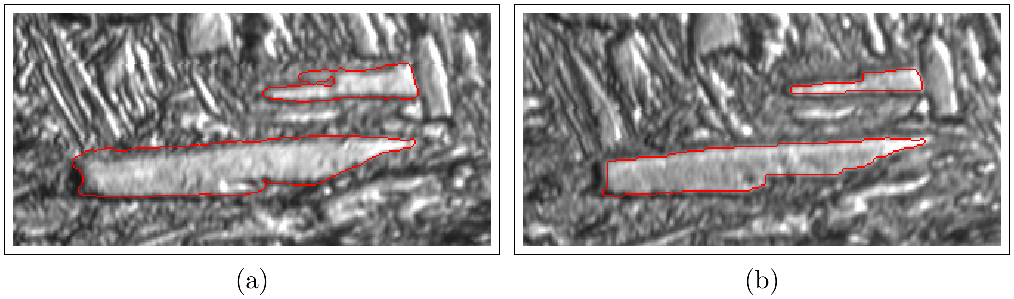


Figure 6: Martensite lath structure in steel [10]. Image courtesy Dave Rowenhorst, NRL. **(a)** Rough segmentation of slice S_i created by hand. Notice the boundaries do not correspond well to object boundaries. **(b)** Segmentation result on slice S_{i+1} by using both shape-preserving and appearance-preserving techniques.

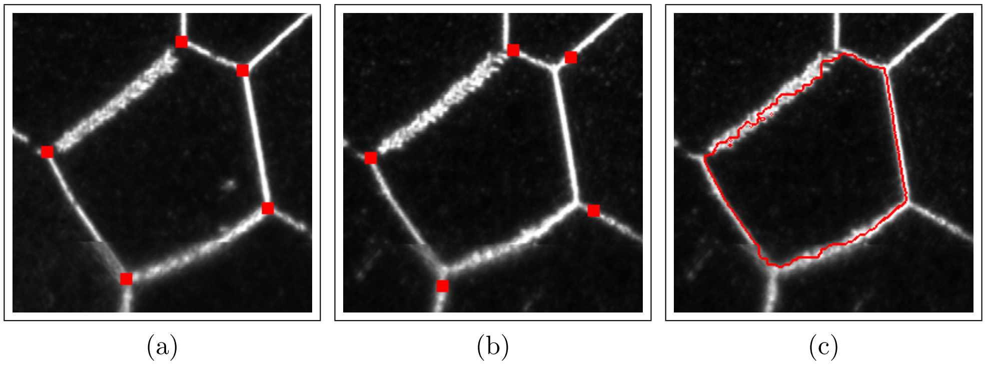


Figure 7: β-Ti grains in Ti-21S [12], courtesy of Dave Rowenhorst, NRL. **(a)** Slice S_i, where the triple junctions shown as red dots (manually selected). **(b)** Slice S_{i+1}, where the triple junctions have been propagated from the previous slice. **(c)** Segmentation result of applying our baseline method in [4] to the region bounded by the points in (b).

Preserving Junction Topology

In addition to propagating a segmentation by minimizing an energy (such as Eq. (1)), it is possible to instead propagate key topological points in some materials objects. For example, polycrystalline metals have triple junctions where three objects (grains) meet. The relationship between triple junctions in two slices is determined by the dihedral angle that models the interfacial energy between the grains [11]. In some cases, we may have prior knowledge about this dihedral angle which can help us determine the location of triple junctions in the new slice S_{i+1}. For example, we can select a number of candidate points for each triple junction in slice S_{i+1}, and choose the candidate that leads to dihedral angles that closely conform to the known prior knowledge.

We conducted a simple experiment on a subset of an electron microscopy titanium volume provided by Dave Rowenhorst, which is shown in Fig. 7. For such material, the dihedral angles at triple junctions are always close to 120°. For each triple junction point in S_i, we find a set of candidate corresponding triple junction points in S_{i+1} by examining the image intensity gradient in S_{i+1}. We select the candidate that leads to dihedral angles that are closest to 120° for each triple junction. Results are shown in Fig. 7(b). Based on the identified triple junctions, the multi-labeling techniques described above can be used to refine the boundaries, as shown in Fig. 7(c).

Conclusion

In this paper, we have shown how to incorporate both shape and appearance consistency to improve material science image segmentation by propagating a 2D segmentation from one slice to another, based on a graph-cut energy minimization framework. For incorporating shape, we preserve the topology consistency of the object skeleton between slices. For incorporating appearance, we model the intensity distribution of the object of interest, and use this model to segment the same object in the new slice. Finally, we proposed an approach to incorporate prior knowledge of the dihedral angles to improve localization of triple junctions, and use this approach to improve segmentation. We have presented sample results for each approach.

Acknowledgments We would like to thank Dave Rowenhorst from the Navy Research Lab for providing the titanium serial section images and Mike Groeber for the IN100 images. This work was funded, in part, by AFOSR FA9550-11-1-0327, NSF-1017199, NSF-0951754, and ARL W911NF-10-2-0060. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of AFOSR, NSF, ARL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References

[1] Y. Boykov and V. Kolmogorov. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 (9) (2004), pp. 1124–1137.

[2] Y. Boykov, O. Veksler, and R. Zabih. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (11) (2001), pp. 1222–1239.

[3] V. Kolmogorov and R. Zabih. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 (2) (2004), pp. 147–159.

[4] J. Waggoner, J. Simmons, and S. Wang. *Proceedings of SPIE (Computational Imaging X)*, vol. 8296 (Burlingame, CA, 2012).

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

[6] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, & Applications* (Englewood Cliffs: Prentice Hall).

[7] J. MacSleyne, M. Uchic, J. Simmons, and M. D. Graef. *Acta Materialia*, 57 (2009), pp. 6251–6267.

[8] M. Groeber, S. Ghosh, M. D. Uchic, and D. M. Dimiduk. *JOM Journal of the Minerals, Metals and Materials Society*, 59 (2007), pp. 32–36.

[9] L. Lam, S.-W. Lee, and C. Y. Suen. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (9) (1992), pp. 869–885.

[10] D. Rowenhorst, A. Gupta, C. Feng, and G. Spanos. *Scripta Materialia*, 55 (1) (2006), pp. 11–16.

[11] C. H. P. Lupis. *Chemical Thermodynamics of Materials* (Elsevier, 1983).

[12] D. Rowenhorst, A. Lewis, and G. Spanos. *Acta Materialia*, 58 (2010), pp. 5511–5519.