Segmentation of Polycrystalline Images Using Voronoi Diagrams

Uzziel Cortez

Advisor: Selim Esedoglu
Department of Mathematics, University of Michigan, Ann Arbor

April 24, 2019

Abstract

We present methods used to develop an algorithm for segmenting images of polycrystalline materials, which was done over the last year fulfilling the research requirement for the Marjorie Lee Brown Scholars Program. Our work makes extensive use of Voronoi Diagrams used to model grain boundary networks, deriving equations used to represent the network in terms of a finite set of points, piecewise constant Mumford-Shah model widely used in computer vision for image segmentation, and gradient descent.

Introduction

Material development is essential to overcoming many hurdles that our society is currently facing. Development of new materials such as: super-alloys, aerogels, and graphene (among many others), have applications in industrial, marine, and aerospace engineering, medicine, electronics, and environmental applications as well. In this application, we focus our work on Polycrystalline materials. Polycrystalline materials are composed of many crystalline parts that are randomly oriented with respect to each other. The material's properties, such as: conductivity, strength, hardness, corrosion resistance, etc., are largely dependent on its microstructure properties such as: grain size, grain boundary distribution, grain deformations, chemical composition, etc. Determining a materials properties through accurate analysis of grain structure is crucial to the development of new materials and subsequent advancement of engineering. Obtaining accurate measurements through existing methods, Electron Backscatter Diffraction and Light Optical Microscopy, can be expensive and tedious[2]. In light of this, we propose an algorithm which takes in an input image, an image of a grain polycrystalline material, and outputs a binary image which accurately represents the grain boundary network in the input image. The methodologies presented here are an adaptation of work done by Elsey, and Slepčev [1] on gradient flow of Voronoi Diagrams with some relaxations concerning topological events in the dynamics.

Voronoi Diagrams

We begin by modeling grain boundary networks through the use of Voronoi Diagrams. A Voronoi Diagram is a partition of a plane into distinct regions, where each region is generated by a point. More formally, given a set of generating points $P = \{p_1, p_2, ..., p_n\}$, a plane is partitioned into n regions, $\{R_1, R_2, ..., R_n\}$, such that:

- Each point p_i lies in exactly one region R_i.
- For any point $q \notin P$ that lies in region R_i, the Euclidean distance from p_i to q will be shorter than the Euclidean distance from p_j to q $\forall j \neq i$

We define the edges making up the boundaries of the Voronoi Diagram, or partition, as $S = \{s_1, s_2, ..., s_k\}$ and we will also define the vertices that define these edges as $V = \{v_1, v_2, ..., v_m\}$.

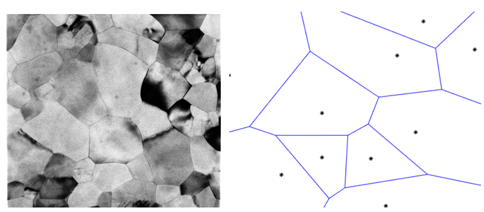


Figure 1: (Left) TEM Image of Polycrystalline Material. (Right) A Voronoi Diagram

Model

Our algorithm will make use of modeling a grain boundary network and use gradient descent to fit this network to the given input image. The first term of the objective function, that is the function which we will be optimizing, is the perimeter energy:

$$E_1 = \sum_{s_k \in S} Length(s_k) = \sum_{\substack{i,j\ s.t.\\ \overline{v_i v_j} = s_k \in S}} |v_i - v_j|$$

Applying gradient descent to this expression allows us to model the grain boundary network evolution dynamics. The second will be given by the piece-wise constant

Mumford-Shah model:

$$E_2 = \sum_i \int_{R_i} (f(x,y) - c_i)^2 dxdy$$

$$c_i = \frac{1}{l} \sum_{(x,y)\in R_i} f(x,y)$$

where $f(x,y)$ is the grayscale constant value at pixel location (x,y) in the image that is to be segmented, c_i is the average grayscale value of pixels in f that lie in region R_i, and l is the number of pixels in region R_i. This term allows us to measure the error in grayscale values of our approximation against the values in the image that is to be segmented. The final objective function then becomes

$$E = \sum_{\substack{i,j\ s.t.\\ \overline{v_i v_j} = s_k \in S}} |v_i - v_j| + \sum_i \int_{R_i} (f(x,y) - c_i)^2 dxdy$$

In order to produce an accurate segmentation of the input image, we need to find the locations of the points $P = \{p_1, p_2, ..., p_n\}$ which will produce line segments on the correct pixels with respect to the input image. We find these locations by minimizing the the objective function E with respect to the points in P.

Calculating Gradients

Perimeter Energy Gradient

$$\frac{\partial E_1}{\partial p_i} = \sum_{s_i \in S} \sum_{v_i \in vertex(s_i)} \frac{\partial Length(s_i)}{\partial v_i} \frac{\partial v_i}{\partial p_i}$$

Computing the gradients of the edges with respect to the vertices in V is a straightforward calculation. We will then focus on computing the gradients of the vertices with respect to the generating points in P and describe the computation of these gradients along two directions and use a change of basis to obtain the gradient in the standard basis.

By construction, each vertex v_i is a circumcenter of the triangle formed by three generating points p_1, p_2, p_3. For clarity we define the following:

- Generating Points: p_1, p_2, p_3
- Edges: $s_1 = (p_2 - p_1), s_2 = (p_3 - p_1), s_3 = (p_2 - p_3)$
- Vertex: v

We can perturb p_1 to $p_1 + z_1$ where z_1 is a vector in the direction of s_1 with magnitude dx_1, then the perpendicular bisector of s_1 moves along the same direction by $\frac{dx_1}{2}$. This results in v moving to $v + w_1$ where w_1 is a vector in the direction of $s_3^{\perp}$ with magnitude dv_1.
The problem then reduces to finding the magnitude of a side of a right triangle. If we let θ be the angle between s_1 and $s_3^{\perp}$ we get the following equation:

$$cos(\theta) = \frac{dx_1}{2}\frac{1}{dv_1} = s_1^T s_3^{\perp} \implies dv_1 = \frac{dx_1}{2 s_1^T s_3^{\perp}}$$

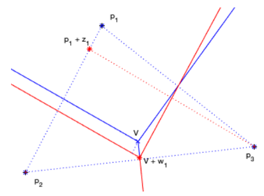


Figure 2: (Blue) Voronoi Diagram before perturbation of p_1. (Red) Voronoi Diagram after perturbation of p_1

(similar calculation for movement of p_1 along s_2) We now have a way to compute the gradients of v_i with respect to p_i along two specific directions. Using a change of basis, we can get the gradients in terms of the standard basis.

$$\frac{\partial v}{\partial p_1} = [w_1\ w_2][s_1\ s_2]^{-1}$$

PC Mumford-Shah Gradient

By letting $g(x,y) = (f(x,y) - c_i)^2$ we can express the gradient as:

$$\frac{\partial E_2}{\partial p_i} = \frac{\partial}{\partial p_i} \sum_i \int_{R_i} g(x,y) dxdy$$
$$= \sum_i \int_{\partial R_i} g(z) v(z)^{\perp} dz$$

Since we are working with images on a discrete grid, we can approximate the gradient by discretizing the expression and getting:

$$\frac{\partial E_2}{\partial p_i} \approx \sum_{R_i} \sum_{s_i \in edges(R_i)} \sum_{k=1}^{N} g(z) v_k(z)^{\perp} \Delta z$$

where

$$v(z)^{\perp} = \frac{L - r}{L} \frac{\partial v_i}{\partial p_1} \hat{n} + \frac{r}{L} \frac{\partial v_j}{\partial p_1} \hat{n}$$

with N being the number of discretized points along an edge and $\hat{n}$ being the normal vector to a given edge. Using this formulation, we can compute a discretized approximation to $\frac{\partial E_2}{\partial p_i}$.

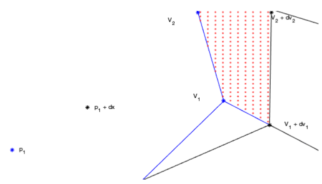


Figure 3: (Blue) Voronoi Diagram before perturbation of p_1. (Black) Voronoi Diagram after perturbation of p_1. (Red) Change in E_2 with respect to one edge of the region.

Handling Topological Events

Before employing gradient descent using the computed gradients described above, we must first handle some special cases:

Collision of generating points

When two generating points p_i, p_j come very close together, we observe extreme dynamics in the evolution of the Voronoi Diagram. To alleviate this issue, we define a repulsion term:

$$R(p_i, p_j) = R(d_{ij}) = e^{\frac{-1}{\theta^2 (r - d_{ij})^2}}$$

where $d_{ij} = |p_i - p_j|$, and θ and r are user tuned parameters. Adding this term to our objective function penalizes points for being close together and applying gradient descent to it will cause points, within the specified distance from each other, to be repelled from each other.

Boundary Events

Since we are interested in image segmentation, we need the generating points to stay within a finite boundary. As gradients are calculated and points are pushed outside of the image boundary, we project the center onto the boundary it passes through.

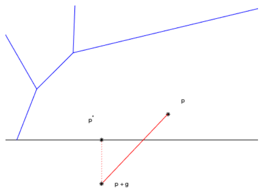


Figure 4: Projection of points onto boundary pushed out by gradient calculation

Collapsed Regions

As the boundary network evolves by gradient descent, low energy states in the boundary network are achieved, resulting in generating points which have regions with small area relative to the surrounding centers. In these cases we enforce the following constraint:

$$\text{Area}(R_i) \leq \tau \implies \text{Removal of } p_i$$

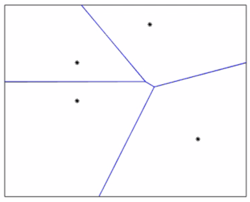
Results

Our results are presented in Figure 5. As we can see, the segmentation produced is very close to the sample image. There is a discrepancy near the junction of the boundaries in the approximation produced, the cause of which can be attributed to the fact that a junction of four boundaries is an unstable boundary network configuration. Minimization of the E_1 would cause a junction of four boundaries to separate into two triple junctions but minimization of E_2 is pushing towards a better segmentation. Our algorithm oscillates back and forth between these two states in the long run.

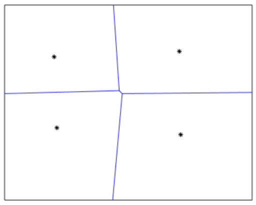
Future Work

Continuation of this work can proceed by preparing the algorithm for segmentation of real grain boundary

(a) A 10x10 sample image to segment.



(b) The initialization for our algorithm



(c) The resulting segmentation

Figure 5

networks. As shown in Figure 1 (Left), grains are not uniform in color and neighboring grains do not necessarily have distinct grayscale values. This can present a challenge since the PC Mumford-Shah model relies on having distinct colors in each region. One possible way to address this could be to do some preprocessing of the image to improve contrast between regions. Additionally, initialization of generating points is currently up to the user. One could employ K-Means clustering on the color values of the image that is to be segmented and use the centroids of the clustering as the initialization of the generating points for the Voronoi Diagram.

Acknowledgments

I would like to thank Selim Esedoglu and Tiago Salvador for the mentoring they provided throughout this project. Additionally, would also like to thank Yi Wen for all her help and support with the development of the algorithm.

References

1. Elsey, Matt, and Dejan Slepčev. "Mean-curvature flow of Voronoi diagrams." Journal of Nonlinear Science 25.1 (2015): 59-85.

2. Trimby, P., et al. "Is fast mapping good mapping? A review of the benefits of high-speed orientation mapping using electron backscatter diffraction." Journal of microscopy 205.3 (2002): 259-269.

3. Tai, Xue-cheng, and Chang-hui Yao. "Image segmentation by piecewise constant Mumford-Shah model without Estimating the constants." Journal of Computational Mathematics, vol. 24, no. 3, 2006, pp. 435-443. JSTOR, www.jstor.org/stable/43693303.