Redistancing dynamics for vector-valued multilabel segmentation with costly fidelity: grain identification in polycrystal images

Matt Elsey Benedikt Wirth

March 22, 2014

Abstract

A novel numerical method for multilabel segmentation of vector-valued images is presented. The algorithm seeks minimizers for a generalization of the piecewise-constant Mumford–Shah energy and is particularly appropriate for energies with a fitting (or fidelity) term that is computationally expensive to evaluate. The framework for the algorithm is the standard alternating-minimization scheme in which the update of the partition is alternated with the update of the vector-valued constants associated with each part of the segmentation. The update of the partition is based on the distance function-based diffusion-generated motion algorithms for mean curvature flow. The update of the vector-valued constants is based on an Augmented Lagrangian method. The scheme automatically chooses the appropriate number of segments in the partition. It is initialized with a partition of many more segments than are expected to be necessary. Adjacent segmentations of the partition are merged when energetically advantageous. The utility of the algorithm is demonstrated in the context of atomic-resolution polycrystalline image segmentation.

1 Introduction

In this article we consider a generalization of the piecewise-constant Mumford–Shah (PCMS) [25] energy which seeks a segmentation of a given, potentially vector-valued 2D or 3D image into multiple segments, where the energy associated with the interface between the segments may depend on the bulk values in the adjacent segments. Our algorithm employs a version of the distance function-based diffusion-generated motion (DFDGM) algorithms [16, 17, 20]. The bulk energy of the PCMS model is only fully evaluated a few times during the algorithm in order to obtain a simple quadratic approximation. This makes our method particularly suitable for computationally expensive bulk energies. We demonstrate the algorithm in the context of polycrystalline image segmentation. In the remainder of this introduction, we briefly review the extensive literature surrounding the PCMS model and its variants, give a brief overview of the DFDGM algorithms, and describe the problem of image segmentation in the context of atomic-resolution polycrystalline images.

1.1 Piecewise constant Mumford–Shah model

Given an input image u on a domain Ω, the well-known Mumford–Shah image segmentation functional [25] searches for a function q and a discontinuity set Γ such that q agrees well with u, is differentiable and slowly-varying away from Γ, and such that the discontinuity set Γ remains small. The function q and the discontinuity set Γ are found by minimizing the functional

$$E[\Gamma,q] = \mu^2 \int_\Omega (q-u)^2 \,\mathrm{d}x + \int_{\Omega-\Gamma} \|\nabla q\|^2 \,\mathrm{d}x + \beta|\Gamma|,$$

with $|\Gamma|$ denoting the total interfacial area of Γ, and μ and β positive weighting parameters.

Under the assumptions that Γ partitions Ω and that q is constant on each part of the partition, one obtains the basic PCMS model. Here, let $\{\Omega_i\}_{i=1,\dots,n}$ be a partition of Ω and associate with each part Ω_i a constant value q_i. The energy functional becomes

$$E[\{\Omega_i, q_i\}_{i=1,\dots,n}] = \sum_{i=1,\dots,n} \int_{\Omega_i} (q_i - u)^2 \,\mathrm{d}x + \frac{\beta}{2} \sum_{i=1,\dots,n} |\partial\Omega_i|, \tag{1}$$

where the parameter β may be rescaled to account for the discarded parameter μ.

In the simplest case, Γ is assumed to be a simple, closed curve dividing the image into two parts. The active contour method of Chan and Vese [8] is a level set method that approximates (1) for $n = 2$ by the energy

$$E(\phi, q_1, q_2) = \int_\Omega (q_1 - u)^2 H_\varepsilon(\phi) + (q_2 - u)^2 (1 - H_\varepsilon(\phi) \,\mathrm{d}x + \beta \int_\Omega \delta_\varepsilon(\phi) |\nabla\phi| \,\mathrm{d}x, \tag{2}$$

where H_ε and δ_ε are smoothed Heaviside and delta functions, respectively. Here, the level set function ϕ determines the partition by $\Omega_1 = \{x : \phi(x) > 0\}$, $\Omega_2 = \Omega \setminus \Omega_1$, while Γ is defined by the zero level set, $\Gamma = \{x : \phi(x) = 0\}$. The partition is updated by L^2-gradient descent on this energy functional with respect to ϕ. The scheme calls for alternating minimization over ϕ and the values q_1 and q_2. It is easily seen that for a fixed partition, the optimal values q_i are given by $q_i = \left(\int_{\Omega_i} u \,\mathrm{d}x\right) / |\Omega_i|$.

The basic approach of applying L^2 gradient descent to (2) given some initial level set function ϕ^0 will often find a local, rather than global, minimum. An approach of Chan, Esedoḡlu, and Nikolova [6] reformulates the segmentation task as a convex optimization problem in a higher-dimensional space. The convex method by Bresson et al. [5] additionally estimates the values q_1, q_2. Note, however, that such reformulations as convex minimisation problems only work for scalar-valued q_1, q_2 and strongly increase the computational cost due to the additional dimensions, for which reason we will follow a different approach.

Chan, Sandberg, and Vese extend the two-phase PCMS model to vector-valued images [7]. The resulting model takes as input a vector-valued image u and searches for a single contour Γ and constant vectors q_1 and q_2 minimizing Equation (2), with $(q_i - u)^2$ interpreted as $||q_i - u||_2^2$, the sum of squared differences of the components of the vectors q_i and u at x. This extension allows for the segmentation of RGB images or grayscale images with damage (e.g. noise, blurring, or occlusion) in multiple channels. In Section 1.3, we will review work which further extends the PCMS model to more complicated and expensive fidelity terms, i.e. ones allowing the identification of a textured or periodically patterned region by a (possibly vector-valued) constant.

For many images, the natural segmentation consists of more than two parts. Vese and Chan present a multiphase level set algorithm for PCMS [34]. Their approach uses p level set functions to segment the image into $n = 2^p$ regions. In their numerical experiments, $p = 2$ or 3, allowing for up to $n = 8$ parts in the segmentation. This approach generates a strong coupling between the p level set functions and is thus difficult to implement in the many-phase case. Furthermore, the length term is approximated in such a way that some parts of the curve may be counted multiple times, losing fidelity to the original model. Jeon et al. [22] provide a different extension to multiphase, extend the two phase level set PCMS algorithm by recursively performing binary segmentation on individual parts of the current partition.

A graph-based approach to image segmentation via the PCMS model was proposed by El–Zehiry, et al. [11] for the two-phase segmentation. In this case, the minimization of the PCMS energy for fixed constants q_1 and q_2 can be recognized as a minimum-cut problem on an associated graph. As such, it can be solved in polynomial time. The authors propose an iterative scheme, alternating the solution of the minimum-cut problem for fixed q_1 and q_2 with the update of these constants until fixed points for the values of q_1 and q_2 are found. These graph cut methods are non-local and obtain global minimizers for the separating curve Γ for any fixed values of q_1 and q_2.

The extension of the graph-based approach to the multiphase problem is non-trivial. It is known that the multicommodity minimum graph cut problem is NP-complete in the case that the number of parts is taken to be a variable. The solution of the multiphase problem is approximated by El–Zehiry, et al. [12] by solving a succession of binary segmentation problems via graph cuts, similar to the approach of Jeon et al. [22] in the level set framework. More recently, various groups including [2, 10, 13] have employed graph-based combinatorial optimization techniques to approximate solutions of the multiphase PCMS segmentation for four or fewer parts. The recent work of El–Zehiry and Grady [13] provides an extensive review of the literature surrounding algorithms for multiphase PCMS segmentation.

None of the existing level set or graph-based approaches are satisfactory for the many-label PCMS image segmentation problem. In this work, we make use of the distance function-based diffusion-generated motion algorithm originally proposed by Esedoḡlu, Ruuth, and Tsai [20] which, under the modification of Elsey, Esedoḡlu, and Smereka

[16], allows for the evolution of many phases (hundreds of thousands) under curvature-based evolutions which arise as L^2 gradient descent for energies involving perimeter penalties of the form (1) and give the correct approximation for $|\Gamma|$. This algorithm is reviewed in Section 1.2. In our approach, many phases are established initially, with the expectation that many of them will not be needed in the final segmentation. Their presence allows the minimization procedure to avoid many local minima. We implement a phase-merging technique to allow unneeded phases to be removed quickly.

1.2 DFDGM algorithms

The class of diffusion-generated motion schemes dates back to the work of Merriman, et al. [23, 24], which introduces the threshold dynamics scheme. The threshold dynamics schemes generate a discrete-in-time approximation of multiphase curvature flow by alternating two basic steps: convolution of the characteristic function of a set with a spherically symmetric kernel and thresholding the resulting convolution output to generate an updated characteristic function. Ruuth [31] extends the threshold dynamics approach to allow for L^2 gradient descent of the energy

$$E[\{\Omega_i\}_{i=1,\dots,n}] = \sum_{i=1}^{n} e_i|\Omega_i| + \frac{\beta}{2}\sum_{i,j=1}^{n} \sigma_{ij}|\Gamma_{ij}|,$$

for arbitrary weights e_i and positive β, σ_{ij}, with Γ_{ij} denoting the interface separating Ω_i and Ω_j. This descent gives the normal velocity

$$v_n(\Gamma_{ij}) = \mu_{ij}\left(e_i - e_j + \beta\sigma_{ij}\kappa_{ij}\right),$$

where κ_{ij} denotes the mean curvature of Γ_{ij}.

Ruuth's approach requires a nontrivial modification of the thresholding step, which requires the numerical approximation to resolve a nonlinear projection of convolution values in the three-phase case. When many phases are present, a weighted average of these projections is used. In the absence of the bulk terms, Esedoḡlu and Otto [19] present a different threshold dynamics-based scheme for the unequal surface energy motion. Their scheme relies on a new energetic interpretation of the basic threshold dynamics scheme, allows for the extension to arbitrary isotropic surface energies and mobilities. The new scheme is unconditionally gradient stable for a variety of surface energies, including the class of surface energies employed in this work. However, it does not consider the bulk energy-type terms which are crucial for the present work, and has not yet been implemented in the many-phase case.

The threshold dynamics scheme suffers from low accuracy on uniform grids, because characteristic functions do not naturally give subgrid resolution on uniform grids. Ruuth [**?**] implements an adaptive mesh refinement version of threshold dynamics. Another approach is to replace the characteristic function with a smoother function. This approach is exemplified by the DFDGM scheme, proposed by Esedoḡlu, Ruuth, and Tsai [20]. This scheme replaces the characteristic function by the signed distance function, and the thresholding step of threshold dynamics with a pointwise redistribution step and a "redistancing" step, which updates an arbitrary level set function to be the signed distance function to the zero level set of the input. Order $N\log N$ schemes (for N grid points) such as Sethian's Fast Marching Method [32] (equivalently, Tsitsiklis's algorithm [33]) and higher order versions [15] ensure that the computational complexity of the DFDGM algorithms is the same as the original threshold dynamics schemes, while significantly greater accuracy is achieved at a given resolution.

Elsey, Esedoḡlu, and Smereka extend the DFDGM scheme to allow for the evolution of many (hundreds of thousands) of phases by tracking many spatially separated phases within each signed distance function, and adding a step to the algorithm which detects and rectifies situations in which two phases contained within a single signed distance function begin to grow too close together [16]. They further achieve unequal surface energies (unequal weights w_{ij}) in the many-phase case [17]. Here we will propose a version of the PCMS scheme which assigns different surface energies to the interface between various parts in a principled way, and utilizes the DFDGM algorithm as part of the minimization scheme for this extended PCMS energy.

1.3 Polycrystalline image segmentation

Polycrystalline materials are characterized by the presence of large regions, called grains or crystals, consisting of atoms arranged in a perfect or nearly-perfect lattice structure. Suppose that a deformation map ψ transforms locations in the given image u to locations in a perfect reference crystal. Then the deformation gradient $G = D\psi \in \mathbb{R}^{2\times 2}$ is simply a rotation matrix in the absence of crystal strain. G is constant, or at least slowly varying, within each grain. One approach to analyzing an atomic-resolution polycrystalline image u is to attempt to construct the deformation map or deformation gradient. This approach is taken by Berkels, et al. [3] and by the present authors [18]. One advantage of this approach is that isolated defects are captured and their associated Burgers vectors can be determined by integrating curl$G$. The drawback is in the computational cost. The deformation map or gradient must be computed everywhere in the computational domain.

For some applications, it is sufficient to know an image segmentation. Each part of the segmentation is associated with a constant deformation gradient. The foregoing discussion suggests a PCMS-type model, where a penalty is paid for the discrepancy between the atomic lattice in the image and the atomic lattice predicted by the deformation gradient associated with the proper part of the image domain. Berkels, et al. advocate for this approach, using the Chan–Vese level sets approach for segmentation into a few crystalline regions. We will use the fidelity or fitting term from that work to penalize for the discrepancy between the observed lattice and the lattice defined by the segmentation task for each region. Boerdgen et al. [4] demonstrate a convex lifting approach which allows one to find global minimizers of this PCMS energy in the multiphase case. However, the approach used there is very computationally expensive from a memory standpoint and may not be able to be extended into three dimensions.

In the present work, we apply the DFDGM algorithm to a variant of the PCMS energy considered by Berkels et al. and Boerdgen et al. The DFDGM approach allows us to use many labels in our segmentation — indeed, our approach is to initialize our computations with a regular grid of many labels. The labeling is then updated by the alternation of DFDGM iterations which help to place the interfaces Γ_{ij} along grain boundaries in the image and by an orientation update and part merging scheme which ensure that appropriate orientations are associated with each region, and that adjacent regions of similar orientation are merged when energetically favorable.

A further advantage of the DFDGM algorithms is the ability to associate different weights with the different interfaces. This is in accordance with the understanding that the interfacial energy in a polycrystal depends on the relative orientations of the two adjoining grains. When the misorientation between grains is small, defects are spaced less tightly along the grain boundary, and interface energy is lower; high misorientation corresponds to high defect density and higher interface energy. For this work, we use a variation of the Read–Shockley surface energy [29], which gives surface energy as an increasing, concave function of misorientation angle. The model used is described fully in Section 2.

The key elements of this work are summarized as follows:

- Our implementation allows for the presence of very many segments in the partition. We initialize with many more segments than needed to provide robustness against local minima found due to a poor choice of initialization.

- In order to perform the minimization quickly, we implement a segment merging step to allow unneeded segments to be removed.

- We allow for different weights, depending on the vector-valued constants associated with neighboring regions, to be associated with each interface. The DFDGM algorithm allows us to evolve the system properly for these non-uniform energies.

- The fitting term we employ is both noisy and expensive to evaluate. We smooth the fitting energy density to allow large time steps in the evolution, and avoid updating the fitting energy density any more often than necessary.

- We utilize the split Augmented Lagrange Method to update the vector-valued constants (rotations) describing each segment. This update is significantly more difficult than that of standard PCMS models due to the non-constant interfacial energy. Furthermore, the number of expensive fitting term evaluations during the update is strongly reduced by the use of quadratic approximations.

2 Grain identification viewed as multilabel segmentation

In this section we more formally introduce the general multilabel segmentation problem and then present the specific version used here for crystal grain segmentation.

2.1 Multilabel segmentation

The classical multilabel segmentation problem reads as follows: Given an image $u : \Omega \to \mathbb{R}^m$ on a bounded open Lipschitz domain $\Omega \subset \mathbb{R}^d$ (or a flat torus in case of periodic boundary conditions), partition Ω into multiple segments $\Omega_1, \ldots, \Omega_n$ based on information from the image (e. g. such that the Ω_i represent different objects visible in u). Here, a valid partition of Ω is a collection of measurable subsets $\Omega_1, \ldots, \Omega_n \subset \Omega$ which are pairwise disjoint and whose union equals Ω. In addition to this partition one typically tries to simultaneously identify a (potentially vector-valued) constant $q_i \in M$ associated with Ω_i (e. g. the average color on Ω_i), where M is some finite-dimensional Euclidean space such as $\mathbb{R}$ or $\mathbb{R}^{d\times d}$. Mathematically, the sought partitioning $\{\Omega_i, q_i\}_{i=1,\ldots,n}$ is defined as the minimizer of an energy composed of a bulk term $\sum_{i=1}^n E_{\text{fit}}[\Omega_i, q_i]$ and an interface energy $\frac{1}{2}\sum_{i,j=1}^n \sigma(q_i,q_j)\mathscr{H}^{d-1}(\partial^*\Omega_i \cap \partial^*\Omega_j)$,

$$E[\{\Omega_i, q_i\}_{i=1,\ldots,n}] = \begin{cases} \displaystyle\sum_{i=1}^n E_{\text{fit}}[\Omega_i, q_i] & \\ \qquad \displaystyle +\frac{\beta}{2}\sum_{i,j=1}^n \sigma(q_i,q_j)\mathscr{H}^{d-1}(\partial^*\Omega_i \cap \partial^*\Omega_j) & \text{if } \{\Omega_i\}_{i=1,\ldots,n} \text{ is a partition of } \Omega\,, \\ \infty & \text{else.} \end{cases} \tag{3}$$

The bulk term assesses how well the $\{\Omega_i, q_i\}$ fit to the image u and is of the form

$$E_{\text{fit}}[\Omega_i, q_i] = \int_{\Omega_i} f_{\text{fit}}(x, q_i)\,\mathrm{d}x \tag{4}$$

with some energy density $f_{\text{fit}} \geq 0$, while the interface energy with weight $\beta > 0$ acts as a regularization of the partition by penalizing the $(d-1)$-dimensional Hausdorff measure $\mathscr{H}^{d-1}(\partial^*\Omega_i \cap \partial^*\Omega_j)$ of the interface between segments Ω_i and Ω_j, weighted by some constant $\sigma(q_i,q_j) \geq 0$. Here ∂^* denotes the essential boundary of a measurable set [1].

The variational problem (3) is well-posed in the following weak sense. Let $\mathscr{S}$ denote the set of measurable subsets of Ω with finite perimeter, i. e. $\mathscr{H}^{d-1}(\partial^* O) < \infty$ for all $O \in \mathscr{S}$. We can identify any $\{\Omega_i, q_i\}_{i=1,\ldots,n} \in (\mathscr{S}\times M)^n$ with a piecewise constant function

$$q \in \mathrm{SBV}(\Omega; M)\,, \quad q = \sum_{i=1}^n q_i \chi_{\Omega_i}\,,$$

where χ_{Ω_i} denotes the characteristic function of Ω_i. $\mathrm{SBV}(\Omega;M)$ is the Banach space of special functions of bounded variation, i. e. Lebesgue-integrable functions $q : \Omega \to M$ whose distributional derivative can be represented as the sum

$$Dq = \nabla q \mathscr{L} + (q^+ - q^-) \otimes n_q \mathscr{H}^{d-1} \llcorner \Gamma_q$$

of a gradient ∇q continuous with respect to the Lebesgue measure $\mathscr{L}$ and a jump part concentrated on a $(d-1)$-dimensional jump set Γ_q with generalized normal n_q and jump size $(q^+ - q^-)$ [1]. The energy (3) can be expressed in terms of q as

$$E[q] = \begin{cases} \int_\Omega f_{\text{fit}}(x, q(x))\,\mathrm{d}x + \beta \int_{\Gamma_q} \sigma(q^+(x), q^-(x))\,\mathrm{d}\mathscr{H}^{d-1}(x) & \text{if } q \in \mathrm{SBV}(\Omega;M) \text{ with } \nabla q = 0, \\ \infty & \text{else.} \end{cases} \tag{5}$$

Theorem 1. *If $\sigma : M \times M \to [0,\infty)$ is symmetric, subadditive, and lower semi-continuous with $\sigma(q,q) = 0$ for all $q \in M$ and $\lim_{|q_1-q_2|\to 0} \frac{\sigma(q_1,q_2)}{|q_1-q_2|} = \infty$ and if $f_{\text{fit}}(x,\cdot)$ is non-negative and lower semi-continuous, then* (5) *admits a minimizer on $\{q \in \mathrm{SBV}(\Omega;M) \,|\, \|q\|_{L^\infty} \leq C\}$ for any $C > 0$.*

Proof. Consider a minimizing sequence q_k, $k \in \mathbb{N}$, along which the energy is uniformly bounded by $\bar{E}$. By a slight generalization of the compactness and lower semi-continuity Theorems 4.7 and 4.8 in [1], $\nabla q_k = 0$ and $\int_{\Gamma_{q_k}} \sigma(q_k^+(x), q_k^-(x))\,\mathrm{d}\mathscr{H}^{d-1}(x) < \bar{E}/\beta$ for all k together with $\|q_k\|_{L^\infty} \leq C$ imply the weak-* convergence in $\mathrm{BV}(\Omega;M)$ of a subsequence, again denoted q_k, against a $q \in \mathrm{SBV}(\Omega;M)$ with $\nabla q = 0$ and

$$\int_{\Gamma_q} \sigma(q^+(x), q^-(x))\,\mathrm{d}\mathscr{H}^{d-1}(x) \leq \liminf_{k\to\infty} \int_{\Gamma_{q_k}} \sigma(q_k^+(x), q_k^-(x))\,\mathrm{d}\mathscr{H}^{d-1}(x)\,.$$

Upon choosing another subsequence we even have pointwise convergence almost everywhere so that

$$\int_\Omega f_{\mathrm{fit}}(x, q(x))\,\mathrm{d}x \leq \liminf_{k\to\infty} \int_\Omega f_{\mathrm{fit}}(x, q_k(x))\,\mathrm{d}x$$

follows from Fatou's lemma and the lower semi-continuity of f_{fit}, which shows that q must be a minimizer. □

Remark 1. Except for the positive definiteness, σ satisfies the axioms of a metric, so it can be interpreted as a metric on the quotient space of M modulo identification of elements r, q with $\sigma(r,q) = 0$.

In applications such as grain segmentation in polycrystal images, each detected region Ω_i should reasonably have a minimum size, e. g. the size of a crystal unit cell. Since the measure of the domain Ω is finite, this implies that the number n of segments in the partition should be bounded by a constant $N \in \mathbb{N}$. In that case one has the following strong well-posedness result.

Theorem 2. *Under the conditions of the previous theorem, energy* (3) *admits a minimizer among all partitions* $\{\Omega_i, q_i\}_{i=1,\ldots,n} \in \bigcup_{n=1}^N (\mathscr{S} \times M)^n$ *with* $|q_i| \leq C$ *for* $i = 1, \ldots, n$.

Proof. Consider a minimizing sequence $\{\Omega_i^k, q_i^k\}_{i=1,\ldots,n_k}$, $k \in \mathbb{N}$, and identify each element of the minimizing sequence with the corresponding SBV function q^k. As in the proof of the previous theorem, upon taking a subsequence the q^k converge weakly-* and pointwise almost everywhere against a $q^* \in \mathrm{SBV}(\Omega;M)$. Since the q^k all have at most N values, so does q^*. Denoting the values by $q_1^*, \ldots, q_n^*$ we may thus set $\Omega_i = \{x \in \Omega \,|\, q^*(x) = q_i^*\}$, $i = 1, \ldots, n$. Now

$$E[\{\Omega_i, q_i^*\}_{i=1,\ldots,n}] = E[q^*] \leq \liminf_{k\to\infty} E[q^k] = \liminf_{k\to\infty} E[\{\Omega_i^k, q_i^k\}_{i=1,\ldots,n_k}]$$

so that $\{\Omega_i, q_i^*\}_{i=1,\ldots,n}$ must be a minimizer. □

Remark 2. The necessary L^∞ bound can in most applications be obtained variationally from the energy. For instance, if the bulk and interfacial energy density satisfy

- there exist $\bar{q} \in M$, $\bar{f} > 0$, and $\varphi : [0,\infty) \to [0,\infty)$ with $\lim_{\rho\to\infty} \varphi(\rho) = \infty$ such that $f_{\mathrm{fit}}(x, \bar{q}) \leq \bar{f}$ and $f_{\mathrm{fit}}(x, \cdot) \geq \varphi(|\cdot|)$ for all $x \in \Omega$,
- for any compact $M_0 \subset M$ there is a compact superset $M_1 \subset M$ with $\sup\{\sigma(q_1,q_2) \,|\, q_1, q_2 \in M_0\} \leq \inf\{\sigma(q_1,q_2) \,|\, q_1 \in M_0, q_2 \in M \setminus M_1\}$ (i. e. the diameter of M_0 is smaller than the distance from $M \setminus M_1$ to M_0),

then a truncation argument leads to a minimizing sequence uniformly bounded in L^∞: Upon taking a subsequence we may assume each sequence element to have the same number n of segments. Furthermore, by applying a diagonal argument, $|q_i^k|$ may be assumed to be monotone. Denote by I the set of indices i belonging to unbounded sequences q_i^k, and set $J = \{1, \ldots, n\} \setminus I$. Now set $M_0 = \{q \in M \,|\, |q| \leq \max(|\bar{q}|, \sup_{i\in J,k} |q_i^k|)\}$ and let $M_1 \subset M$ be the corresponding set from the condition on σ. It is sufficient to consider the minimizing subsequence along which $q_i^k \in \{q \in M \,|\, \varphi(|q|) > \bar{f}\} \setminus M_1$ for all $i \in I$. However, replacing the q_i^k, $i \in I$, all by $\bar{q}$, the energy decreases along the sequence, and at the same time all q_i^k lie in the compact set M_0.

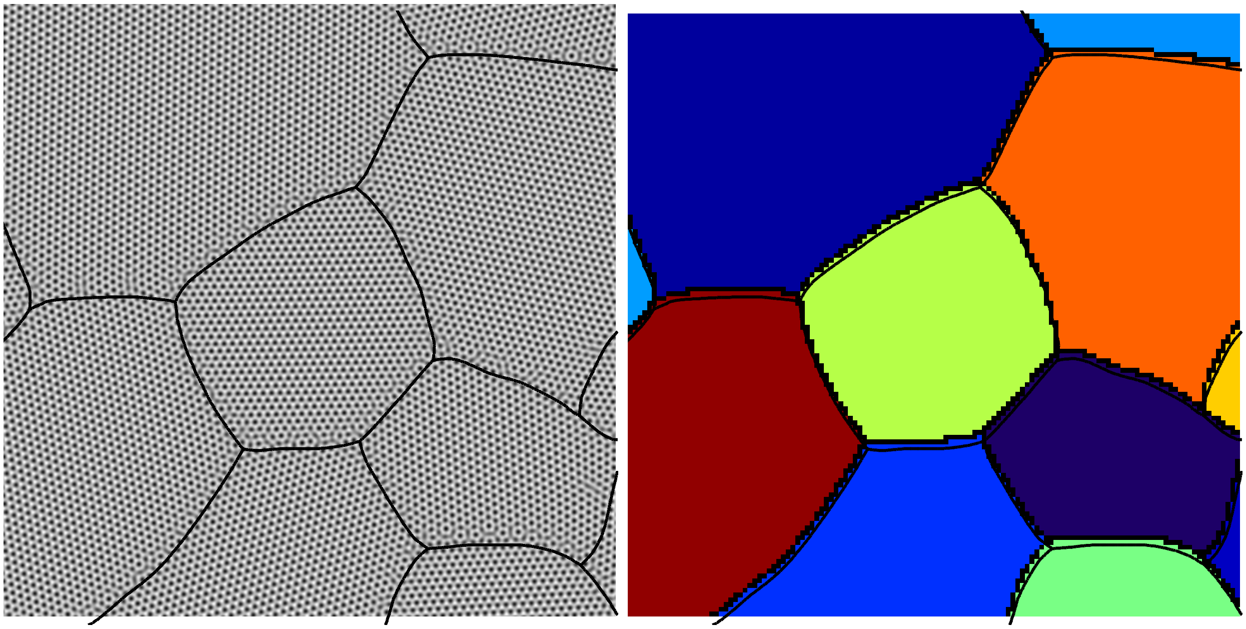

Figure 1: Left: Two-dimensional polycrystal image (for a hexagonal crystal lattice) with grains, i. e., the regions of constant lattice orientation indicated by black lines. Right: Automatically segmented grains with color-coding according to lattice orientation.

2.2 Grain segmentation in polycrystal images

In this work we are concerned with a particular multilabel segmentation, the segmentation of a polycrystal image into the different crystal grains.

A polycrystal is the typical state of crystalline materials. In such materials the constituents (e. g. the atoms) are locally arranged in a regular Bravais lattice $\{n_1v_1+n_2v_2+n_3v_3\,|\,n_1,n_2,n_3\in\mathbb{Z}\}$ with lattice vectors $v_1,v_2,v_3\in\mathbb{R}^3$. The orientation of this lattice is usually only constant on subsets of the material domain $\Omega\subset\mathbb{R}^3$, which form a partition of Ω and are called crystal grains (see Fig. 1 for a two-dimensional version). Within each grain $\Omega_i\subset\Omega$, the crystal lattice is (almost) a perfect Bravais lattice with lattice vectors R_iv_1,R_iv_2,R_iv_3 for some rotation $R_i\in\mathrm{SO}(3)$, where $\mathrm{SO}(3)$ denotes the special orthogonal group in $\mathbb{R}^3$. The interfaces where two grains of different orientation meet are called grain boundaries. Here the preferred lattice structure of the atoms is disturbed, which causes the grain boundary to be associated with a physical energy. This energy depends mainly on the difference between the orientations $R\in\mathrm{SO}(3)$ and $Q\in\mathrm{SO}(3)$ of the two adjacent grains. For R and Q sufficiently close the grain boundary typically shows up as a string of equispaced point defects in the crystal (so-called dislocations) from which an expression for the interfacial energy density σ can be derived, the so-called the Read–Shockley law [29]

$$\sigma(R,Q)=\gamma_{\mathrm{RS}}(\theta_{R,Q})\quad\text{with}\quad\gamma_{\mathrm{RS}}(\theta)=\theta(A-\log\theta)\,.$$

A represents a material constant and $\theta_{R,Q}$ the misorientation angle, i. e. the smallest angle by which one grain needs to be rotated to have the same orientation as the other. If $\mathrm{P}\subset\mathrm{SO}(3)$ denotes the point group of the crystal, i. e. the set of rotations which leave the crystal lattice invariant, then this misorientation angle can be computed as

$$\theta_{R,Q}=\min_{P\in\mathrm{P}}|\arg(\lambda_{RPQ^{-1}})|\,,$$

where $\lambda_{RPQ^{-1}}$ denotes one of the two complex eigenvalues of $RPQ^{-1}\in\mathrm{SO}(3)$ and $\arg\equiv -i\log$ denotes the complex angle of a complex number with absolute value one, $\arg e^{i\theta}=\theta \mathrm{mod} 2\pi$, so that $\arg(\lambda_{RPQ^{-1}})$ has the interpretation of the rotation angle of RPQ^{-1}.

Our aim is to automatically identify the grains $\Omega_i \subset \Omega$ and their lattice orientations $R_i \in \mathrm{SO}(3)$ from a three-dimensional gray scale image $u : \Omega \to \mathbb{R}$ of the crystalline material as in Fig. 1 left. To this end we view the task as a vector-valued multilabel segmentation problem and borrow the fidelity term

$$E_{\mathrm{fit}}[\Omega_i, R_i] = \int_{\Omega_i} \sum_{k=1}^{K} (u(x) - u(x + R_i x_k))^2 \,\mathrm{d}x \tag{6}$$

from [3]. Here, $x_1, \ldots, x_K \in \mathbb{R}^3$ represent a fixed stencil of vectors describing the unrotated lattice, for instance the lattice vectors or the position vectors of all nearest neighbors of a central atom. If R_i correctly describes the orientation of grain Ω_i, then the image u locally is periodic with respect to translation by $R_i x_k$, $k = 1, \ldots, K$, so that the integrand is zero (up to image noise or illumination changes); else the integral will be strictly positive. For the regularization of the detected grain boundaries we will here choose a surface tension σ imitating the actual physical surface tension,

$$\sigma(R, Q) = \gamma(\theta_{R,Q}), \tag{7}$$

where $\gamma : [0, \pi] \to \mathbb{R}$ is concave with $\gamma(0) = 0$. For algorithmic reasons we will not exactly choose $\gamma = \gamma_{\mathrm{RS}}$ but instead

$$\gamma(\theta) = \begin{cases} 0 & \text{if } \theta = 0, \\ \gamma_{\max}\left[1 - (1 - \gamma_{\min})(\sin(\frac{\theta}{2})/\sin(\frac{\theta_{\max}}{2}) - 1)^2\right] & \text{if } 0 < \theta < \theta_{\max}, \\ 1 & \text{else} \end{cases} \tag{8}$$

(where we choose constants $\gamma_{\max} = 1$, $\gamma_{\min} = \frac{1}{10}$, $\theta_{\max} = \frac{\pi}{3}$). In more detail, we will exploit that γ is piecewise quadratic in $\sin\frac{\theta}{2}$ in order to obtain an explicit formula for a particular step of the algorithm, however, $\gamma = \gamma_{\mathrm{RS}}$ could also be implemented with only slightly more computational effort. Our motivation for imitating the physical interface energy is twofold:

- The physical interfacial energy density σ determines the angles at which crystal grains meet at triple or higher order junctions. The same holds true for the relation between the σ used in our segmentation energy and the triple junctions of the detected grains $\Omega_1, \ldots, \Omega_n$. To better capture the correct angles our choice of σ should be close to the physically correct one.
- In our algorithm, a grain will sometimes become visible as a collection of multiple adjacent regions Ω_i with almost the same orientations (these grains will then be merged together at a later stage during the algorithm). If two such regions form a triple junction with a region of different orientation, the configuration of this triple junction should be as if the two similar grains actually were just one grain, i. e. one should see an almost smooth grain boundary. This can only be achieved if regions with similar orientations have small interfacial energy.

Since the interfacial energy density (7) can readily be extended from $\mathrm{SO}(3)$ to a metric on the quotient space of $\mathbb{R}^{3\times3}$ modulo the crystal point group P, and since f_{fit} can be extended from $\mathrm{SO}(3)$ to $\mathbb{R}^{3\times3}$ by ∞, as a corollary of Theorem 2 we immediately obtain the following.

Corollary 3. *Let $u : \Omega \to \mathbb{R}$ be continuous and $\gamma : [0, \pi] \to [0, \infty)$ be concave with $\gamma(0) = 0$. The energy* (3) *with fidelity term* (6) *and interfacial energy density* (7) *admits a minimizer in $\bigcup_{n=1}^{N} (\mathscr{S} \times \mathrm{SO}(3))^n$.*

Of course, the model can also be adapted to images of two-dimensional polycrystals. In that case, $\Omega \subset \mathbb{R}^2$ and the crystal orientations R_i are in $\mathrm{SO}(2)$. We will deal with the two-dimensional case alongside the more general 3D case.

2.3 Minimization procedure

We seek a configuration $\{\Omega_i, R_i\}_{i=1,\ldots,n}$ minimizing (3) with (6) and (7). This energy depends on both the partition $\{\Omega_i\}_{i=1,\ldots,n}$ and the rotations $\{R_i\}_{i=1,\ldots,n}$ in a complicated way in both the fitting term E_{fit} and the surface energy term. In particular for large images u, the evaluation of the fitting term E_{fit} is very costly, since it interpolates the (discrete) image u at $x + Rx_k$ for all pixels x and stencil vectors x_k. Furthermore, the representation and update of the grain boundaries should be done very efficiently. As we are interested only in minimizers of (3) and not in the dynamics of an associated flow, we proceed via an alternating minimization scheme consisting of two pieces: (1) L^2 gradient descent of (3) with respect to the partition $\{\Omega_i\}_{i=1,\dots,n}$ and with $\{R_i\}_{i=1,\dots,n}$ held fixed and (2) update of the grain orientations $\{R_i\}_{i=1,\dots,n}$ with the partition $\{\Omega_i\}_{i=1,\dots,n}$ fixed. During the latter orientation update, any two adjacent grains Ω_i, Ω_j are interpreted as one if their orientations R_i, R_j coincide, and before the next gradient flow of the interfaces, such grains are merged into a new grain covering the region $\Omega_{ij} = (\Omega_i \cup \Omega_j)$. These minimization procedures are described completely in Sections 3.1 and 3.2.

3 Minimization via alternating redistancing dynamics and Augmented Lagrangian method

This section describes the algorithm used to perform the multilabel segmentation of the crystal grains. As previously mentioned, we alternate between updating the partition, as detailed in Section 3.1, and updating the orientations, which will be explained in Section 3.2. The connection of both parts is described in Section 3.3.

3.1 Redistancing dynamics for unequal energy curvature flow with external potential field

This part of the minimization procedure seeks to decrease the energy (3) via a local update of the partition $\{\Omega_i\}_{i=1,\dots,n}$, with the orientations $\{R_i\}_{i=1,\dots,n}$ held fixed. In this case, (3) simplifies to

$$E[\{\Omega_i\}_{i=1,\dots,n}] = \begin{cases} \displaystyle\sum_{i=1}^{n} \int_{\Omega_i} f_i(x)\,\mathrm{d}x + \frac{\beta}{2} \sum_{i,j=1}^{n} \sigma_{ij} \mathscr{H}^2(\partial^*\Omega_i \cap \partial^*\Omega_j) & \text{if } \{\Omega_i\}_{i=1,\dots,n} \text{ is a partition of } \Omega\,, \\ \infty & \text{else,} \end{cases}$$

where $f_i(x) = f_{\text{fit}}(x, R_i)$ are known functions defined on all of Ω and the $\sigma_{ij} = \sigma_{ji} = \sigma(R_i, R_j)$ are constants. L^2 gradient descent on this energy gives rise to the interface normal velocities

$$v_{n_{ij}}(x) = \big((f_i(x) - f_j(x)) + \beta \sigma_{ij} \kappa_{ij} \big), \tag{9}$$

for $x \in (\partial^*\Omega_i \cap \partial^*\Omega_j)$, where n_{ij} is the unit normal vector pointing outwards from Ω_i into Ω_j and κ_{ij} is the signed mean curvature of $(\partial^*\Omega_i \cap \partial^*\Omega_j)$ at x.

The algorithm we employ builds on the signed distance function-based diffusion-generated motion algorithm [20]. The key pieces behind this algorithm are (1) the implicit representation of the sets Ω_i as the zero super level set of the signed distance function

$$d_i(x) = \begin{cases} \inf_{y \in \partial^*\Omega_i} |x - y|, & x \in \Omega_i, \\ -\inf_{y \in \partial^*\Omega_i} |x - y|, & x \notin \Omega_i, \end{cases}$$

and (2) the observation that convolution of the signed distance function with a spherically-symmetric kernel moves the zero level set of $d_i(x)$ a distance proportional to the mean curvature of the interface at x. This observation was first made for the characteristic function of sets, giving rise to the well-known Merriman–Bence–Osher threshold dynamics scheme [23].

Combining Algorithm 1 and Algorithm 6 of Esedoḡlu et. al's work [20], one obtains our Algorithm 1 for the evolution described by (9) in the case that $\sigma_{ij} \equiv 1$.

Two further extensions to Algorithm 1 are needed. The algorithm presented is useful for only a limited number of grains (small n), as each grain is represented by a signed distance function on all of Ω. The storage and computational requirements are then proportional to n, which rapidly becomes infeasible for systems with many grains. This difficulty can be alleviated by maintaining many spatially-separated grains with the same signed distance function [16]. A swapping procedure which prevents two (initially spatially-separated) grains from coalescing during the evolution is detailed there.

The other element missing from Algorithm 1 is the allowance for unequal surface energies σ_{ij} as specified by (9). This extension is already known for the signed distance function-based diffusion-generated motion algorithm [17]. For the convenience of the reader, a complete description of the algorithm we employ is given in Algorithm 2.

Algorithm 1 Basic algorithm for $v_{n_{ij}}(x) = (f_i(x) - f_j(x)) + \beta \kappa_{ij}$.

Given initial sets $\{\Omega_i^0\}_{i=1,\ldots,n}$ through their signed distance functions $\{d_i^0(x)\}_{i=1,\ldots,n}$, fixed external fields $\{f_i(x)\}_{i=1,\ldots,n}$, and a time step $\tau > 0$, generate the sets $\{\Omega_i^s\}_{i=1,\ldots,n}$ at discrete times $t = s\tau$ by repeating steps 1–3 for $i = 1, \ldots, n$.

```
1. CONVOLVE: Compute $A_i(x) = \left(G_{\beta\tau} * d_i^{s-1}\right)(x) - 2\tau f_i(x)$, where $G_{\beta\tau}(x) = \frac{1}{(4\pi\beta\tau)^{d/2}} e^{-|x|^2/(4\beta\tau)}$.
2. REDISTRIBUTE: $B_i(x) = \frac{1}{2}\left(A_i(x) - \max_{k \neq i} A_k(x)\right)$.
3. REDISTANCE: Compute the signed distance function $d_i^s(x)$ as the zero super level set of $B_i$.
```

In practice, the fitting energy densities f_{fit} defined by (4) and (6) vary quite rapidly, as errors in fitting are penalized more strongly on atoms than in the spaces between atoms. This occurs because a randomly-selected orientation asks the fitting term to compare gray values at x to randomly-selected nearby image locations. The atomic gray intensities occur with less frequency than the gray intensities corresponding to non-atomic positions, so a poorly chosen orientation will feel a greater penalty at atoms. In order to promote smoothness of interfaces and to allow the evolution to take larger time steps, we redefine

$$f_i(x) = \left(\sum_{k=1}^{K} (u(x) - u(x + R_i x_k))^2 \, \mathrm{d}x \right) * G_\rho ,$$

with $\rho \propto ||x_k||^2$ chosen to smooth f_{fit} over the predicted atomic spacing, $||x_k||$.

In order to perform the convolution steps efficiently, we enforce periodic boundary conditions and utilize the fast Fourier transform. The locality of the energy ensures that this choice of boundary condition only affects the numerical results near the boundary of the image domain Ω. Other choices of boundary condition would require an appropriate modification of the fitting term to prevent evaluations of $u(y)$ for $y \notin \Omega$.

3.2 Updating the orientations

Here, we minimize the energy with respect to the orientations $\{R_i\}_{i=1,\ldots,n}$ for a fixed partition $\{\Omega_i\}_{i=1,\ldots,n}$. In that case, the energy reduces to

$$E[\{R_i\}_{i=1,\ldots,n}] = \sum_{i=1}^{n} E_{\text{fit}}[\Omega_i, R_i] + \frac{\beta}{2} \sum_{i,j=1}^{n} \sigma(R_i, R_j) \mathscr{H}^2(\Gamma_{ij}) ,$$

where we abbreviated $\Gamma_{ij} = \partial^* \Omega_i \cap \partial^* \Omega_j$. Note that the energy is only defined on $\mathrm{SO}(3)^n$, a lower-dimensional subset of $(\mathbb{R}^{3\times3})^n$. Furthermore, it is non-smooth whenever two neighboring grains Ω_i and Ω_j have the same orientation and thus $\sigma(R_i, R_j) = 0$. Finally, the fitting term is very costly to evaluate for large images. In particular, the image u has to be interpolated at all points $x + R_i x_k$, $k = 1, \ldots, K$, for all pixels x. For an efficient minimization, we employ several instruments:

- Instead of performing a minimization on $\mathrm{SO}(3)^n$ or even $(\mathbb{R}^{3\times3})^n$ we will rephrase and minimize the energy in terms of unit quaternions, which allow a more efficient and much less redundant description of lattice orientations.
- We will employ a split Augmented Lagrangian method which allows to separate the smooth fitting term from the non-smooth interfacial energy, each of which can then be minimized using separate, adapted and thus efficient procedures.
- The smooth energy terms will be minimized by a higher order, quickly converging Riemannian optimization method, while the non-smooth terms can be explicitly minimized.

Algorithm 2 Complete algorithm for $v_{n_{ij}}(x) = (f_i(x) - f_j(x)) + \beta\sigma_{ij}\kappa_{ij}$.

Given: Initial sets $\{\Omega_i^0\}_{i=1,\ldots,n}$, surface energies $\{\sigma_{ij}\}_{i,j=1,\ldots,n}$, external fields $\{f_i(x)\}_{i=1,\ldots,n}$, a time step $\tau > 0$, and positive constants ε and H.
Let $G_\alpha(x) = \frac{1}{(4\pi\alpha)^{d/2}} e^{-|x|^2/(4\alpha)}$, define $\phi_{i;jk} = \sigma_{ij} + \sigma_{ik} - \sigma_{jk}$, $\phi^\star = \max_{i,j,k} \phi_{i;jk}$ and

$$\psi(x, d_j^s, \phi_j(x)) = \left(\frac{\phi_j(x)}{\phi^*} \left(G_{\beta\phi^*\tau} * d_j^s \right)(x) + \left(1 - \frac{\phi_j(x)}{\phi^*} \right) d_j^s(x) \right) - 2\tau f_j(x).$$

Form a partition $\{\Xi_j^0\}_{j=1,\ldots,\bar{n}}$ such that each part Ξ_j^0 is the union of a subset of (spatially separated) parts $\{\Omega_i^0\}_{i=1,\ldots,n}$.
Set d_j^0 to be the signed distance function to the collection Ξ_j^0.
Given $j \in \{1,\ldots,\bar{n}\}$, we use the notation $\hat{j} = \hat{j}(x)$ to indicate the index $\hat{j} \in \{1,\ldots,n\}$ such that $\Omega_{\hat{j}}^s \subseteq \Xi_j^s$ and $\operatorname{dist}(x, \Omega_{\hat{j}}^s) \leq \operatorname{dist}(x, \Omega_i^s)$ for all i with $\Omega_i^s \subseteq \Xi_j^s$.
Update the collections $(\Xi_j^s)_{j=1,\ldots,m}$ at discrete times $t = s\tau$ by repeating steps 1–4.

1. UPDATE: For each grid location x, define $R(x) = \{i : d_i^s(x) > -\varepsilon\}$, and let $r(x) = \#R(x)$.
   For $i \notin R(x)$, set $A_i(x) = -\infty$.
   (a) If $R(x) = \{i\}$, set $A_i(x) = d_i^s(x)$.
   (b) If $R(x) = \{i, j\}$, set $A_i(x) = \psi(x; d_i^s, \sigma_{\hat{i}\hat{j}})$, and $A_j(x) = \psi(x; d_j^s, \sigma_{\hat{i}\hat{j}})$.
   (c) If $R(x) = \{i, j, k\}$, set $A_i(x) = \psi(x; d_i^s, \phi_{\hat{i};\hat{j}\hat{k}})$, $A_j(x) = \psi(x; d_j^s, \phi_{\hat{j};\hat{i}\hat{k}})$, and $A_k(x) = \psi(x; d_k^s, \phi_{\hat{k};\hat{i}\hat{j}})$.
   (d) If $r(x) > 3$,
      - For each $i \in R$, compute

        $$T_i(x) = \frac{1}{\binom{r(x)-1}{2}} \sum_{\substack{j,k \in R \setminus \{i\} \\ j<k}} \psi(x; d_i^s, \phi_{\hat{i};\hat{j}\hat{k}}).$$

      - Next compute

        $$w_i(x) = \begin{cases} \varepsilon, & T_i(x) < -H \\ \varepsilon + (1-\varepsilon)\left(\frac{1}{2} + \frac{T_i(x)}{2H}\right), & |T_i(x)| < H \\ 1, & T_i(x) > H. \end{cases}$$

      - Set

        $$A_i(x) = \frac{\displaystyle\sum_{\substack{j,k \in R \setminus \{i\} \\ j<k}} w_i w_j w_k \, \psi(x; d_i, \phi_{\hat{i};\hat{j}\hat{k}})}{\displaystyle\sum_{\substack{j,k \in R \setminus \{i\} \\ j<k}} w_i w_j w_k}.$$

2. REDISTRIBUTE: For $i = 1, \ldots, m$, construct

   $$B_i(x) = \frac{1}{2}\left(A_i(x) - \max_{j \neq i} A_j(x) \right)$$

   to remove overlaps and vacuums from the previous step.
3. REDISTANCE: For $i = 1, \ldots, m$, set $C_i(x)$ to be the signed distance function to the zero-level set of $B_i(x)$.
4. SWAP: As necessary, swap appropriate grains between signed distance functions C_i to ensure a minimum separation between grains associated with the same signed distance function. Redistance around swapped grains and denote the resulting signed distance functions as $d_i^{s+1}(x)$.

- For the higher order optimization, the Hessian will partly be approximated by a quasi-Newton update, which in turn only requires a (fast) piecewise linear interpolation of the crystal image u.
- The quadratic approximation to the smooth energy terms is only updated when necessary, thereby saving many fitting term evaluations.

The below paragraphs provide the details of the above-mentioned tools.

Representing orientations in the Lie group of unit quaternions. Unit quaternions have long been established as an efficient representation of rotations or orientations (see e. g. [28] and references therein). For the reader's convenience we briefly summarize all relevant properties. A quaternion can be defined as a four-dimensional vector,

$$q = (q_0, \vec{q}) \in \mathbb{R} \times \mathbb{R}^3 .$$

We shall denote the set of quaternions by $\mathbb{H}$. Together with the usual vector addition, the product

$$pq = (p_0, \vec{p})(q_0, \vec{q}) = (p_0 q_0 - \vec{p} \cdot \vec{q}, p_0 \vec{q} + q_0 \vec{p} + \vec{p} \times \vec{q})$$

turns $\mathbb{H}$ into an algebra. The conjugate of a quaternion q is defined as $\bar{q} = (q_0, -\vec{q})$ and its norm as $\|q\| = \sqrt{\bar{q}q} = \sqrt{q_0^2 + |\vec{q}|^2}$. Taking $(1, \vec{0})$ as the unit element, the multiplicative inverse of q is given by $q^{-1} = \frac{\bar{q}}{\|q\|^2}$ so that $\mathbb{H}$ forms a multiplicative group. The norm satisfies the property $\|pq\| = \|p\| \, \|q\|$ so that the set of unit quaternions, $\mathbb{U} = \{q \in \mathbb{H} \,|\, \|q\| = 1\}$, represents a subgroup. Every rotation $R \in \mathrm{SO}(3)$ can be described by a rotation axis $n_R \in \mathbb{R}^3$ and a rotation angle ω_R, and the mapping

$$(\omega_R, n_R) \mapsto (\cos \tfrac{\omega_R}{2}, n_R \sin \tfrac{\omega_R}{2})$$

thus provides an identification of a unit quaternion with a rotation (since $q \in \mathbb{U}$ and $-q$ describe the same rotation, the unit quaternions $\mathbb{U}$ represent a two-fold covering of $\mathrm{SO}(3)$). This identification is consistent with the composition of rotations, i. e. $R(p)R(q) = R(pq)$, where $R(q) \in \mathrm{SO}(3)$ denotes the rotation associated with $q \in \mathbb{U}$. Vectors $v \in \mathbb{R}^3$ can be identified with the quaternions $(0, v) \in \mathbb{H}$, and the rotation of v by $R(q)$ can be expressed via the relation

$$(0, R(q)v) = q\,(0, v)\,\bar{q} .$$

The advantage of parameterizing rotations via unit quaternions instead of rotation matrices lies in the fewer redundant degrees of freedom. Furthermore, compared to other representations of rotations such as axis-angle pairs, the unit quaternion description contains no singular points where the parameterization degenerates. For these reasons our algorithm will work on the set of unit quaternions so that we reformulate our energy as $E : \mathbb{U}^n \to \mathbb{R}$,

$$E[\{q_i\}_{i=1,\dots,n}] = \sum_{i=1}^{n} E_{\mathrm{fit}}[\Omega_i, q_i] + \frac{\beta}{2} \sum_{i,j=1}^{n} \sigma(q_i, q_j) \mathscr{H}^2(\Gamma_{ij})$$

with

$$E_{\mathrm{fit}}[\Omega_i, q_i] = \int_{\Omega_i} \sum_{k=1}^{K} (u(x) - u(x + q_i x_k \bar{q}_i))^2 \,\mathrm{d}x ,$$

$$\sigma(o, q) = \gamma\big(\min_{p \in \mathrm{P}} 2 \arccos |(op\bar{q})_0|\big) = \gamma\big(\min_{p \in \mathrm{P}} 2 \arcsin \big|\overrightarrow{(op\bar{q})}\big|\big) ,$$

where we used the implicit identification $v \equiv (0, v)$ for all $v \in \mathbb{R}^3$ and the arrow indicates the vector part of a quaternion. This time, P is the set of all unit quaternions representing a rotation from the crystal point group.

The set of unit quaternions is not a vector space. The optimization for the orientations $q_1, \dots, q_n$ can either be done in $\mathbb{H}^n$, using standard constrained optimization techniques with the additional constraints $\|q_i\| = 1$, $i = 1, \dots, n$, or it can be performed directly in the nonlinear space $\mathbb{U}$ so that the constraints are intrinsically satisfied. We will choose the latter approach since it reduces the dimensionality of the problem and therefore is more efficient. To this end we will exploit the Lie group structure of $\mathbb{U}$. Interpreting $\mathbb{U}$ to be embedded in $\mathbb{H} \equiv \mathbb{R} \times \mathbb{R}^3$ one can identify the tangent space to $\mathbb{U}$ at $(1,\vec{0})$ with three-dimensional vectors, $T_{(1,\vec{0})}\mathbb{U} = \{(0,v)\,|\,v\in\mathbb{R}^3\}$. The tangent space at $q\in\mathbb{U}$ is given by $T_q\mathbb{U} = qT_{(1,\vec{0})}\mathbb{U}$. All tangent spaces can be equipped with the (bi-invariant) inner product induced from $\mathbb{R}\times\mathbb{R}^3$, turning $\mathbb{U}$ into a Riemannian manifold. The exponential map $\exp_q : T_q\mathbb{U}\to\mathbb{U}$ and logarithm $\log_q : \mathbb{U}\to T_q\mathbb{U}$ on $\mathbb{U}$ can then be epressed as

$$\exp_q(v) = q\exp_{(1,\vec{0})}(\bar{q}v)\,, \qquad \log_q(p) = q\log_{(1,\vec{0})}(\bar{q}p)\,,$$

where the exponential map and logarithm with respect to the base point $(1,\vec{0})$ are given by

$$\exp_{(1,\vec{0})}(0,v) = (\cos|v|, \tfrac{v}{|v|}\sin|v|)\,, \qquad \log_{(1,\vec{0})}(q_0,\vec{q}) = (0, \tfrac{\vec{q}}{|\vec{q}|}\arccos q_0)\,.$$

Parallel transport of a tangent vector $v\in T_q\mathbb{U}$ to $T_p\mathbb{U}$ can be performed as $p\bar{q}v$. These operations will later be used within a Riemannian optimization method.

Updating orientations by the split Augmented Lagrange Method. While the energy E_{fit} is smooth in the orientations q_i (assuming a smooth input image u), the surface tension σ has a singularity on $\{(q,q)\,|\,q\in\mathbb{U}\}$ so that derivative-based minimization methods are inappropriate. For such combinations of smooth and non-smooth terms, splitting type methods in which both terms are treated separately are typically very efficient [21]. We will choose a split Augmented Lagrangian method. The classical Augmented Lagrange method for a constrained minimization problem

$$\text{minimize } f(x) \quad \text{subject to } g(x) = 0$$

augments the Lagrangian $L(x,\lambda) = f(x) + \lambda\cdot g(x)$ with Lagrange multiplier λ by a penalty term of the form $\frac{\mu}{2}|g(x)|^2$ with $\mu > 0$ large enough, $L_A(x,\lambda) = L(x,\lambda) + \frac{\mu}{2}|g(x)|^2$. The central observation now is that the first order optimality conditions of the constrained optimization problem are equivalent to $g(x) = 0$ and $x = \operatorname{argmin}_x L_A(x,\lambda)$, which leads to the following fixed point iteration for x and λ,

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x L_A(x,\lambda^k)\,,\\ \lambda^{k+1} &= \lambda^k + \mu g(x^{k+1})\,, \end{aligned}$$

where the optimization for x does not have to be exact. If f is convex and g linear, then the reformulation $b_k = \lambda_k/\mu$ turns this method into the Bregman iteration which is widely used in the image processing community [27],

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x f(x) + \tfrac{\mu}{2}|g(x)+b^k|^2\,,\\ b^{k+1} &= b^k + g(x^{k+1})\,. \end{aligned}$$

Now any minimization problem of the form $\min_z f_1(z) + f_2(\Psi(z))$ with f_1 and Ψ smooth but f_2 nonsmooth can be reformulated as the constrained problem of minimizing $f(x) := f_1(z) + f_2(\psi)$ for $x\equiv(z,\psi)$ subject to $g(x) := \Psi(z) - \psi = 0$. If the Augmented Lagrange method is applied to this constrained problem, the update of x^{k+1} can be split in the sense of an alternating minimization into an update for z and a separate update for ψ, resulting in a split Augmented Lagrangian scheme. The advantage is that the update for z only involves smooth terms amenable to gradient-based techniques, while the update of ψ typically is a very simple and fast operation. For one-homogeneous f_2, the split Augmented Lagrange method often seems to yield sufficient energy decrease already after a few iterations.

Let us introduce the set $\mathscr{I} = \{(i,j)\,|\,i>j,\,\mathscr{H}^2(\Gamma_{ij})\neq 0\}$ of index pairs belonging to grain interfaces. For the purpose of applying the split Augmented Lagrange method we rewrite our objective energy as

$$E[\{q_i\}_{i=1,\dots,n},\{\psi_{ij}\}_{(i,j)\in\mathscr{I}}] = \sum_{i=1}^n E_{\text{fit}}[\Omega_i,q_i] + \beta\sum_{(i,j)\in\mathscr{I}}\tilde{\sigma}(\psi_{ij})\mathscr{H}^2(\Gamma_{ij}) \quad \text{with } \tilde{\sigma} = \gamma(2\arcsin|\cdot|)\,,$$

which is to be minimized under the constraints

$$\psi_{ij} - \overrightarrow{(q_i p_{ij}\bar{q}_j)} = 0$$

for $(i,j) \in \mathscr{I}$. Here, $p_{ij} \in \mathrm{P}$ shall denote the point group element minimizing $\left|\overrightarrow{(q_i p_{ij} \bar{q}_j)}\right|$, which we assume fixed for the moment. The corresponding split Augmented Lagrange or Bregman method then reads

$$\begin{aligned}
\{q_i^{k+1}\}_i &= \operatorname*{argmin}_{\{q_i\}_i} \sum_{i=1}^n E_{\mathrm{fit}}[\Omega_i, q_i] + \tfrac{\mu}{2}\beta \sum_{(i,j)\in\mathscr{I}} \mathscr{H}^2(\Gamma_{ij}) \left|\psi_{ij}^k - \overrightarrow{(q_i p_{ij} \bar{q}_j)} + b_{ij}^k\right|^2 , \\
\psi_{ij}^{k+1} &= \operatorname*{argmin}_{\psi_{ij}} \tilde{\sigma}(\psi_{ij}) + \tfrac{\mu}{2}\left|\psi_{ij} - \overrightarrow{(q_i^{k+1} p_{ij} \bar{q}_j^{k+1})} + b_{ij}^k\right|^2 & \forall (i,j) \in \mathscr{I} , \\
b_{ij}^{k+1} &= b_{ij}^k + \psi_{ij}^{k+1} - \overrightarrow{(q_i^{k+1} p_{ij} \bar{q}_j^{k+1})} & \forall (i,j) \in \mathscr{I} .
\end{aligned}$$

For the first line we will use a combined Riemannian Newton and BFGS trust region method, while the other two updates will be performed explicitly. Furthermore, at the end of each iteration we update

$$p_{ij} = \operatorname*{argmin}_{p\in\mathrm{P}} \left|\overrightarrow{(q_i p \bar{q}_j)}\right| .$$

Since the energy is not convex we may only expect convergence against a local minimizer $\{q_i\}_{i=1,\ldots,n}$, however, grains with suboptimal orientation q_i will quickly be shrunk away by the mean curvature motion part of the algorithm so that only grains with the correct orientations survive.

The iteration is stopped if the supremum norm of the constraint decreases below a fixed threshold ϑ (set to 10^{-1} in our experiments),

$$\max_{(i,j)\in\mathscr{I}} \| \psi_{ij}^{k+1} - \overrightarrow{(q_i^{k+1} p_{ij} \bar{q}_j^{k+1})} \|_\infty \le \vartheta ,$$

which indicates that the Lagrange multiplier has well converged and thus that the optimality conditions are sufficiently achieved.

Combined Riemannian Newton and BFGS trust region method for smooth terms. Within the split Augmented Lagrangian method, the smooth minimization of

$$F[\{q_i\}_i] := \sum_{i=1}^n E_{\mathrm{fit}}[\Omega_i, q_i] + \tfrac{\mu}{2}\beta \sum_{(i,j)\in\mathscr{I}} \mathscr{H}^2(\Gamma_{ij}) \left|\psi_{ij} - \overrightarrow{(q_i p_{ij} \bar{q}_j)} + b_{ij}\right|^2$$

for given fixed $\psi_{ij}, p_{ij}, b_{ij}$ will be performed by a trust region method with a quadratic model of the objective. In detail, our implementation uses the very robust algorithm [9, Alg. 7.3.1-4]. The quadratic approximation of the objective function ensures a fast, superlinear convergence and furthermore is used to identify the decreasing directions at saddle points (of which the energy possesses several ones).

As previously mentioned, the trust region method is applied on the Riemannian manifold $\mathbb{U}^n$, turning it into a Riemannian optimization method [30]. For the reader's convenience and to show how the Riemannian structure is exploited, we briefly state the trust region iteration. In each step we minimize a quadratic approximation $m_k(\{s_i\}_i)$ to $F[\{\exp_{q_i^k}(s_i)\}_i]$ within a spherical trust region of radius Δ. If the step decreases the energy sufficiently, it is accepted, and the validity radius Δ of the model is increased, else it is decreased. In detail, fixing parameters $\vartheta_1 < 1 < \vartheta_2$, $\eta_1 < \eta_2 < 1$, the iteration reads

```
for $k = 1, 2, \dots$
  define the quadratic model $m_k(\{s_i\}_i) = F[\{q_i^k\}_i] + DF[\{q_i^k\}_i]\{s_i\}_i + \frac{1}{2}B_k(\{s_i\}_i, \{s_i\}_i)$
  define the minimization step as $\{s_i\}_i = \operatorname*{argmin}_{s_i \in T_{q_i^k}\mathbb{U}, \sum_{i=1}^n |s_i|^2 \leq \Delta^2} m_k(\{s_i\}_i)$
  define the ratio $\rho = \frac{F[\{\exp_{q_i^k}(s_i)\}_i] - F[\{q_i^k\}_i]}{m_k(\{s_i\}_i)}$ between energy and model decrease
  if $\rho < \eta_1$                                                  (unsuccessful step)
    decrease $\Delta$ by factor $\vartheta_1$
  else                                                                  (successful step)
    $q_i^{k+1} = \exp_{q_i^k}(s_i)$ for $i = 1, \dots, n$
    if $\rho \geq \eta_2$                                           (very successful step)
      increase $\Delta$ by factor $\vartheta_2$
    endif
  endif
endfor
```

Here, $DF[\{q_i^k\}_i]\{s_i\}_i = \frac{\mathrm{d}}{\mathrm{d}t}F[\{\exp_{q_i^k} ts_i\}_i]|_{t=0}$ denotes the directional derivative of F along the tangent direction $\{s_i\}_i$, and $B_k : (T_{q_i}\mathbb{U})^n \times (T_{q_i}\mathbb{U})^n \to \mathbb{R}$ is an approximation to the Hessian of $F[\{\exp_{q_i^k}(s_i)\}_i]$ with respect to $\{s_i\}_i$. The update $q_i^{k+1} = \exp_{q_i^k}(s_i)$ ensures that all iterates $\{q_i^k\}_i$ automatically lie on the manifold $\mathbb{U}^n$, greatly simplifying the energy landscape. The minimization of the quadratic model m_k within the trust region is implemented according to [9, Alg. 7.3.4-5].

Interpreting F as a function from $\mathbb{H}^n \equiv \mathbb{R}^{4n}$ to $\mathbb{R}$, the Hessian D^2F can be expressed as an $n \times n$ block matrix with block (i, j) given by $\partial_{q_i}\partial_{q_j}F \in \mathbb{R}^{4\times4}$. This matrix can be decomposed into the sum of a diagonal block matrix B^1 with blocks $B_{ii}^1 = \partial_{q_i}^2 E_{\text{fit}}[\Omega_i, q_i]$ and a sparse block matrix B^2 with blocks defined by

$$v^T B_{ij}^2 w = \mu\beta \begin{cases} \mathscr{H}^2(\Gamma_{ij}) \left[\overrightarrow{(vp_{ij}\bar{q}_j)} \cdot \overrightarrow{(q_i p_{ij}\bar{w})} - \left(\psi_{ij} - \overrightarrow{(q_i p_{ij}\bar{q}_j)} + b_{ij}\right) \cdot \overrightarrow{(vp_{ij}\bar{w})}\right] & \text{if } (i,j) \in \mathscr{I}\,, \\ \mathscr{H}^2(\Gamma_{ij}) \left[\overrightarrow{(wp_{ji}\bar{q}_i)} \cdot \overrightarrow{(q_j p_{ji}\bar{v})} - \left(\psi_{ji} - \overrightarrow{(q_j p_{ji}\bar{q}_i)} + b_{ji}\right) \cdot \overrightarrow{(wp_{ji}\bar{v})}\right] & \text{if } (j,i) \in \mathscr{I}\,, \\ \sum_{(k,i)\in\mathscr{I}} \mathscr{H}^2(\Gamma_{ij})\overrightarrow{(q_k p_{ki}\bar{v})} \cdot \overrightarrow{(q_k p_{ki}\bar{w})} + \sum_{(i,k)\in\mathscr{I}} \mathscr{H}^2(\Gamma_{ij})\overrightarrow{(vp_{ik}\bar{q}_k)} \cdot \overrightarrow{(wp_{ik}\bar{q}_k)} & \text{if } i = j\,. \end{cases}$$

B^2 is readily computed, however, the computation of B^1 poses the following difficulties:

- The Hessian of u has to be computed at all stencil points around each voxel, i. e. at all points $x + x_k$, $k = 1, \dots, K$, with x running over all voxels. The interpolation of u and its derivatives at those points will be the bottleneck of the whole algorithm, hence we would like to avoid the computation of higher order derivatives, even more so since higher order derivatives require smoother and thus more expensive interpolation routines.

- For efficiency reasons we would like to interpolate u trilinearly on each voxel, which only allows for one (weak) first derivative; more generally the image u might contain noise leading to spurious high second derivatives and thus less numerical robustness.

For the above reasons we will instead approximate B_{ii}^1, $i = 1, \dots, n$, solely based on the derivatives $\partial_{q_i}E_{\text{fit}}[\Omega_i, q_i]$, using a BFGS update. Denoting by $B^{1,k}$ the k^{th} approximation of B^1 and choosing $B^{1,0} = I$, the BFGS update in our Riemannian setting is defined by [26, 30]

$$(Tv)^T B_{ii}^{1,k+1} Tw = v^T B_{ii}^{1,k} w - \frac{(v^T B_{ii}^{1,k} s_i^k)(w^T B_{ii}^{1,k} s_i^k)}{(s_i^k)^T B_{ii}^{1,k} s_i^k} + \frac{(v^T y_i^k)(w^T y_i^k)}{(y_i^k)^T s_i^k}$$

with $s_i^k = \log_{q_i^k} q_i^{k+1}$, $y_i^k = \partial_s E_{\text{fit}}[\Omega_i, \exp_{q_i^k} s]|_{s=s_i^k} - \partial_s E_{\text{fit}}[\Omega_i, \exp_{q_i^k} s]|_{s=0}$, and T the parallel transport from the last iterate q_i^k to the current one, q_i^{k+1}. Since each block is four by four (i. e. a sum of at most four rank-one matrices), the BFGS approximation is expected to be very good already after very few iterations (in fact we only need to approximate the Hessian on the three-dimensional tangent space to $\mathbb{U}$ so that only three good rank-one updates are required).

The bilinear operator B_k in the trust region method represents the Hessian of $F[\{\exp_{q_i^k} s_i\}_i]$, which is obtained by projection onto the tangent space $T_{q_1^k}\mathbb{U} \times \ldots \times T_{q_n^k}\mathbb{U}$,

$$B_k = \begin{pmatrix} I - q_1^k \otimes q_1^k & & \\ & \ddots & \\ & & I - q_n^k \otimes q_n^k \end{pmatrix} (B^{1,k} + B^2) \begin{pmatrix} I - q_1^k \otimes q_1^k & & \\ & \ddots & \\ & & I - q_n^k \otimes q_n^k \end{pmatrix} .$$

This does not affect the Hessian sparsity. Since the employed algorithm [9, Alg. 7.3.4-5] for the minimization of m_k performs a Cholesky factorization of the Hessian, we finally add a term $q_i^k \otimes q_i^k$ on the i^{th} diagonal block, which makes the matrix definite also in the direction normal to $\mathbb{U}^n$. This does not affect the result since the normal direction is ignored in the trust region minimization anyway.

Explicit update of constraint variable. The update of ψ_{ij} in the split Augmented Lagrange method represents the minimization of

$$G^k(\psi_{ij}) = \tilde{\sigma}(\psi_{ij}) + \tfrac{\mu}{2} \left| \psi_{ij} - \overrightarrow{(q_i^{k+1} p_{ij} \bar{q}_j^{k+1})} + b_{ij}^k \right|^2$$

with $\tilde{\sigma} = \gamma(2 \arcsin |\cdot|)$. Recall that $\gamma \colon [0,\infty) \to [0,\infty)$ is concave with $\gamma(0) = 0$ and jumps at 0. Thus G^k is not convex, not even continuous. Nevertheless, since G only involves a single variable ψ_{ij}, the minimization can be done globally. Furthermore, the particular form (8) of the interface energy density was chosen in order to obtain a simple update formula for ψ_{ij}. Writing $\psi_{\max} = \sin(\frac{\theta_{\max}}{2})$, we have

$$\tilde{\sigma}(\psi_{ij}) = \gamma_{\max} \cdot \begin{cases} 0 & \text{if } \psi_{ij} = 0\,, \\ 1 - (1 - \gamma_{\min})(|\psi_{ij}|/\psi_{\max} - 1)^2 & \text{if } |\psi_{ij}| < \psi_{\max}\,, \\ 1 & \text{else.} \end{cases}$$

If the penalty parameter μ is chosen large enough to make G^k convex away from 0 (which we shall ensure in the code via $\mu > 2\gamma_{\max}(1 - \gamma_{\min})/\psi_{\max}^2$), the minimizing ψ_{ij} is given by $\psi_{ij} = \psi$ if $G^k(\psi) < G^k(0)$ and $\psi_{ij} = 0$ else, where

$$\psi = w \min \left(1, \frac{1 - \kappa \frac{\psi_{\max}}{|w|}}{1 - \kappa} \right)$$

with the abbreviations $w = \overrightarrow{(q_i^{k+1} p_{ij} \bar{q}_j^{k+1})} - b_{ij}^k$ and $\kappa = \frac{2\gamma_{\max}}{\mu} \frac{1-\gamma_{\min}}{\psi_{\max}^2}$. Note that a different choice of γ could be used as well, only then the global optimum would not be given explicitly but would have to be found by e. g. a Newton iteration.

Decreasing computation time by infrequent fitting energy evaluations. The split Augmented Lagrange method is a very easy technique to deal with smooth and non-smooth energy components, but it only yields (slow) linear convergence. Unfortunately, though, it has to be iterated until convergence. Indeed, after the update of all orientations we will have to determine whether any two adjacent grains should be merged into one (see Section 3.3), which is only the case if the two grains have the same orientation (up to a small numerical threshold difference). Now the interface energy with its special feature of zero energy for zero orientation mismatch is only involved in the update of the auxiliary constraint variables ψ_{ij}. Only if the constraint is sufficiently satisfied this shows an effect on the grain orientations, thus requiring many split Augmented Lagrange iterations and involving many evaluations of the costly fidelity term and its derivative.

To reduce the associated computation time we exploit that orientations do not drastically change during these iterations. Hence, after a few of the previously described BFGS updates the fitting energy $E_{\text{fit}}[\Omega_i, q_i]$ can be replaced by the obtained quadratic approximation. Each evaluation of this quadratic model now has a computational complexity of the order of the number of grains instead of the number of pixels, which reduces the computational effort by several orders of magnitude. Whenever the split Augmented Lagrange method converges (with respect to the above introduced stopping criterion), the quadratic model is updated via a few additional split Augmented Lagrange iterations using the true fitting energy and thus producing a new quadratic BFGS approximation. This update of the quadratic model initially leads to a stronger constraint violation so that the split Augmented Lagrange method needs several further iterations to converge. This procedure is repeated until the stopping criterion of the Augmented Lagrange method stays satisfied even after the update.

A simplified version for 2D. The segmentation of 2D images is a special case of the 3D version. Here, all orientations lie in a one-dimensional subgroup of $\mathbb{U}$, the set of unit complex numbers, which in the above notation can be expressed as $q = (\cos\alpha, (\sin\alpha, 0, 0))$ for an orientation angle α. In essence, we simply apply the above-described procedure restricting to unit complex numbers, however, by expressing all energies and iterations in terms of the orientation angles α_i instead of the corresponding complex number, many steps simplify and become more efficient. In particular, in 2D the objective of the Augmented Lagrange method turns into

$$E[\{\alpha_i\}_{i=1,\dots,n}, \{\psi_{ij}\}_{(i,j)\in\mathscr{I}}] = \sum_{i=1}^{n} E_{\text{fit}}[\Omega_i, \alpha_i] + \sum_{(i,j)\in\mathscr{I}} \hat{\sigma}(\psi_{ij})\mathscr{H}^2(\Gamma_{ij}) \quad \text{with } \hat{\sigma} = \gamma(|\cdot|)$$

and with the constraints

$$\psi_{ij} - (\alpha_i - \alpha_j + \alpha_{p_{ij}}) = 0$$

for $(i,j) \in \mathscr{I}$, where $\alpha_{p_{ij}}$ represents a rotation angle from the point group. To obtain a simple update formula for the constraint variables ψ_{ij} we again choose

$$\hat{\sigma}(\psi_{ij}) = \gamma_{\max} \cdot \begin{cases} 0 & \text{if } \psi_{ij} = 0\,, \\ 1 - (1 - \gamma_{\min})(|\psi_{ij}|/\psi_{\max} - 1)^2 & \text{if } |\psi_{ij}| < \psi_{\max}\,, \\ 1 & \text{else.} \end{cases}$$

Also the joint Newton- and quasi-Newton method can now be performed in Euclidean instead of curved Riemannian space.

3.3 Initialization, grain merging, and termination

Our strategy to combine the interface and the orientation update can be summarized in the following algorithmic steps.

1. **Initialization with a large number of small grains.** Initially we partition the domain into little cubes, each of which represents a segment Ω_i, and we assign them with random orientations (see ① in Fig. 2). This procedure has several advantages: Tiling the domain into small cubes has the effect that all interfaces already lie close to their final position so that only little interface motion has to occur in order to capture the correct grain interfaces. Furthermore, by initializing the orientations randomly it is likely that during the following orientation update at least one little cube per grain arrives at the correct globally minimizing orientation. During the subsequent interface motion, these cubes will grow at the expense of those cubes with less optimal orientations, which will finally shrink away.

2. **Initial orientation update neglecting interface energies.** The algorithm is started by updating all orientations q_i, however, for the interface energy weight β set to zero (② in Fig. 2). In that case the split Augmented Lagrange method just reduces to the combined Riemannian Newton and quasi-Newton method for the fitting energy. The reason for setting $\beta = 0$ is that many interfaces are not yet correctly positioned so that a large number of the Ω_i intersect with two or more grains. Those cubes will be rather indifferent to different orientations and would for $\beta > 0$ thus simply acquire the orientation of one of the neighboring grains, which is often too premature.

3. **Alternating interface flow, orientation update, and grain merges.** The core of the algorithm consists of repeating the following three steps.

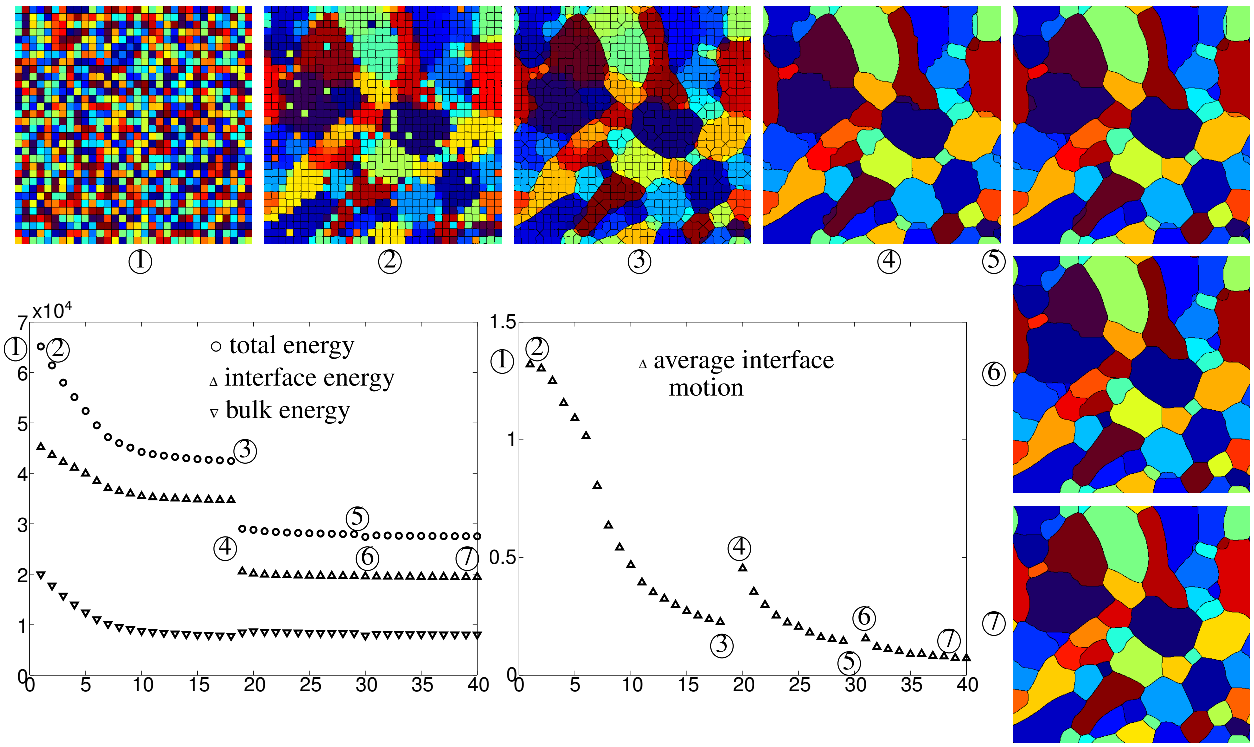

Figure 2: Illustration of the algorithmic steps as described in the text: Initialization ①, initial orientation update ②, first gradient flow for the interfaces ③, first orientation update with grain merge ④, further alternating interface and orientation updates with grain merges ⑤-⑦. Colors indicate grain orientation.

(a) L^2 **gradient flow for interfaces.** Perform the algorithm described in Section 3.1 until the energy decrease during the past 10 steps falls below a threshold value (in our experiments chosen as 10 % of the absolute energy). The first such interface update is visualized in Figure 2 as step ③.

(b) **Orientation update.** Perform the algorithm described in Section 3.2

(c) **Merging adjacent grains with matching orientations.** The interface between any two adjacent grains Ω_i, Ω_j with equal orientations is associated with zero interface energy, since $\Omega_i \cup \Omega_j$ may be interpreted as a single grain. Hence, after updating the orientations we merge all adjacent grains with misorientation angle below a numerical threshold (10^{-2} in our code), which speeds up the subsequent gradient flow for the partition $\{\Omega_i\}_i$ as well as the next update of the corresponding orientations $\{R_i\}_i$. The new grain is assigned the orientation of one of its parents. Note that in case of multiple connected grains whose pairwise misorientation angle lies below the threshold, all these grains will be merged into one. The first grain merge after an orientation update is visualized in Figure 2 as step ④.

4. **Stopping criterion.** Stop if the energy decrease during the past 10 steps falls below a threshold value (in our experiments chosen as 1 % of the absolute energy).

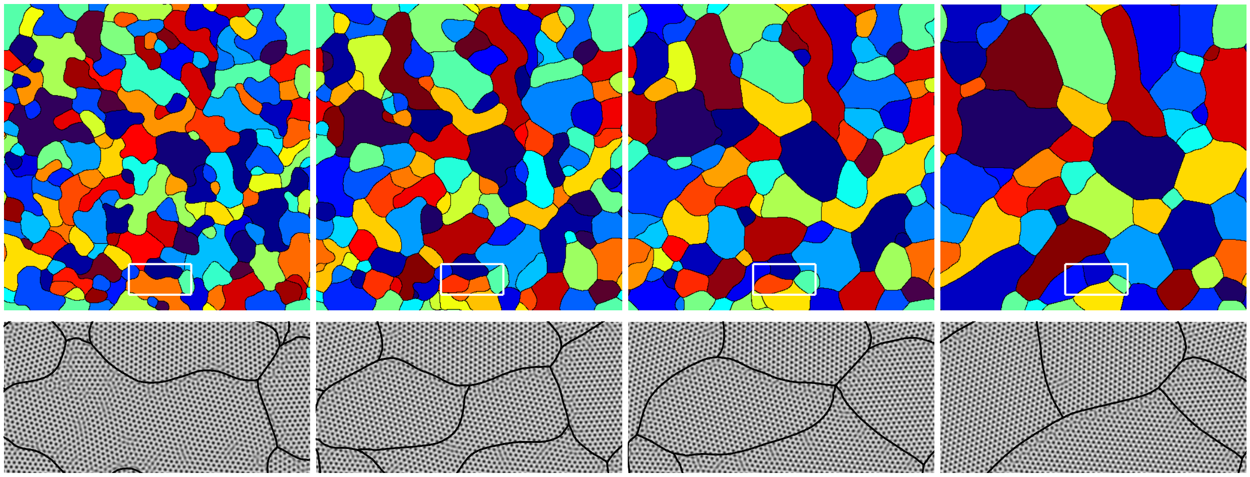

Figure 3: Steps 50, 200, 800, and 2500 of a phase field crystal coarsening simulation segmented by the proposed method (the bottom shows a zoom into the indicated region; colors indicate grain orientation).

4 Results

Figures 3 to 7 show the performance of the algorithm on test experiments. Figure 3 depicts several time snap shots of a 2D phase field crystal (PFC) simulation, which is a model for simulating the time evolution of atomic crystals [14]. A zoom into the actual phase field crystal is shown at the bottom, while the large images show the result of our automatic grain segmentation method. The grain segmentation clearly shows a coarsening of the polycrystalline structure over time: the average grain size increases, while small grains shrink until they vanish. Note that during the initial stage of coarsening, small grains seem to suddenly appear in a few places (the zoom shows one such instance). This is due to the conceptual difficulty of defining a grain in a crystalline material with many defects: a single dislocation (a point defect of the crystal) can sometimes either be interpreted as an isolated defect within a grain or as part of a low-angle grain boundary which traverses the region previously interpreted as a single grain. Our method chooses one or the other interpretation solely based on the energy, and the energetic character of the defect may vary during the crystal evolution.

Figures 4 and 5 show segmentation results for artificial, periodic 2D PFC polycrystals, a polycrystal of hexagonal grains (the original PFC image is not shown) as well as an array of circular grains embedded in a matrix of constant orientation. The segmentation of the hexagonally tiled polycrystal is once performed for a constant interfacial energy $\gamma = \gamma_{\max}$ and once for the above-proposed misorientation-dependent interfacial energy. Note that in the former case more grains are merged and that triple junctions all have 180 degree angles, wich is not true for the misorientation-dependent interfacial energy.

The results are harder to visualize and comprehend in 3D, for which reason we choose a 128x128x128 input image of an artificial, periodic 3D polycrystal, which is composed of cubic grains (Fig. 6 shows the input data as well as the segmentation result). For a slightly larger simulation, eight copies of this polycrystal are stacked together into a 256x256x256 input image. Again, the method correctly identifies the cubic grains, random subsets of which are depicted in Fig. 7.

Table 1 shows the distribution of computation time over the different components of the algorithm for a small 3D example. Note that the employed techniques to avoid the very expensive fidelity term evaluations indeed lead to reducing the time spent inside the fidelity term evaluation to times comparable to the convolution-redistancing algorithm for the interface motion. The expensive components of the algorithm roughly scale linearly with the number of voxels so that the segmentation of a 256x256x256 polycrystal image without parallelization roughly takes 16 h. However, parallelization leads to a decrease in computing time approximately linear in the number of CPUs. Note that 2D simulations are more efficient due to the simpler representation of crystal orientations and the smaller stencil size. Indeed, the non-parallelized segmentation of a 1024x1024 crystal image lies at the order of five to ten minutes.

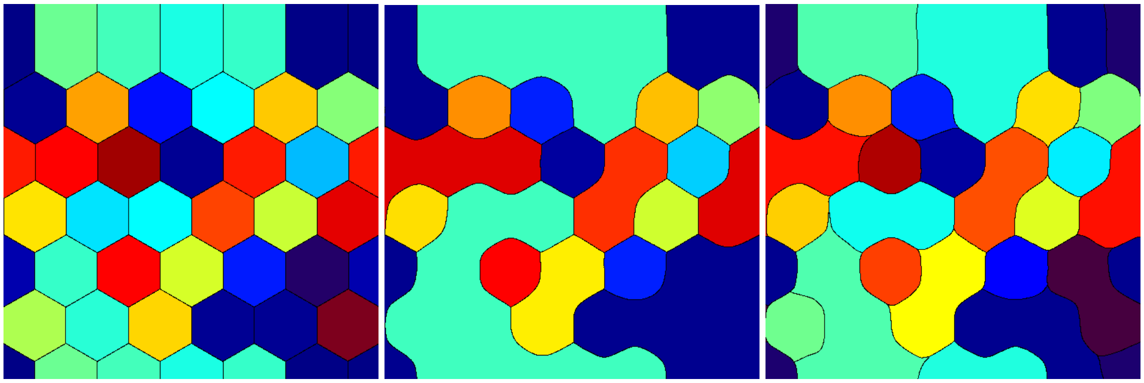

Figure 4: Left: Orientation angles in a synthetic polycrystal. Center: Segmentation result with uniform interfacial energy. Right: Segmentation result with misorientation-dependent interfacial energy. Colors indicate grain orientation.

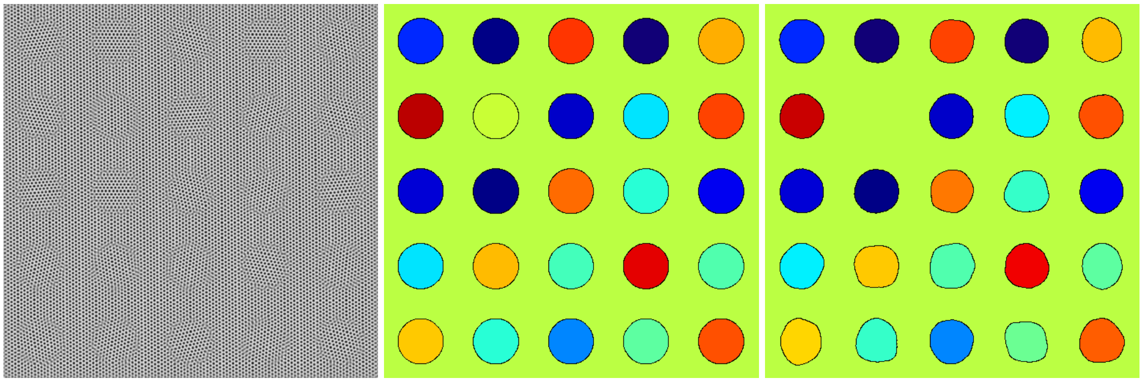

Figure 5: Left: PFC image; middle: orientation angles in synthetic polycrystal; right: segmentation result with inhomogenous interface energies.

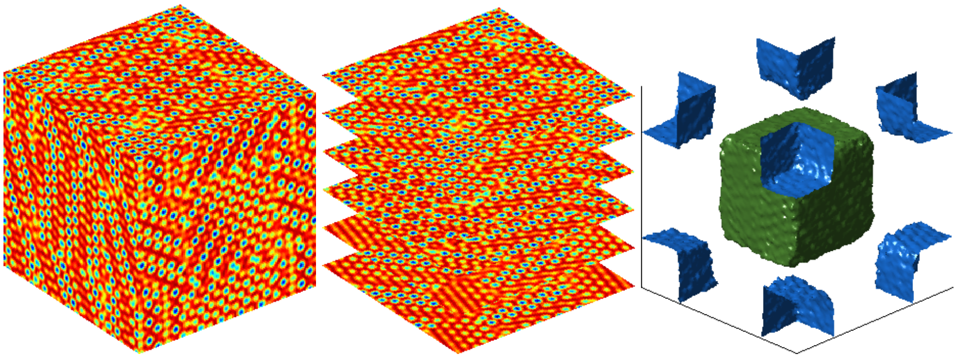

Figure 6: Input data (left, middle) and selected grains from the segmentation result (right) for an artificial polycrystal composed of cubic grains.

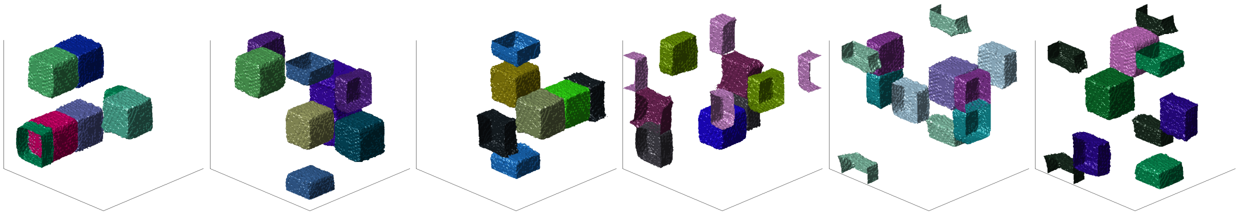

Figure 7: Randomly chosen grains from the segmentation result for a 2x2x2 stack of the periodic block from Fig. 6.

gradient flow for interfaces		**59%**
convolution	1%	
fidelity density evaluation	18%	
recombination	7%	
redistancing	14%	
grain tracking and swapping	18%	
initial update of orientations		**19%**
fidelity term evaluation	18%	
further orientation updates		**20%**
fidelity term evaluation	17%	
solution of trust-region subproblem	0%	
grain merging	2%	
energy evaluation for stopping criterion & output		**2%**

Table 1: Fraction of time spent in each part of the algorithm on a serial processor for the segmentation of a 256x256x4 3D crystal image. Total computation time was 15 min 9 s.

Acknowledgements

The authors thank Robert V. Kohn for many valuable discussions on the subject. M.E. gratefully acknowledges the support of National Science Foundation grant OISE–0967140.

References

[1] Ambrosio, L., Fusco, N., and Pallara, D. *Functions of bounded variation and free discontinuity problems*. Oxford Mathematical Monographs. Oxford University Press, New York, 2000.

[2] Bae, E., Yuan, J., and Tai, X.-C. Global minimization for continuous multiphase partitioning problems using a dual approach. *Int. J. Comput. Vis. 92*, 1 (2010), 112–129.

[3] Berkels, B., Rätz, A., Rumpf, M., and Voigt, A. Extracting grain boundaries and macroscopic deformations from images on atomic scale. *J. Sci. Comput. 35*, 1 (2008), 1–23.

[4] Boerdgen, M., Berkels, B., Rumpf, M., and Cremers, D. Convex relaxation for grain segmentation at atomic scale. In *VMV 2010 - Vision, Modeling & Visualization* (2010), D. Fellner, Ed., Eurographics Association, pp. 179–186.

[5] Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.-P., and Osher, S. Fast global minimization of the active contour/snake model. *J. Math Imaging Vis. 28* (2007), 151–167.

[6] Chan, T. F., Esedoglu, S., and Nikolova, M. Finding the global minimum for binary image restoration. In *Proceedings of the International Conference on Image Processing* (2005), vol. 1, pp. 121–124.

[7] CHAN, T. F., SANDBERG, B. Y., AND VESE, L. A. Active contours without edges for vector-valued images. *J. Vis. Commun. Image Represent. 11*, 2 (2000), 130–141.

[8] CHAN, T. F., AND VESE, L. A. Active contours without edges. *IEEE Trans. Image Process. 10*, 2 (2001), 266–277.

[9] CONN, A. R., GOULD, N. I. M., AND TOINT, P. L. *Trust-Region Methods.* SIAM, 2000.

[10] DELONG, A., AND BOYKOV, Y. Globally optimal segmentation of multi-region objects. In *Proceedings of the International Conference on Computer Vision* (2009), pp. 285–292.

[11] EL-ZEHIRY, N., SAHOO, P., XU, S., AND ELMAGHRABY, A. Graph cut optimization for the Mumford–Shah model. In *Proceedings of the International Conference on Visualization, Imaging and Image Processing (IASTED)* (2007), pp. 182–187.

[12] EL-ZEHIRY, N. Y., AND ELMAGHRABY, A. A graph cut based active contour for multiphase image segmentation. In *Proceedings of the International Conference on Image Processing (ICIP)* (2008), pp. 3188–3191.

[13] EL-ZEHIRY, N. Y., AND GRADY, L. Combinatorial optimization of the discretized multiphase Mumford–Shad functional. *Int. J. Comput. Vis. 104* (2013), 270–285.

[14] ELDER, K. R., AND GRANT, M. Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals. *Phys. Rev. E 70* (2004), 051605.

[15] ELSEY, M., AND ESEDOGLU, S. Fast and accurate redistancing by directional optimization. *SIAM J. Sci. Comput. 36*, 1 (2014), A219–A231.

[16] ELSEY, M., ESEDOḠLU, S., AND SMEREKA, P. Diffusion generated motion for grain growth in two and three dimensions. *J. Comp. Phys. 228*, 21 (2009), 8015–8033.

[17] ELSEY, M., ESEDOḠLU, S., AND SMEREKA, P. Simulations of anisotropic grain growth: Efficient algorithms and misorientation distributions. *Acta Mater. 61* (2013), 2033–2043.

[18] ELSEY, M., AND WIRTH, B. Fast automated detection of crystal distortion and crystal defects in polycrystal images. *SIAM Multiscale Model. Simul. 12*, 1 (2014), 1–24.

[19] ESEDOḠLU, S., AND OTTO, F. Threshold dynamics for networks with arbitrary surface tensions. *Commun. Pure Appl. Math.* (2014).

[20] ESEDOḠLU, S., RUUTH, S., AND TSAI, R. Diffusion generated motion using signed distance functions. *J. Comp. Phys. 229*, 4 (2010), 1017–1042.

[21] GOLDSTEIN, T., AND OSHER, S. The split Bregman method for L1–regularized problems. *SIAM J. Imag. Sci. 2*, 2 (2009), 323–343.

[22] JEON, M., ALEXANDER, M., PEDRYCZ, W., AND PIZZI, N. Unsupervised hierarchical image segmentation with level set and additive operator splitting. *Pattern Recognit. Lett. 26* (2005), 1461–1469.

[23] MERRIMAN, B., BENCE, J., AND OSHER, S. Diffusion generated motion by mean curvature. In *Computational Crystal Growers Workshop* (1992), J. E. Taylor, Ed., American Mathematical Society, American Mathematical Society, pp. 73–83.

[24] MERRIMAN, B., BENCE, J. K., AND OSHER, S. Motion of multiple junctions: A level set approach. *J. Comput. Phys. 112*, 2 (1994), 334–363.

[25] MUMFORD, D., AND SHAH, J. Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math. 42*, 5 (1989), 577–685.

[26] Nocedal, J., and Wright, S. J. *Numerical optimization*, second ed. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006.

[27] Osher, S., Burger, M., Goldfarb, D., Xu, J., and Yin, W. An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul. 4*, 2 (2005), 460–489.

[28] Patala, S., Mason, J. K., and Schuh, C. A. Improved representations of misorientation information for grain boundary science and engineering. *Progress in Materials Science 57* (2012), 1383–1425.

[29] Read, W. T., and Shockley, W. Dislocation models of crystal grain boundaries. *Physical Review 78*, 3 (1950), 275–289.

[30] Ring, W., and Wirth, B. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization 22*, 2 (2012), 596–627.

[31] Ruuth, S. J. A diffusion-generated approach to multiphase motion. *J. Comput. Phys. 145* (1998), 166–192.

[32] Sethian, J. A. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci. 93* (1996), 1591–1595.

[33] Tsitsiklis, J. N. Efficient algorithms for globally optimal trajectories. *IEEE Trans. Autom. Control 40*, 9 (1995), 1528–1538.

[34] Vese, L. A., and Chan, T. F. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vis. 50*, 3 (2002), 271–293.