Reconstruction of grains in polycrystalline materials from incomplete data using Laguerre tessellations

Lukas Petrich[1], Jakub Staněk[2], Mingyan Wang[3], Daniel Westhoff[1], Luděk Heller[4], Petr Šittner[4], Carl E. Krill III[3], Viktor Beneš[5] and Volker Schmidt[1]

Far-field three-dimensional X-ray diffraction (3DXRD) microscopy allows for quick measurement of the centers of mass and volumes of a large number of grains in a polycrystalline material, along with their crystal lattice orientations and internal stresses. However, the grain boundaries—and, there-

[1]Institute of Stochastics, Faculty of Mathematics and Economics, Ulm University, 89069 Ulm, Germany
[2]Department of Mathematics Education, Faculty of Mathematics and Physics, Charles University, 18675 Prague, Czech Republic
[3]Institute of Functional Nanosystems, Faculty of Engineering, Computer Science and Psychology, Ulm University, 89081 Ulm, Germany
[4]Institute of Physics, Academy of Sciences of Czech Republic, 18221 Prague, Czech Republic
[5]Department of Probability and Mathematical Statistics, Faculty of Mathematics and Physics, Charles University, 18675 Prague, Czech Republic

fore, individual grain shapes—are not observed directly. The present paper aims to overcome this shortcoming by reconstructing grain shapes based only on the incomplete morphological data described above. To this end, cross-entropy optimization is employed to find a Laguerre tessellation that minimizes the discrepancy between its centers of mass and cell sizes and those of the measured grain data. The proposed algorithm is highly parallel and is thus capable of handling many grains ($> 8\,000$). The validity and stability of the cross-entropy approach are verified on simulated and experimental datasets.

Keywords: 3DXRD microscopy; polycrystalline material; cross-entropy method; Laguerre tessellation; incomplete data

Brief title: Reconstruction of grain boundaries from incomplete data

Corresponding author: Lukas Petrich (Address: Helmholtzstraße 18, D-89069 Ulm; Phone: +49 731/5023590; Fax: +49 731/5023649; Email: lukas.petrich@uni-ulm.de)

1 Introduction

Recently, novel diffraction-based imaging techniques capable of measuring the centers of mass, volumes and crystal lattice orientations of thousands of grains in a polycrystal have became available, see Abdolvand et al. (2015); Johnson et al. (2008); Ludwig et al. (2009); Poulsen (2004); Sedmák et al. (2016). One of the most promising methods yielding large datasets ($> 10\,000$ grains) is 3D X-ray diffraction (3DXRD) microscopy (Poulsen, 2004). In the far-field setup (Abdolvand et al., 2015; Sedmák et al., 2016), the sample rotates with respect to the X-ray beam and only the diffraction signal from the grains is collected. This leads to an incomplete dataset, since there is no information on the location and shape of the grain boundaries. However, for many applications it is crucial to have complete knowledge of the microstructure. A prominent example for such a situation is the analysis of the mechanical behavior of a polycrystal of defined microstructure (Materials by Design) by means of the finite-element (FE) method. Since the 3DXRD technique provides additional information on the elastic strain and stress tensors for each grain in samples subjected to external forces, it is possible to compare this experimental information to a comprehensive FE analysis of the mechanical behavior of the material, see Abdolvand et al. (2015); Sedmák et al. (2016).

One possibility to arrive at the desired complete model of microstructure is to add a near-field detector to the 3DXRD setup or to employ other imaging techniques, such as diffraction contrast tomography (Johnson et al., 2008). This is an excellent strategy if smaller datasets (< 1000 grains) are considered, but it can be impractical when large datasets are required for meaningful simulations of mechanical properties of engineering materials.

Alternatively, the missing information can be added by utilizing the available far-field diffraction data and reconstructing the grain boundaries on the computer, which yields an approximation of the grain shapes, e.g., based on Laguerre tessellations (Lautensack

and Zuyev, 2008). This has been demonstrated by Lyckegaard et al. (2011), where a fast heuristic is described that allows for quick reconstruction of grain boundaries from 3DXRD data. In Quey and Renversade (2018) the task of reconstructing the grain boundary network based solely on the centers of mass and volumes from a 3DXRD measurement was formulated as an optimization problem, which led to much better results at the cost of longer runtimes. Both methods are discussed in-depth later on. The present paper takes a similar approach, but the optimization is performed with an algorithm that is more robust against local minima. Moreover, instead of following a mostly sequential procedure, the algorithm is highly parallel and can be run on a distributed system, making it applicable to datasets with a large number of grains. In a forthcoming paper we will exploit this property even further by considering more general tessellation models having curved cell boundaries, such as the spherical growth model, (see, e.g., Šedivý et al., 2018), which are better suited than flat-faced Laguerre tessellations to the modeling of microstructures manifesting non-negligible boundary curvatures. For these general tessellation models, no efficient formulas for computing the centers of mass and volumes of the tessellation cells are available, and the resulting high computational effort can be tackled only by distributing the workload over many CPU cores. To our knowledge, there are no methods proposed in the literature capable of solving this problem. However, the intention of the present paper is to establish our reconstruction method for fitting tessellations in the well-studied Laguerre case and compare its results to those of other methods from literature.

The remainder of this paper is structured as follows: We begin in Section 2 with a recap of the fundamentals of Laguerre tessellations and an introduction to our proposed reconstruction procedure. Then, to prove the validity of our approach, we apply the method to four different datasets in Section 3 and discuss the results in Section 4, where we also compare our method with two other techniques from the literature and suggest

possible extensions to our approach. This is followed by concluding statements.

2 Models and Methods

In this section we recall some fundamentals of Laguerre tessellations and present our reconstruction method.

2.1 Laguerre Tessellations

A tessellation in $\mathbb{R}^3$ is a countable collection of sets (called cells), $\mathcal{T} = \{C_i \subset \mathbb{R}^3 : i \in \mathbb{N}\}$, such that

1) $\text{int}(C_i) \cap \text{int}(C_j) = \emptyset \quad \text{for } i \neq j$,

2) $\bigcup_{i\in\mathbb{N}} C_i = \mathbb{R}^3$,

3) $\mathcal{T}$ is locally finite (i.e. $\#\{C_i \in \mathcal{T} : C_i \cap B \neq \emptyset\} < \infty$ for all bounded $B \subset \mathbb{R}^3$),

4) C_i is a compact set for all $i \in \mathbb{N}$.

In this paper we consider only tessellations in a bounded subset $W \subset \mathbb{R}^3$, which are defined analogously. Note that condition 4) implies that the cells of a tessellation are bounded polyhedra if we additionally assume that they are convex sets. For various classes of tessellations compare, e.g., Šedivý et al. (2018).

A Laguerre tessellation, also called a Laguerre diagram or power diagram, is a generalization of the well-known Voronoi tessellation (Møller, 1994). Let $x, y \in \mathbb{R}^3$ and $r \geq 0$; then the power distance (Laguerre distance measure) is defined by

$$\text{pow}(y, (x, r)) = ||y - x||^2 - r^2,$$

where $||\bullet||$ is the Euclidean norm on $\mathbb{R}^3$. A Laguerre tessellation is given by a set of weighted generating points $\{(x_i, r_i)\}_{i\in\mathcal{I}}$ with seed point $x_i \in \mathbb{R}^3$ and weight $r_i \geq 0$ for $i \in \mathcal{I} \subset \mathbb{N}$. The i-th cell C_i of a Laguerre tessellation is then defined as

$$C_i = \{y \in \mathbb{R}^3 : \mathrm{pow}(y, (x_i, r_i)) \leq \mathrm{pow}(y, (x_j, r_j)) \text{ for each } j \in \mathcal{I}\}.$$

Note that in contrast to Voronoi tessellations, for which all weights r_i are equal to each other, the Laguerre cell C_i defined above may be empty (i.e., the cell may have no interior points) and, even if $C_i \neq \emptyset$, it is not guaranteed that its seed point x_i lies inside C_i. Moreover, while for a given Voronoi tessellation its generating points are uniquely determined, a Laguerre tessellation can be generated by uncountably many sets of generating points, see Duan et al. (2014); Lautensack and Zuyev (2008). On the other hand, Laguerre tessellations provide much more flexibility compared to Voronoi tessellations. In fact, every tessellation in $\mathbb{R}^3$ (as defined above) that is normal—i.e., each k-dimensional face lies at the intersection of exactly $4-k$ cells for $k = 0, \ldots, 3$—is a Laguerre tessellation, see Lautensack and Zuyev (2008).

Recall that the aim of the present paper is to find a Laguerre tessellation whose cells match given volumes and centers of mass (also called centroids) and thus to reconstruct the grain boundaries of a material sample captured by a 3DXRD measurement. Of course, we cannot in general assume that the grain boundaries of the measured polycrystalline material follow a Laguerre tessellation exactly; therefore, we seek a Laguerre tessellation that best approximates the given experimental data. For a set of volumes and centers of mass, it is not clear whether there is only one such "optimal" Laguerre tessellation. We do know, on the other hand, that the Laguerre tessellation generators are not uniquely determined (consider, e.g., adding a constant value to all weights). We show below, however, that these issues are of no consequence for the reconstruction method developed in the present paper.

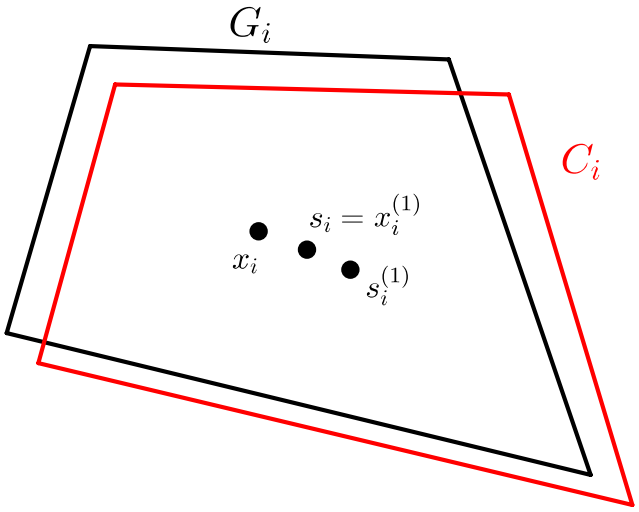


Figure 1: Visualization of the two-step reconstruction method. The Laguerre cell G_i has the correct center of mass s_i, but its seed point x_i is unknown and we want to approximate it. The Laguerre cell C_i is given by the seed $x_i^{(1)} = s_i$, and the weight $r_i^{(1)}$ is the same for both cells G_i and C_i.

2.2 Two-Step Reconstruction Method

A very fast, heuristic method for finding a Laguerre tessellation whose cells approximately match given volumes and centers of mass is presented in Lyckegaard et al. (2011). Since the method is comprised of two steps, we refer to it in the following as the two-step reconstruction method, and we employ it to come up with a Laguerre tessellation that acts as a starting point for our approach.

Let $\{(s_i, v_i)\}_{i\in\mathcal{I}}$ be the input dataset that we want to match where s_i is the center of mass and v_i is the volume of the i-th grain. In the first step, we create the tessellation $\mathcal{T}_1 = \{C_i\}_{i\in\mathcal{I}}$ with generators $\{(x_i^{(1)} = s_i, r_i^{(1)} = \left(\frac{3v_i}{4\pi}\right)^{\frac{1}{3}})\}_{i\in\mathcal{I}}$. We use s_i as the seed point of the i-th Laguerre cell because, even if the center of mass does not coincide with the position of the seed $x_i^{(1)}$, we expect them to be close to each other. The choice of weights is motivated by visualizing the generators as a system of spheres with centers $x_i^{(1)}$ and radii $r_i^{(1)}$. If the spheres are non-overlapping and densely packed, the corresponding

Laguerre cells will have similar shapes and in particular similar volumes. The notion of nearly spherical cells with volumes $v_i = \frac{4}{3}\pi {r_i^{(1)}}^3$ leads to the formula for the weights, $r_i^{(1)} = \left(\frac{3v_i}{4\pi}\right)^{\frac{1}{3}}$.

In the second step, we assume that the weights of $\mathcal{T}_1 = \{C_i\}_{i\in\mathcal{I}}$ are already good enough and we only want to improve the seed points. For that we consider a Laguerre tessellation $\{G_i\}_{i\in\mathcal{I}}$ with the same weights $\{r_i = r_i^{(1)}\}_{i\in\mathcal{I}}$ as $\mathcal{T}_1$, where each cell has the correct center of mass s_i but whose seed points $\{x_i\}_{i\in\mathcal{I}}$ are unknown. Denote by $s_i^{(1)}$ the center of mass of the cell C_i. If the centroid (and seed of $\mathcal{T}_1$) s_i and the seed x_i are close to each other, the shapes and sizes of the cells C_i and G_i will be similar, and, hence, the vectors $v = x_i - s_i$ and $v^{(1)} = x_i^{(1)} - s_i^{(1)}$ are similar as well, see Figure 1. By setting $v = v^{(1)}$ we obtain an approximation of $x_i = 2s_i - s_i^{(1)}$, and we can define the final Laguerre tessellation $\mathcal{T}_2$ with the generating points $\{(x_i = 2s_i - s_i^{(1)}, r_i = \left(\frac{3v_i}{4\pi}\right)^{\frac{1}{3}})\}_{i\in\mathcal{I}}$.

2.3 Reconstruction Based on the Cross-Entropy Method

The cross-entropy (CE) method is a versatile stochastic optimization method that can be used to solve many continuous optimization problems, see Kroese et al. (2006); Rubinstein and Kroese (2004). In Spettl et al. (2016) it was successfully applied to the fitting of Laguerre tessellations to 3D image data obtained by X-ray tomography. The idea of the method is that the global minimum $\mathbf{z}^* = \arg\min_{\mathbf{z}\in\mathbb{R}^m} c(\mathbf{z})$ of an m-dimensional cost function $c : \mathbb{R}^m \to \mathbb{R}$ for $m \in \mathbb{N}$ can be described by a degenerate probability distribution where $\mathbf{z}^*$ has probability 1. The CE method generates a sequence of probability distributions that converges to this degenerate probability distribution. If there is more than one global minimum of the cost function c, the CE method finds one of them.

More precisely, the CE method works as follows. Denote by $f(\mathbf{z};\theta)$ the probability density of possible locations of a global minimum $\mathbf{z}^*$, which depends on a certain parameter vector $\theta \in \Theta \subset \mathbb{R}^{\widetilde{m}}$ for $\widetilde{m} \in \mathbb{N}$. Generate a sample $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)}$ of length $n \in \mathbb{N}$

from the probability density $f(\mathbf{z};\theta)$ and compute the corresponding values of the cost function $c(\mathbf{z}^{(1)}),\ldots,c(\mathbf{z}^{(n)})$. The set of $\ell < n$ best-performing members of the sample (i.e., those with the lowest values of the cost function) are called the elite set. The parameter vector θ is updated by computing the maximum likelihood estimate based on this elite set. This procedure is repeated until the distribution corresponding to $f(\mathbf{z};\theta)$ is nearly deterministic.

A common choice for the probability density $f(\mathbf{z},\theta)$ is the product of normal densities, i.e.,

$$f(\mathbf{z};\theta) = \varphi(z_1;\mu_1,\sigma_1)\cdot\varphi(z_2;\mu_2,\sigma_2)\cdot\ \ldots\ \cdot\varphi(z_m;\mu_m,\sigma_m),$$

where $\mathbf{z}=(z_1,\ldots,z_m)$, $\theta=(\mu_1,\sigma_1,\ldots,\mu_m,\sigma_m)$ for some $\mu_1,\ldots,\mu_m\in\mathbb{R}$, $\sigma_1,\ldots,\sigma_m>0$, and $\varphi(z;\mu,\sigma)=(2\pi\sigma^2)^{-0.5}\exp\left(-(z-\mu)^2/(2\sigma^2)\right)$ is the density of the normal distribution with mean μ and standard deviation σ. With this choice of the density $f(\mathbf{z};\theta)$, we can easily update the parameters $\mu_1,\ldots,\mu_m$ with the component-wise sample means of the elite set and $\sigma_1,\ldots,\sigma_m$ with the component-wise sample standard deviation of the elite set.

The algorithm of the CE method is thus given as follows:

1. **Initialization.** Choose an initial parameter $\theta^{(0)}=(\mu_1^{(0)},\sigma_1^{(0)},\ldots,\mu_m^{(0)},\sigma_m^{(0)})$ and set $k=0$.

2. **Sampling.** Generate a sample $\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(n)}$ from $f(\mathbf{z};\theta^{(k)})$ and select the elite set $\mathbf{z}^{(e_1)},\ldots,\mathbf{z}^{(e_\ell)}$ from this sample with respect to the cost function c.

3. **Updating.** Calculate the component-wise sample means and standard deviations of the elite set; i.e., calculate

$$\bar{z}_1=\frac{1}{\ell}\sum_{i=1}^{\ell}z_1^{(e_i)},\ldots,\bar{z}_m=\frac{1}{\ell}\sum_{i=1}^{\ell}z_m^{(e_i)}$$

and

$$\hat{z}_1 = \sqrt{\frac{1}{\ell - 1} \sum_{i=1}^{\ell} \left(\bar{z}_1 - z_1^{(e_i)} \right)^2}, \ldots, \hat{z}_m = \sqrt{\frac{1}{\ell - 1} \sum_{i=1}^{\ell} \left(\bar{z}_m - z_m^{(e_i)} \right)^2},$$

where $z_j^{(e_i)}$ is the j-th component of $\mathbf{z}^{(e_i)}$. Set the new parameter

$$\theta^{(k+1)} = \left(\mu_1^{(k+1)} = \bar{z}_1, \sigma_1^{(k+1)} = \hat{z}_1 \, , \ldots, \mu_m^{(k+1)} = \bar{z}_m, \sigma_m^{(k+1)} = \hat{z}_m \right).$$

4. **Iteration.** The algorithm is stopped if the cost function does not decrease significantly for a certain number of steps. Otherwise, set $k = k + 1$ and repeat from Step 2.

Note that in our case $(\mu_1, \ldots, \mu_m) = (x_{1,1}, x_{1,2}, x_{1,3}, r_1, \ldots, x_{N,1}, x_{N,2}, x_{N,3}, r_N)$, where $\{(x_i, r_i)\}_{i \in \mathcal{I}}$ with $\#\mathcal{I} = N$ and $x_i = (x_{i,1}, x_{i,2}, x_{i,3})$ for all $i \in \mathcal{I}$, are generating points that converge during the CE procedure to the generating points of a Laguerre tessellation minimizing the cost function. For the number of grains N in the dataset that we want to reconstruct, it thus holds that $m = 4N$. Conversely, the generating point of the i-th cell is given by $((\mu_{4i-3}, \mu_{4i-2}, \mu_{4i-1}), \mu_{4i}) = (x_i, r_i)$. Each element of the sample $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ hence corresponds to all generating points of a Laguerre tessellation used for probing the cost function. Because the weights of the Laguerre tessellation must be non-negative, we employ truncated normal distributions for the weights, in which negative values are assigned a probability of zero.

Since each element of the sequence $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)}$ can be generated independently, it is possible to split the sampling step of each iteration into n tasks consisting of drawing a generator vector $\mathbf{z}^{(i)}$ from $f(\mathbf{z}; \theta)$, constructing the Laguerre tessellation, and computing the cost function. These tasks can be executed on multiple CPU cores, possibly distributed over different machines. This approach has been studied in Evans et al. (2007).

Furthermore, the updating step of the CE algorithm can also be sped up by sorting the generator vectors with respect to their cost, selecting the best ℓ generator vectors $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\ell)}$, and computing the component-wise mean and standard deviation in a parallel fashion, e.g., based on the MapReduce programming model (Dean and Ghemawat, 2008). In summary, it is thus possible to utilize a high number of CPU cores and run the algorithm in a cluster setting, resulting in a significant speedup, see Evans et al. (2007).

2.3.1 Cost Function

Recall that our aim is to find a Laguerre tessellation in order to approximate the grain boundaries of a polycrystalline material for which we know only the centers of mass and volumes of the grains. For this purpose, we use the following notation:

- $D = \{(s_i, v_i)\}_{i \in \mathcal{I}}$ is the input dataset, with $s_i = (s_{i,1}, s_{i,2}, s_{i,3}) \in \mathbb{R}^3$ denoting the center of mass and $v_i > 0$ the volume of the i-th grain,
- $G_{\mathcal{T}} = \{(x_i, r_i)\}_{i \in \mathcal{I}}$ is the sequence of seed points $x_i = (x_{i,1}, x_{i,2}, x_{i,3}) \in \mathbb{R}^3$ and weights $r_i \geq 0$ of the Laguerre tessellation $\mathcal{T}$ that approximates the data, the i-th cell of which has center of mass $\hat{s}_i \in \mathbb{R}^3$ and volume $\hat{v}_i \geq 0$.

Note that we consider a Laguerre tessellation $\mathcal{T}$ restricted to a bounded sampling window $W \subset \mathbb{R}^3$, and the cells at the boundary of W—which might be cut and are thus no longer Laguerre cells—are treated in the same way as cells in the interior. The reason for this is that it is practically impossible to identify the boundary cells of the input dataset without knowing the exact shape of the grains. Any edge treatment based on reconstructed tessellations during the CE method yields a different set of boundary cells for different tessellations, and this spoils the comparability of the cost function, which is necessary for identification of the elite set. Furthermore, the quality of the reconstruction would decrease at the boundary.

As we want to employ the CE method, we need to define a cost function c that measures the distance between the input dataset D and the set of generating points of the Laguerre tessellation $G_{\mathcal{T}}$. A possible choice for the cost function is

$$c_D(G_{\mathcal{T}}) = c_D\left(x_{1,1}, x_{1,2}, x_{1,3}, r_1, \ldots, x_{N,1}, x_{N,2}, x_{N,3}, r_N\right) = \frac{1}{N}\sum_{i\in\mathcal{I}} c_{D,loc}\left(\hat{s}_i, \hat{v}_i\right)$$

with

$$c_{D,loc}\left(\hat{s}_i, \hat{v}_i\right) = \frac{3}{4}\min\left(\frac{|s_i - \hat{s}_i|}{r_i'}, 1\right) + \frac{1}{4}\min\left(\frac{|v_i - \hat{v}_i|}{v_i}, 1\right), \tag{1}$$

where N is the cardinality of the index set $\mathcal{I}$, i.e., the number of grains in the polycrystalline material, and $r_i' = \sqrt[3]{\frac{3v_i}{4\pi}}$ is the radius of a ball with volume v_i.

The intuition for this cost function is to quantify the discrepancy between the input grains and the corresponding Laguerre cells as relative errors for the volume and the centroids. Since the usual definition of relative error is not applicable for the latter, we consider the (Euclidean) distance between the input centroid and the centroid of the Laguerre cell and normalize it by the volume-equivalent radius to introduce a measure of the grain's "mean elongation." Both errors could become arbitrarily large and are thus bounded by one. This is necessary because, as we will show later, under certain circumstances the variance parameters $\sigma_1, \ldots, \sigma_m$ in the CE method depend on the cost function and, more precisely, large costs lead to high variances, which in turn lead to rather arbitrary tessellations having many empty cells. The two bounded errors are combined using a convex combination, in which the error of the centroids is weighted three times the error of the volumes. The idea is that the centroids correspond to three degrees of freedom instead of only one for the volumes. The total cost is then the mean cost of the individual cells.

2.3.2 Variance Injection and Stopping Conditions

In some cases, the variance of distribution $f(\mathbf{z};\theta)$ decreases too quickly, and the CE algorithm can hence converge to a local minimum instead of the global minimum $\mathbf{z}^*$. To avoid such behavior, we can use variance injection, a method introduced in Botev and Kroese (2004) and successfully employed, e.g., in Kroese et al. (2006); Spettl et al. (2016). With this method we occasionally increase the variance of the distribution $f(\mathbf{z};\theta)$ to ensure that more realizations of this distribution leave the neighborhood of a local minimum. Usually, variance injection is applied when the cost function c has not decreased significantly over a period of τ iterations. The size of the variance increment can depend on the current value of the cost function. If, as in our case, $c(\mathbf{z}) = \sum_{i=1}^{m} c_{loc}(z_i)$, where $\mathbf{z} = (z_1, \ldots, z_m)$ and $c_{loc} : \mathbb{R} \to \mathbb{R}$, the increment of the standard deviation σ_i can be proportional to the value $c_{loc}(z_i)$. When variance injection does not produce further improvements, the CE algorithm is terminated.

More precisely, denote $c_{min}^{(k)} = \min_{i=1,\ldots,n} c_D(\mathbf{z}^{(i)})$, where $(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)})$ is the sample of the k-th step. If

$$\left| \frac{c_{min}^{(k)} - \max_{t \in \{k-\tau,\ldots,k-1\}} c_{min}^{(t)}}{c_{min}^{(k)}} \right| < \delta_{inject}$$

for $\delta_{inject} > 0$, a variance injection is performed. Let $c_{loc,i} = c_{D,loc}(\hat{s}_i, \hat{v}_i)$ be the value of the local cost function of the i-th cell as defined in (1), and let $c_{loc,i}^* = \max_{j \in I_i} \{c_{loc,j}\}$ with $I_i = \{j \in \mathcal{I} : C_i \cap C_j \neq \emptyset\}$ be the maximal value of the cost function of cells in the neighborhood of the i-th cell (including C_i). Then, variance injection for the i-th cell increases the corresponding standard deviations (i.e. $\sigma_{4i-j}^{(k)}$ for $j = 0, \ldots, 3$) by $\kappa\sqrt{c_{loc,i}^*}$ where $\kappa > 0$. With this strategy, we increase the variance more for cells that do not fit the input data well or, more precisely, for generators that surround a badly reconstructed cell and thus have the most influence over its shape.

When variance injection does not improve the current state significantly—that is, if

$$\left| \frac{c_{min}^{(k)}}{c_{min}^{(k')}} \right| > \gamma,$$

where $\gamma \in (0, 1)$ and $c_{min}^{(k')}$ is the minimal cost function prior to the previous variance injection—we stop using it.

The CE algorithm is terminated when

$$\left| \frac{c_{min}^{(k)} - \max_{t \in \{k-\tau, \ldots, k-1\}} c_{min}^{(t)}}{c_{min}^{(k)}} \right| < \delta_{term},$$

where $\delta_{inject} > \delta_{term} > 0$.

Another possibility to prevent the CE algorithm from getting stuck in a local minimum is to use dynamic smoothing, see Botev and Kroese (2004); Kroese et al. (2006); Spettl et al. (2016).

2.3.3 Initial Configuration and Choice of Control Parameters

In order to keep the runtime short, it is desirable to start with a good initial tessellation. For that we employ the two-step reconstruction method, namely tessellation $\mathcal{T}_2$ from Section 2.2, which proved to give good results on the tested datasets.

The initial standard deviations for the i-th cell are chosen as $\sigma_{4i-j}^{(0)} = \sqrt{c_{loc,i}}$, where $j = 0, \ldots, 3$, and $c_{loc,i}$ is the value of the local cost function of this cell. For the sample size we use $n = 2000$, for the size of the elite set $\ell = 100$, for the variance injection threshold $\delta_{inject} = 0.05$, for the termination threshold $\delta_{term} = 0.01$, for the number of iterations considered for variance injection and termination $\tau = 10$, for the smoothing factor of the variance injection $\kappa = 0.25$, and for the stopping criterion of the variance injection $\gamma = 0.9$. Note that, in particular, higher values for the parameters n and ℓ can be used to trade short runtime against a better fit. The proposed set of values aims to

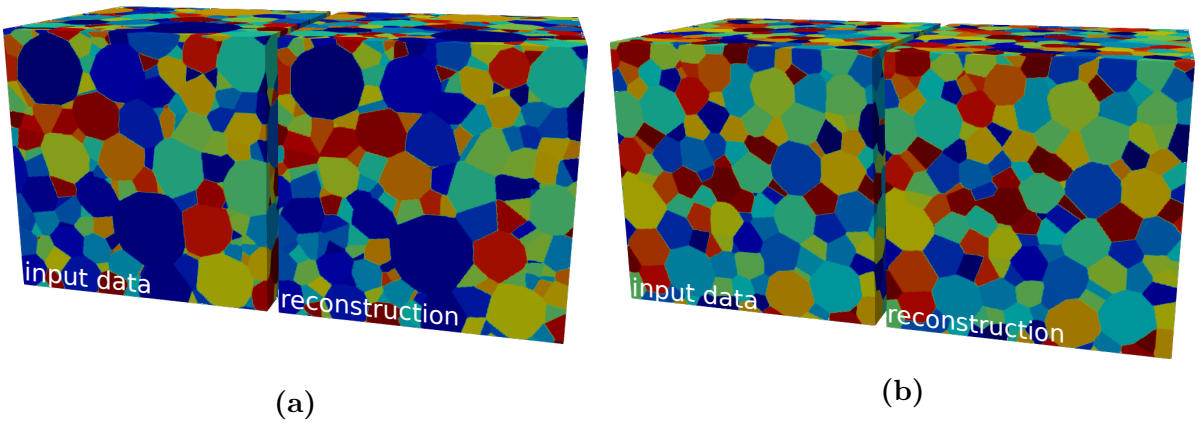
keep both balanced.

3 Results

In order to evaluate our reconstruction method, we use four different datasets. The first two datasets are realizations of random Laguerre tessellations, which allow us to quantify the performance of the algorithm under ideal circumstances. Based on two additional experimental datasets, the applicability of the method to real world data is investigated. In all four scenarios we use a set of centroids and volumes as input data and find a Laguerre tessellation that minimizes the cost function introduced in Section 2.3.1. In order to quantify the accuracy of the results, we compare characteristics of each input cell/grain in a given dataset and its corresponding cell of the tessellation gained from the reconstruction method. Note that because we want to evaluate the fit of the whole dataset and be consistent with the behavior of the cost function, all cells/grains are considered, including those at the boundary of the sampling window. Since we cannot assume that the considered optimization problem has a unique solution (apart form the parameterization of the generators), we analyze the stability of our method empirically. To this end, we run the procedure five times for each input dataset. Because the CE method is a Monte Carlo algorithm, different results are returned each time, and by comparing the results of the repetitions we get an estimate for the stability. All computations were performed on shared cluster nodes equipped with Intel Xeon E5-2670 octa-core processors. The parallel implementation of the algorithm employed twelve threads, which fully saturated twelve CPU cores.

3.1 Application to Laguerre Tessellations

In the first scenario we consider simulated data, i.e., two random Laguerre tessellations. Doing so, we know that the reconstruction problem has a solution and, in theory, we

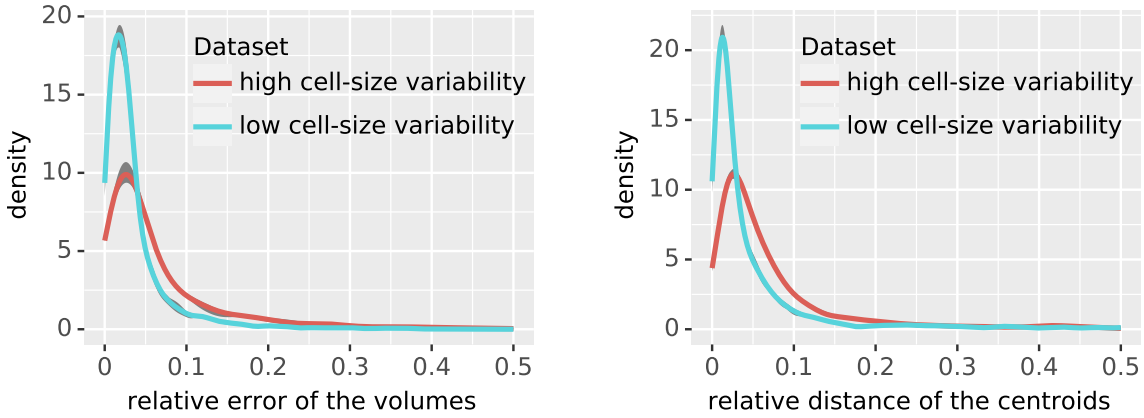


(a) **(b)**

Figure 2: 3D visualization of (simulated) input data (left tessellation) and reconstructed data (right tessellation), shown for datasets having (a) high and (b) low cell-size variability. Input cells and their corresponding reconstructed cells are drawn in the same color.

could find a perfect reconstruction. This allows us to quantify the accuracy of the reconstruction method without the errors introduced through the representation of grain boundaries by Laguerre tessellations.

In order to generate the data, we consider a bounded sampling window $W \subset \mathbb{R}^3$ with periodic boundary conditions and add random balls until the ratio of the total volume of all balls to the volume of W exceeds a predefined volume fraction ρ_{sim}. The midpoints of the balls are uniformly distributed in W, and the radii are gamma-distributed with some shape parameter $\alpha_{sim} > 0$ and rate parameter $\beta_{sim} > 0$. A force-biased algorithm (Bezrukov et al., 2002; Mościński et al., 1989) moves the balls until they no longer overlap. Their midpoints and radii are then considered as generators for a Laguerre tessellation. As input data for the reconstruction algorithm we compute the centroid and volume of each cell. Because we apply periodic boundary conditions to the point process but not to construction of the Laguerre tessellation, some marked points are copied and thus generate multiple cells. In our analysis, we used $W = [0, 99]^3$ and $\rho_{sim} = 0.63$. The parameters of the gamma-distribution were chosen such that the radii



(a) Relative errors of the volumes of the reconstructed cells.

(b) Distances of centroids of reconstructed cells to input centroids, relative to the volume-equivalent radii of the input cells.

Figure 3: Kernel density plots of error descriptors of the reconstructed tessellations for the simulated datasets. The gray bands show the difference between minimum and maximum density of the five repetitions for each reconstruction. The colored lines correspond to the mean densities.

of the first dataset ($\alpha_{sim} = 0.789, \beta_{sim} = 0.3$) have a high variance, while the second dataset ($\alpha_{sim} = 6.94, \beta_{sim} = 1.5$) is more homogeneous. The first one comprises 1702 cells, whereas the other one has 1439 cells.

The input datasets and their reconstructions are depicted in Figure 2. By visual inspection, it is apparent in both cases that the general structure is approximated very well. For a more detailed evaluation of the goodness of fit we consider the relative errors of the cell volumes and the distances of the output centroids to the input centroids relative to the volume-equivalent radius of the input cells. Note that both characteristics are part of the cost function considered in Section 2.3.1. Plots showing their kernel density estimates can be seen in Figure 3. These plots indicate that for both datasets the centroids and volumes of the reconstructed cells match those of the input cells quite well, but the more homogeneous dataset gives slightly better results. Another important criterion for a good fit is whether the number of reconstructed cells corresponds to the

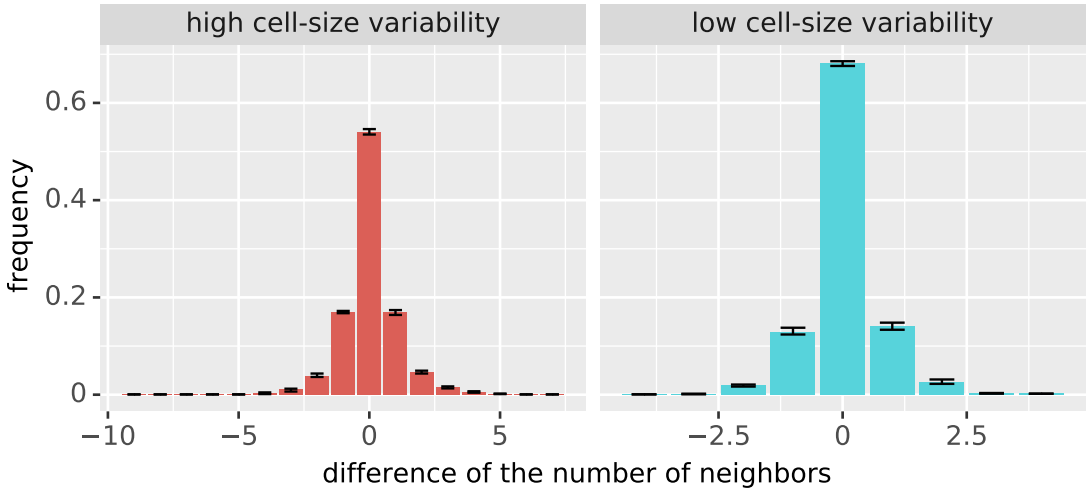


Figure 4: Histograms of the difference between the number of neighbors of the input cells and the number of neighbors of the reconstructed cells for the simulated datasets. Error bars show the minimum and maximum difference in the five repetitions of the reconstruction.

number of input grains (or, in this case, cells). Discrepancies can occur, since in a Laguerre tessellation not every generator produces a non-empty cell. For the dataset with high cell-size variability the mean number of empty cells is about 6.4 (0.37%), whereas the latter quantity is 1.2 (0.08%) for the low-variability case, which indicates good agreement with the input datasets. Note that these numbers are computed as the mean values of five repeated simulations for each of the two datasets.

Since both cell volumes and distances of the centroids are part of the cost function, it is desirable to look at other characteristics, as well. In Figure 4 histograms of the difference between the number of neighbors for the input cells and for the corresponding output cells are presented. To put this into perspective, each cell in the dataset with low cell-size variability has on average 11.3 neighbors with standard deviation 4.92, whereas in the dataset with with high cell-size variability each cell has on average 10.7 neighbors with standard deviation 8.23. It is apparent that the more homogeneous dataset has

fewer cells with non-zero difference, which indicate a better fit. But even for the more heterogeneous dataset, only for very few cells does the number of neighbors differ by more than two compared to the number of neighbors of the corresponding original cell. The barely visible gray bands in Figure 3 and the small error bars in Figure 4 are indicative of very little difference in the goodness-of-fit between the five repetitions of the reconstruction, which implies great stability of the CE method.

Computation of the reconstructions took about 19.2 h for the dataset with lower cell-size variability and 27.2 h for the more heterogeneous case. Since the program uses a Monte Carlo algorithm and ran on shared cluster nodes, the mean runtime of all five repetitions is presented in order to give a more accurate estimate.

3.2 Application to 3DXRD Image Data with Knowledge of Grain Boundaries

The experimental dataset with known grain boundaries was taken from a sample of aluminum alloyed with 5 wt% copper (AlCu), which had been homogenized for 24 h at 500°C and then cold-rolled to 50% reduction in thickness. Subsequently, a 1.4 mm-diameter cylindrical specimen was cut from the plate by spark erosion and annealed at 620°C for 10 min. Following this heat treatment, the sample was characterized at room temperature by 3D X-ray diffraction (3DXRD) microscopy carried out at beamline BL20XU of the synchrotron radiation facility SPring-8 in Japan. A 32-keV X-ray beam was employed to illuminate a layer of the sample 300 µm in height; ten such layers were recorded (with 50 µm overlap), resulting in a total measured specimen height of 1.55 mm. Within each layer, both far-field and near-field diffraction data were collected while the sample was rotated about its long axis by 360° in 0.48° steps. The crystallographic orientation and center of mass of individual grains were extracted from far-field data using the program GrainSpotter (Schmidt, 2014), and 3D grain morphologies were

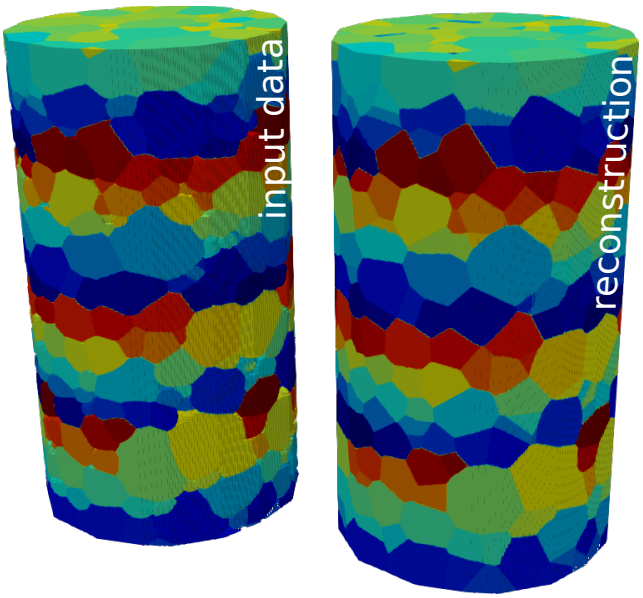
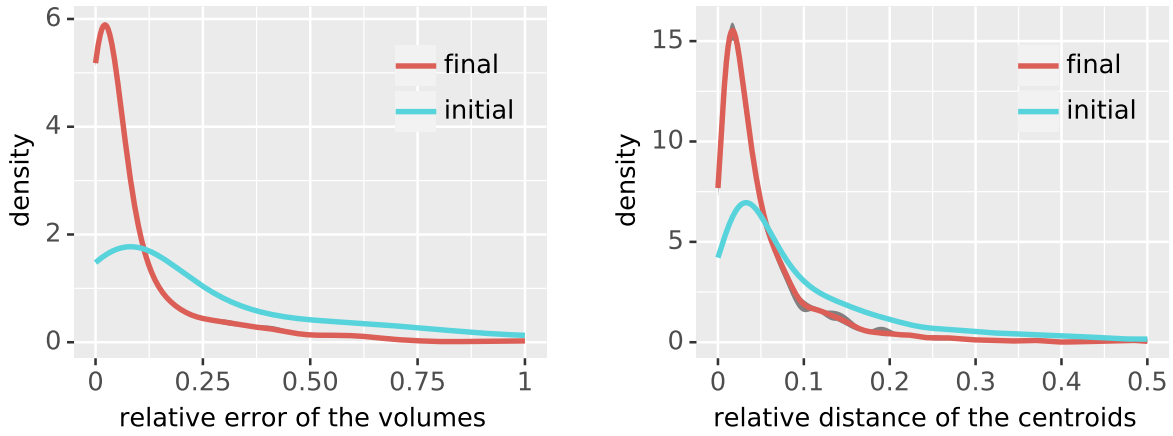


Figure 5: 3D visualization of the (experimental) input data (left) and of the reconstructed data (right) for an AlCu dataset with known grain boundaries. Input grains and their corresponding reconstruction as cells of a Laguerre tessellation are drawn in the same color.

reconstructed from near-field data via GrainSweeper (Schmidt et al., 2008), see Figure 5. Following reconstruction, the individual layers were stitched together, yielding a map containing 928 grains.

The cylindrical window was approximated with an 18-sided prism. This ensured that the cells at the boundary of the window are still polyhedra and the formulas for computing the volume and centroid remain the same. Similar to the previous section, we compute the volumes and centroids of each grain restricted to the sampling window. These act as input data for our reconstruction procedure. However, with this dataset we can evaluate how the reconstructed cell boundaries compare to the grain boundaries of real-world data.

Comparing the measured dataset and the reconstructed dataset visually (Figure 5), we observe a good fit. A more in-depth analysis of the relative errors of the volumes



(a) Relative errors of the volumes of the reconstructed cells.

(b) Distances of centroids of reconstructed cells to input centroids, relative to the volume-equivalent radii of the cells.

Figure 6: Kernel density plots of error descriptors of the initial tessellation (two-step method) and the final reconstructed tessellation for the experimental dataset with known grain boundaries. The gray bands show the difference between minimum and maximum density of the five repetitions for each reconstruction. The red lines correspond to the mean densities.

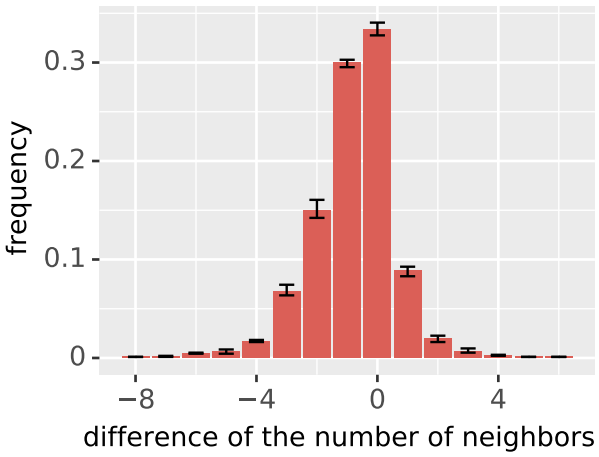


Figure 7: Histogram of the difference between the number of neighbors of the input grains and the number of neighbors of the reconstructed cells for the experimental dataset with known grain boundaries. Error bars show the minimum and maximum difference in the five repetitions of the reconstruction.

shows slightly larger errors than for the simulated datasets considered in Section 3.1. Nevertheless, the centroids were fitted with the same precision as for the simulated data, and in both cases the errors are significantly lower compared to the initial configuration from the two-step method, see Figure 6. The (mean) number of empty cells for the experimental dataset with known grain boundaries is about 2.8 (0.3%) and is thus quite similar to that of the high cell-size variability dataset described above.

In Figure 7 it is shown that the difference in number of neighbors is scattered over a wider range than for the simulated datasets and that the reconstructed cells tend to have slightly more neighbors than in the experimental input dataset. Here, the number of neighbors of the input dataset is on average 10.6 with standard deviation 5.54. Still, for 89% of the cells, the number of neighbors differs between reconstruction and experiment by less than two.

For this dataset the reconstruction took on average only about 6.9 h to compute.

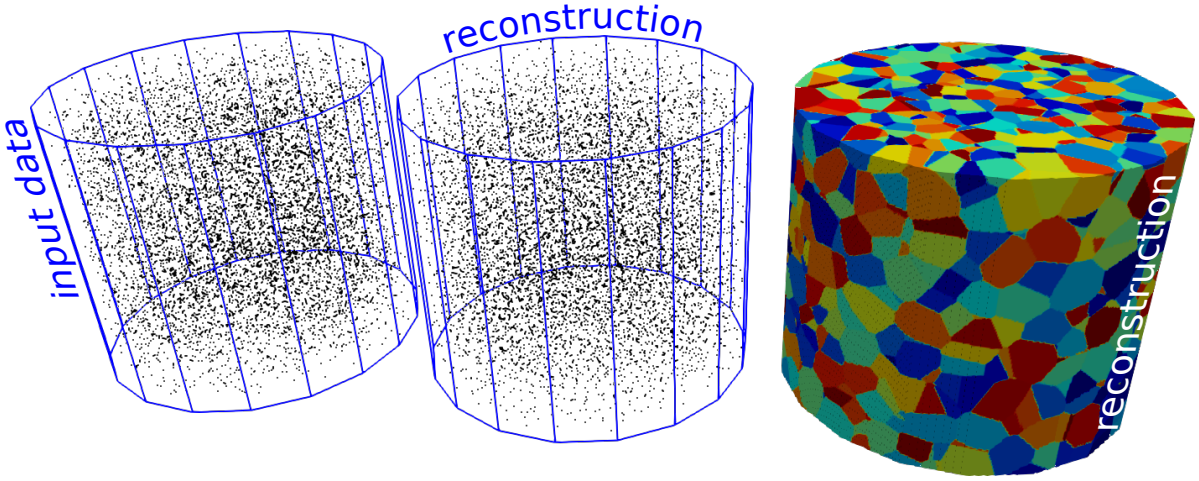
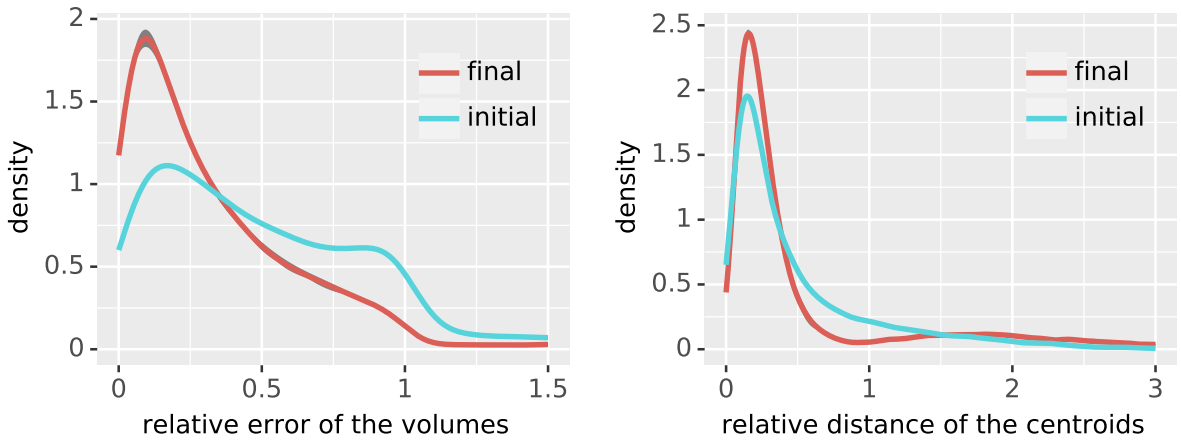


Figure 8: 3D visualization of the input centroids (left), the reconstructed centroids (center) and the reconstructed grain boundaries (right) of the NiTi dataset with unknown grain boundaries.

3.3 Application to 3DXRD Image Data without Knowledge of Grain Boundaries

The fourth scenario is a cylindrical cutout (diameter = 100 µm, height = 85 µm) comprising 8063 grains from a larger cylindrical far-field 3DXRD measurement with about 15000 grains of a nickel titanium (NiTi) wire subjected to 425 MPa tensile stress. In contrast to the previous datasets, only the centers of mass and volumes in addition to elastic strain and stress tensors for each grain were determined. Consequently, no information about the grain boundaries is available for validation.

The dataset and its reconstruction are depicted in Figure 8. Density plots of the relative errors of the volumes and the relative distances of the centers of mass are shown in Figure 9. Especially when considering the errors in volume, it is apparent that the program has a much harder time finding a good tessellation for NiTi than for the AlCu sample considered above; nevertheless, the reconstruction is much better than the initial configuration from the two-step method. One explanation for this could be that Laguerre



(a) Relative errors of the volumes of the reconstructed cells.

(b) Distances of centroids of reconstructed cells to input centroids, relative to the volume-equivalent radii of the grains.

Figure 9: Kernel density plots of error descriptors of the initial tessellation (two-step method) and the final reconstructed tessellation for the experimental dataset with unknown grain boundaries. The gray bands show the difference between minimum and maximum density of the five repetitions for each reconstruction. The red lines correspond to the mean densities.

tessellations having flat cell boundaries might not approximate the grain boundaries of the sample well enough. This is supported by the fact that the method proposed by Quey and Renversade (2018), which will be discussed in the next section, also struggles with matching the volumes, as can be seen in Figure 11. While the majority of the centroids lie in close proximity to their input counterparts, a few show high relative errors of about 2. In all five repetitions of the reconstruction an average of 52.6 (0.0065%) empty cells is observed, which is the lowest percentage of all considered datasets.

Note that in order to reduce the runtime and maintain it within the cluster's time limit of six days, we changed the sample size n of the CE method from 2000 to 1500. This way, the computation took on average about 122.3 h.

4 Discussion

In this section we discuss the results presented in the previous section, compare them with other approaches from literature and suggest possible extensions to our method.

4.1 Analysis of Goodness-of-Fit and Stability

In the previous section we have seen that the proposed reconstruction method is able to find Laguerre tessellations that capture the characteristics of the input datasets reasonably well. The fit degrades slightly for the experimental data, but this is to be expected, as real grain boundaries are not perfectly described by Laguerre tessellations. The corresponding error in representation adds to the error of the optimization itself.

Even though it is not clear whether there exists a unique solution to the posed optimization problem, all considered characteristics (in particular those that are not part of the cost function) show very little scatter between individual repetitions of the reconstruction. Because the CE method is a global optimization algorithm, it has the ability to find different tessellations with the same volumes and centroids. But since the latter

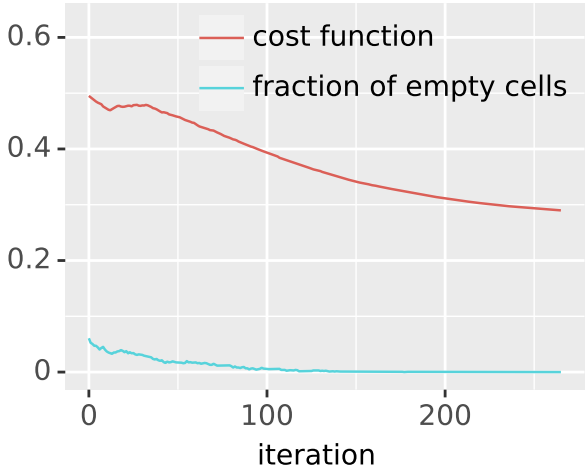


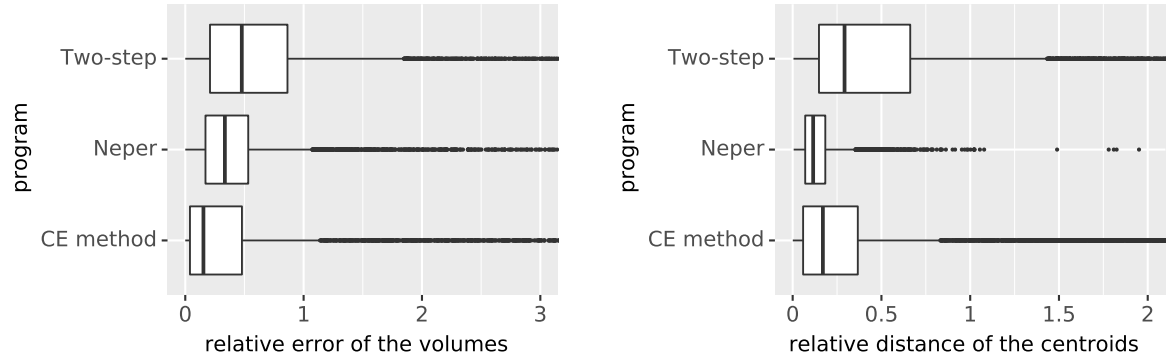
Figure 10: Cost function and percentage of empty cells during optimization for the dataset with unknown grain boundaries. The points at iteration zero correspond to the initial Laguerre tessellation from the two-step reconstruction method considered in Section 2.2.

possibility did not actually occur, we conclude that the non-uniqueness of the solution is of minor practical relevance.

As to be expected, the runtime of the reconstructions depends on the number of grains in the input dataset. Even without considering that the program ran on shared computers, the time it took to terminate is already acceptable. However, it could easily be sped up further by employing more processors.

4.2 Comparison to Other Approaches from Literature

In contrast to the approach proposed in the present paper, the two-step reconstruction method proposed in Lyckegaard et al. (2011), see Section 2.2, is a very fast, heuristic approach that does not aim to find the best result. We use this property to kick-start the optimization in order to find tessellations that better fit the data. This can



(a) Relative errors of the volumes of the reconstructed cells.

(b) Distances of centroids of the reconstructed cells to input centroids, relative to the volume-equivalent radii of the grains.

Figure 11: Box plots of error descriptors of the reconstructed tessellations from the two-step method, Neper and the CE method proposed in the present paper for the NiTi dataset. Note that the high number of outliers is caused by the skewness of the distributions.

easily be seen, e.g., for the NiTi dataset in Figures 9 and 11, but also by considering the decrease in value of the cost function during the optimization procedure (Figure 10), where the initial value corresponds to the results obtained by the two-step reconstruction method. Here, we show the results for the NiTi dataset because it best represents real world applications. For some datasets, the two-step method produces a relatively large number of empty cells, which the CE method is able to reduce. This behavior is also depicted in Figure 10. The cost for the better fit, however, is a much longer runtime of several hours, compared to only a few seconds for the two-step reconstruction method.

In Quey and Renversade (2018) a generic algorithm is presented that constructs Laguerre tessellations that fit properties of experimental datasets, in particular the volumes and centroids of the grains. For this purpose, a different optimization problem is considered, in which the cost function c_D^{Neper} of the generator vector $G_{\mathcal{T}}$ of the Laguerre

tessellation $\mathcal{T}$, given the input dataset D, is defined by

$$c_D^{Neper}(G_{\mathcal{T}}) = \frac{1}{2N\bar{r}'}\sqrt{\sum_{i\in\mathcal{I}}\left(\|s_i - \hat{s}_i\|^2 + (r_i' - \hat{r}_i')^2\right)} + pN_e,$$

with the centroid s_i and the volume-equivalent radius r_i' of the i-th input grain, the centroid $\hat{s}_i$ and the volume-equivalent radius $\hat{r}_i'$ of the corresponding i-th Laguerre cell, and the mean volume-equivalent radius $\bar{r}'$ of the $N = \#\mathcal{I}$ input grains. The term pN_e multiplies a penalization factor $p \geq 0$ by the number of empty cells N_e, thus punishing empty cells. Furthermore, instead of applying a global Monte Carlo approach, in Quey and Renversade (2018) a local optimization procedure is suggested with a sophisticated method for propagating changes in the tessellation, thereby allowing for efficient recalculation of the cost function. However, this comes at the cost of a higher susceptibility to local minima. An implementation of their procedure is available in the open-source software "Neper." In Figure 11 box plots of the errors of the reconstructed Laguerre tessellations for the NiTi dataset from Neper and the CE method proposed in the present paper are shown. The relative errors of the volumes are smaller for the CE method, but Neper does a better job at fitting the centers of mass. One reason for this is that the corresponding cost functions weight the two types of errors differently. Note that we constrained the reconstruction procedures such that the runtime does not exceed the cluster's time limit of six days. We thus stopped Neper after this time and reduced the sample size for the CE method from 2000 to 1500, which led to a runtime of about five days. Under these circumstances both programs produce reasonable reconstructions. Given more CPU time both probably would have returned a more accurate fit. However, one advantage of the CE method is that it scales very well with the number of CPU cores. This means that if more CPUs are available the real runtime can be cut down significantly. This becomes especially important when dealing with huge datasets ($> 10\,000$ grains). The same order of parallelization is not possible with the sequential

updates that Neper uses.

4.3 Possible Extensions

Since our method relies only on centroids and volumes of tessellation cells, it is easy to adapt the procedure to more flexible tessellation models (see e.g. Šedivý et al., 2018), which may be better suited to the microstructure of a given experimental dataset. For example, if a material is known to have curved grain boundaries, one might choose spherical growth tessellation models (Teferra and Graham-Brady, 2015) instead of the flat-faced Laguerre tessellations to achieve a more accurate fit. The only requirement imposed by the proposed CE method on the tessellation models is that for each generator the volume and centroid of the corresponding cell can be computed, but even if no closed formulas for the volumes and centroids are available, a discretized approximation of the tessellation may be used to compute the cost function. The resulting increase in computational complexity can be handled only by highly parallel algorithms, such as the cross-entropy method, since other approaches—like the one proposed in Quey and Renversade (2018)—would lead to impractically long runtimes. In a future paper we will extend the CE approach to other tessellation models and study the results for datasets having curved grain boundaries.

Furthermore, the weights in the cost function can be changed to return a better fit, e.g., with respect to the volumes of the grains. Going even further, the cost function could be tailored to incorporate additional information, should additional characteristics describing the grain boundaries be part of the measurement.

5 Conclusion

In this work we presented a method for finding the generators of a Laguerre tessellation for a given set of cell volumes and centroids. This is of interest when reconstructing grain

boundaries from far-field 3DXRD measurements yielding only the volumes and centers of mass for a large number of grains, but not their shapes and positions. The resulting Laguerre tessellation can then be used for a finite element analysis of the mechanical behavior of the specimen. It was shown that the reconstructed Laguerre tessellations fit the actual grain morphologies well. Furthermore, by employing the highly parallel cross-entropy optimization, our approach can utilize a large number of CPU cores, making it a good solution for big datasets. In a future paper we will extend the proposed method to more general tessellation models that potentially allow for more accurate fits of complex grain microstructures.

Acknowledgements

This research was partially funded by the German Science Foundation (DFG, project number SCHM 997/23-1) and the Czech Science Foundation (GACR, project number 17-00393J). We are grateful to the Japan Synchrotron Radiation Research Institute for the allotment of beam time on beamline BL20XU of SPring-8 (Proposal 2015A1580). The authors acknowledge support by the State of Baden-Württemberg through bwHPC.

References

Abdolvand, H., Majkut, M., Oddershede, J., Schmidt, S., Lienert, U., Diak, B. J., Withers, P. J., and Daymond, M. R. (2015). On the deformation twinning of Mg AZ31B: a three-dimensional synchrotron X-ray diffraction experiment and crystal plasticity finite element model. *International Journal of Plasticity*, 70:77–97.

Bezrukov, A., Bargieł, M., and Stoyan, D. (2002). Statistical analysis of simulated random packings of spheres. *Particle & Particle Systems Characterization*, 19(2):111–118.

Botev, Z. and Kroese, D. P. (2004). Global likelihood optimization via the cross-entropy method, with an application to mixture models. In Ingalls, R. G., Rossetti, M. D., Smith, J. S., and Peters, B. A., editors, *Proceedings of the 2004 Winter Simulation Conference*, pages 529–535. IEEE.

Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Duan, Q., Kroese, D. P., Brereton, T., Spettl, A., and Schmidt, V. (2014). Inverting Laguerre tessellations. *The Computer Journal*, 57(9):1431–1440.

Evans, G. E., Keith, J. M., and Kroese, D. P. (2007). Parallel cross-entropy optimization. In Henderson, S. G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J. D., and Barton, R. R., editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 2196–2202. IEEE.

Johnson, G., King, A., Honnicke, M. G., Marrow, J., and Ludwig, W. (2008). X-ray diffraction contrast tomography: a novel technique for three-dimensional grain mapping of polycrystals. II. The combined case. *Journal of Applied Crystallography*, 41(2):310–318.

Kroese, D. P., Porotsky, S., and Rubinstein, R. Y. (2006). The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, 8(3):383–407.

Lautensack, C. and Zuyev, S. (2008). Random Laguerre tessellations. *Advances in Applied Probability*, 40(03):630–650.

Ludwig, W., King, A., Reischig, P., Herbig, M., Lauridsen, E. M., Schmidt, S., Proudhon, H., Forest, S., Cloetens, P., Rolland du Roscoat, S., Buffière, J. Y., Marrow, T. J., and Poulsen, H. F. (2009). New opportunities for 3D materials science of polycrystalline materials at the micrometre lengthscale by combined use of X-ray diffraction and X-ray imaging. *Materials Science and Engineering: A*, 524(1-2):69–76.

Lyckegaard, A., Lauridsen, E. M., Ludwig, W., Fonda, R. W., and Poulsen, H. F. (2011). On the use of Laguerre tessellations for representations of 3D grain structures. *Advanced Engineering Materials*, 13(3):165–170.

Møller, J. (1994). *Lectures on Random Voronoi Tessellations*. Springer, New York.

Mościński, J., Bargieł, M., Rycerz, Z. A., and Jacobs, P. W. M. (1989). The force-biased algorithm for the irregular close packing of equal hard spheres. *Molecular Simulation*, 3(4):201–212.

Poulsen, H. F. (2004). *Three-Dimensional X-Ray Diffraction Microscopy*. Springer, Berlin, Heidelberg.

Quey, R. and Renversade, L. (2018). Optimal polyhedral description of 3D polycrystals: Method and application to statistical and synchrotron X-ray diffraction data. *Computer Methods in Applied Mechanics and Engineering*, 330:308–333.

Rubinstein, R. Y. and Kroese, D. P. (2004). *The Cross-Entropy Method*. Springer, New York.

Schmidt, S. (2014). GrainSpotter: a fast and robust polycrystalline indexing algorithm. *Journal of Applied Crystallography*, 47(1):276–284.

Schmidt, S., Olsen, U. L., Poulsen, H. F., Sørensen, H. O., Lauridsen, E. M., Margulies, L., Maurice, C., and Jensen, D. J. (2008). Direct observation of 3-D grain growth in Al-0.1% Mn. *Scripta Materialia*, 59(5):491–494.

Šedivý, O., Westhoff, D., Kopeček, J., Krill III, C. E., and Schmidt, V. (2018). Data-driven selection of tessellation models describing polycrystalline microstructures. *Journal of Statistical Physics*, 172(5):1223–1246.

Sedmák, P., Pilch, J., Heller, L., Kopeček, J., Wright, J., Sedlák, P., Frost, M., and Šittner, P. (2016). Grain-resolved analysis of localized deformation in nickel-titanium wire under tensile load. *Science*, 353(6299):559–562.

Spettl, A., Brereton, T., Duan, Q., Werz, T., Krill III, C. E., Kroese, D. P., and Schmidt, V. (2016). Fitting Laguerre tessellation approximations to tomographic image data. *Philosophical Magazine*, 96(2):166–189.

Teferra, K. and Graham-Brady, L. (2015). Tessellation growth models for polycrystalline microstructures. *Computational Materials Science*, 102:57–67.