Stochastic Modeling of Multiphase Materials Based on Digital Image Data

Vom Fachbereich Mathematik der Technischen Universität
Kaiserslautern zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation von

Katharina Losch

Gutachter:
Prof. Dr. Claudia Redenbach
Prof. Dr. Markus Kiderlen

Disputation: 17. Juli 2017

D 386

Acknowledgments

I gratefully acknowledge the funding received from the Deutsche Forschungsgemeinschaft within the GrK 1932 and from the Fraunhofer ITWM within a scholarship. I'm very grateful for this funding. The Fraunhofer ITWM also provided me with a working place and access to various programs, such as MAVI, ToolIP and MAVIlib. I am very thankful to the Fraunhofer ITWM department of image analysis for allowing me to work in this group.

All samples were prepared at Fachbereich Maschinenbau within the Lehrstuhl für Werkstoffkunde. The SEM images were also obtained there.

The FIB-SEM images were otained within Nanostructuring Center at Technische Universität Kaiserslautern by Dr. Thomas Henning Löber.

The decurtained FIB-SEM images were kindly provided by Jan-Henrik Fitschen.

The synchrotron images were obtained by Dr. Alexander Rack at ESRF, Grenoble.

Preface

I would like to thank Claudia Redenbach and Katja Schladitz for their constant support, especially with regards to questions concerning stochastic geometry, spatial and general statistics and image analysis.

Furthermore I would like to thank Luis Cruz-Orive for interesting conversations and references to literature with regards to stereology and Oliver Wirjadi for technical support.

Additionally, I would like to thank Jan-Henrik Fitschen, Sebastian Schuff, Frank Balle and Gabriele Steidl for the excellent cooperation within our project group at GrK 1932.

Finally I would like to thank Erwin Kraft and my family for their emotional support and advice on good C++ programming style.

Contents

Notations

f	grey value or binary image
ϵ_S	erosion by a structuring element S
δ_S	dilation by a structuring element S
γ_S	opening by a structuring element S
ϕ_S	closure by a structuring element S
V	volume
S	surface area
M	integral of mean curvature
χ	Euler number
v_q	quartile dispersion coefficient
cv	coefficent of variation ($= \frac{\hat{\sigma}}{\hat{\mu}}$)
Al	Aluminium
SiC	Silicon carbide
Cu	Copper
$\mathbf{F}_{loc}$	local adjacency system
$\mathbf{F}$	adjacency system
$B_r^d(x)$	d-dimensional (closed) ball of radius r centered at x.
ω_d	Lebesgue measure of the d-dimensional unit ball $B_1^d(0)$ in $\mathbb{R}^d$
$\mathbb{K}$	compact (closed and bounded) subsets of $\mathbb{R}^d$
$\mathcal{K}$	compact convex subsets of $\mathbb{R}^d$, also called *convex bodies*
$\mathcal{K}' = \mathcal{K} \setminus \emptyset$	non-empty convex bodies
$\mathcal{K}'_{\mathbf{0}}$	non-empty convex bodies with centre $\mathbf{0}$
$d_H(A, B)$	Hausdorff distance between the sets A and B
$\mathcal{H}_d^k$	k-dimensional Hausdorff measure in space of dimension d
ν_d	d-dimensional Lebesgue measure
$\mathbb{F}$	closed subsets of $\mathbb{R}^d$
$\mathbb{O}$	open subsets of $\mathbb{R}^d$
$V_k(K)$	intrinsic volumes for convex body K
AMC17	SiC-particle reinforced Al alloy with 17vol% reinforcement
$\mathbb{1}_A$	characteristic function of set A, $\mathbb{1}_A(x) = 1$ if $x \in A$, $\mathbb{1}_A(x) = 0$ else

Introduction

Overview

Multiphase materials combine properties of several materials, which makes them interesting for high-performing components. This thesis considers a certain set of multiphase materials, namely Silicon-carbide (SiC) particle-reinforced Aluminium (Al) metal matrix composites and their modeling based on stochastic geometry models.

Stochastic modeling is used for the generation of virtual material samples: Once we have fitted a model to the material statistics, we can obtain independent three-dimensional "samples" of the material under investigation without the need of any actual imaging. Additionally, by changing the model parameters, we can easily simulate a new material composition.

The materials under investigation have a rather complicated micro-structure, as the system of particles has many degrees of freedom: Size, shape, orientation and spatial distribution. Additionally, the metal matrix has its own grain micro-structure with Aluminium-Copper (Al_2Cu) precipitations occurring on the grain boundaries. In order to identify an adequate model, we have to take into account the failure behaviour of the material, since calculations based on our model should lead to the same failure results as experiments on the material.

Material failures can occur via various ways: cracks can appear first on phase boundaries, specifically between the Al grains or at the boundary between SiC and Al. Another mechanism of failure is the breaking of SiC grains under load. Usually, the failure is a combination of these mechanisms, see [Kai03, BGS04, SS92, HWS94].

Due to the specific size of the SiC particles, which have an nominal average equivalent spherical diameter of less than 3.0 μm, the materials under investigation cannot be imaged by standard CT-imaging, which only allows for a resolution of $1-2$ μm, in special cases of synchrotron radiation of 0.3 μm. Instead, most image data is obtained by two-dimensional SEM imaging. Three-dimensional images can be obtained by the FIB-SEM method, however, this method is very time-consuming and cost-intensive and is additionally quite limited in the size of the obtained three-dimensional image. This means that images obtained by FIB-SEM are too small to adequately sample particles with an equivalent diameter of 3.0 μm. For these cases, model parameters have to be estimated from two-dimensional sections alone. For samples with particles of an equivalent diameter of 0.7 μm or 0.3 μm the FIB-SEM

image size and resolution are however well-suited.

In this thesis, we present a stochastic micro-structure model aimed for particle-reinforced metal-matrix composites, based on the specific examples of SiC-reinforced Al alloys. Modeling engineering materials by stochastic geometry models is a widespread approach to generate virtual samples for simulations: For example, in [Esc12] grains in concrete are modelled using random polytopes, in [Lau07, LR12, Lie14] foams are modelled by random tessellations, and in [AJ10] a model for densely packed bending fibres was introduced. Models for the particle system of a SiC-particle reinforced Al alloy were presented in [CC06, CSG06], where finite element simulations on binarised FIB-SEM images were compared to modeling the particles by ellipsoids and spheres.

We model the reinforcement particles in terms of size, shape and orientation distribution. Additionally, we show how to incorporate matrix grains into the model and propose a method to model precipitations that often occur on phase boundaries. There are several ways to produce this type of material, in our case we investigate a composite that was extruded, see [Kai03]. The extrusion process adds a small anisotropy with regards to the particle orientation, which we incorporated into our model. Also, our model is easily adapted to other metal-matrix composites. Since for many materials it is not easy to obtain 3D images, we present methods to estimate parameters from 2D section images.

In the following, we will introduce the materials under investigation in detail. In Chapter 1 we will present the mathematical foundations needed to establish the models. This includes basics in stochastic geometry, spatial statistics, and morphological image analysis. In Chapter 2 we present an overview of already existing methods to prepare image data for measurement in 3D and 2D. In Chapter 3 we present our model for the matrix' grain distribution, particle distribution and precipitations. In Chapter 4 we show methods how to estimate the model parameters from 2D sections. Chapter 5 presents our results and conclusions. Parts of Chapter 2 and Chapter 3 are submitted to appear in [LSB$^+$ed].

Materials under investigation

The subject of our investigations are particle reinforced metal matrix composites, specifically SiC reinforced Aluminium alloys. These materials consist of an Al-Cu wrought alloy with SiC particles embedded in the matrix. The material is produced using powder metallurgy techniques: The components are mixed as fine powders at high temperatures and compacted to billets isostatically, while staying below the melting temperature of the Al alloy and the SiC particles. This ensures an even distribution of the particles within the matrix. Finally, the billets are extruded, which causes an anisotropic orientation of the SiC particles. Details of the techniques mentioned can be found in [HK04, Mat, TGC02].

Due to the particle reinforcement, these materials have a higher yield strength than Al that was not reinforced, while preserving the light weight of the Al. Usually, the reinforcement phase of this type of material accounts for up to 25% of the total vol-

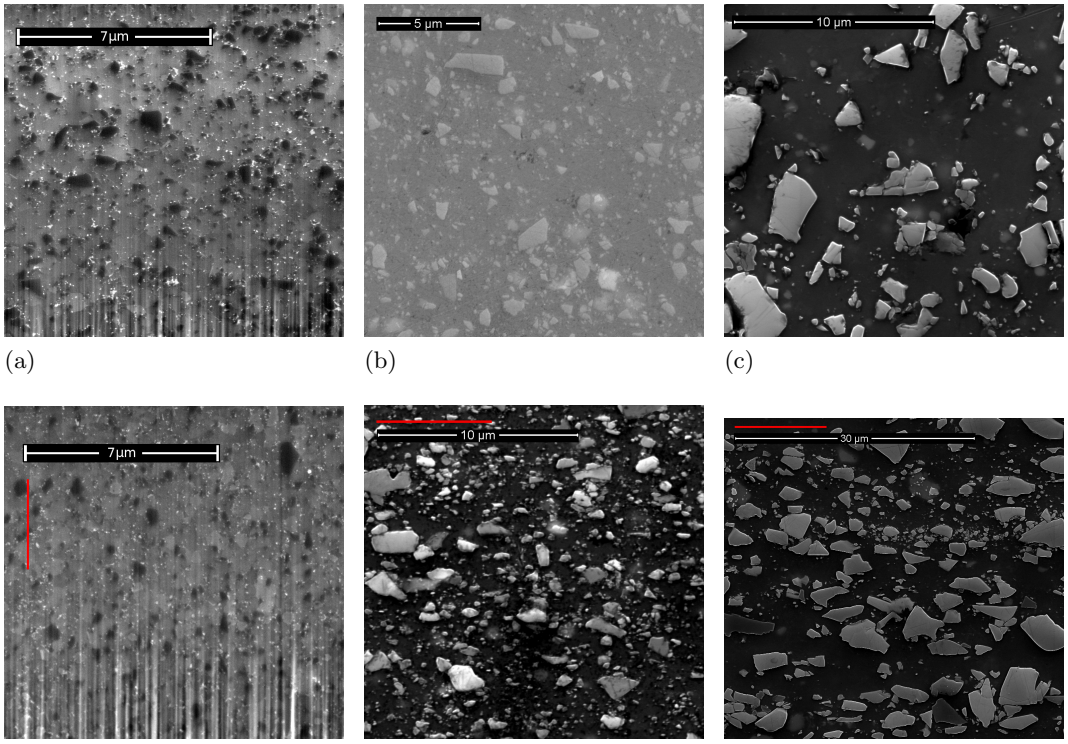


Figure 1: Detail of SEM images of *AMC17* (a),(d): *AMC17xxfine*, obtained within an FIB-SEM. (b),(e): *AMC17xfine*. (c),(f): *AMC17xe*. Top: transverse section, Bottom: Longitudinal Section. The extrusion axis is parallel to the red line section.

ume of the material. In the samples investigated for this thesis, the volume content of the reinforcement phase was set to 17%. During the cooling of the material, the Al alloy formed Al grains with Al_2Cu precipitations on the grain boundaries.

There were three different types of materials under investigation, which differ by size and potentially also by shape of the reinforcement particles. The nominal particle size, measured on 2D sections as the diameter of the corresponding circle, and stated by the manufacturer was 0.3, 0.7 and 3.0 μm. The technical names of the materials are *AMC17xxfine* (at 0.3 μm equivalent diameter), *AMC17xfine* (at 0.7 μm equivalent diameter), and *AMC17xe* (at 3.0 μm equivalent diameter). By *AMC17* we will refer to the general class to which these materials belong, namely the Al alloy reinforced with 17vol% SiC particles.

Imaging methods and data

SEM images

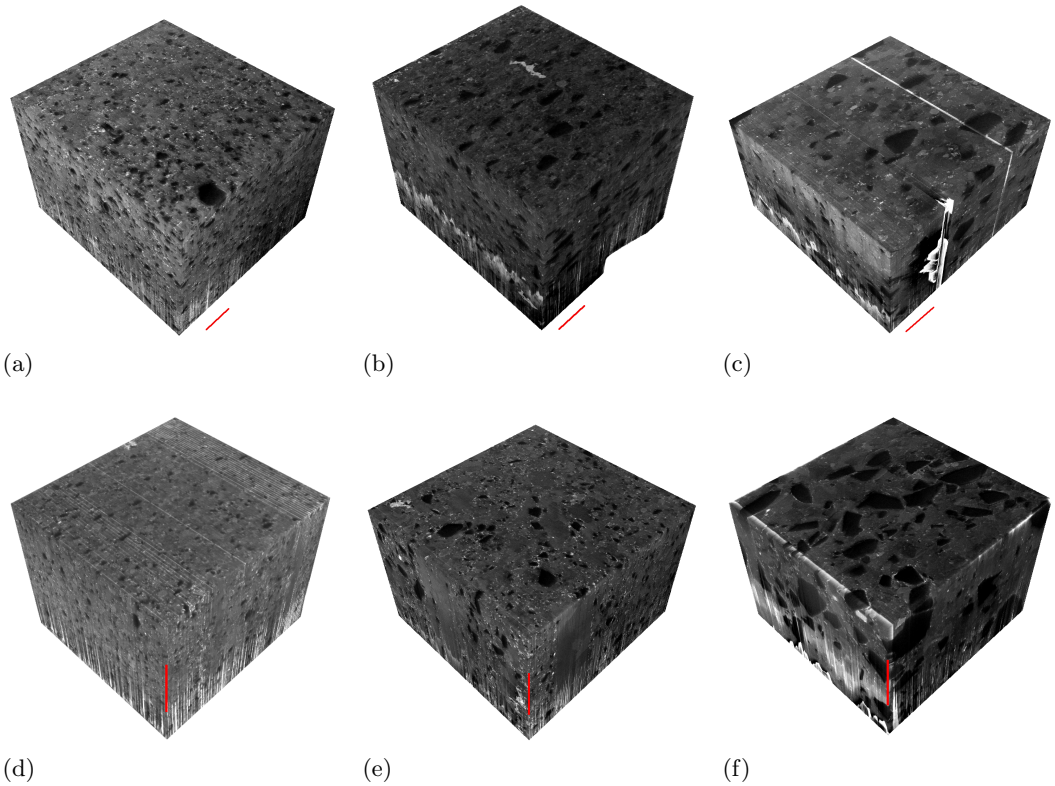
The standard method to investigate samples of *AMC17* is scanning electron microscope (SEM) imaging, yielding 2D images at high resolutions. Due to the anisotropy introduced by the extrusion, the SEM images are obtained on longitudinal as well as on transverse sections of the material, where the longitudinal sections include

the extrusion direction, while the transverse sections are perpendicular to it. In Figure 1 the materials are displayed based on images of transverse and longitudinal cuts. Note that the resolutions are different for each material, so that the particles can be resolved well for each material composition.

FIB-SEM images

The method of imaging mentioned above, which results in 3D images, is using the focused-ion-beam-SEM (FIB-SEM) set-up. In this set-up, the focused ion beam is used to abrade the sample layer by layer. Between the abrasion processes, images of the current layer are obtained via SEM. This yields a stack of SEM images that are combined to a 3D image. Theoretically, the SEM image could have a resolution with pixel edge length of 1 nm, but the focused ion beam's resolution is about 50 nm. At the same time, each milling and imaging takes some time, which puts a limit on the number of slices that can be used before the ion beam becomes unstable (personal communication with S. Schuff and T. H. Löber). Generally, about 400 slices are obtained. At a resolution of approximately 15×15 $(nm)^2$ of the SEM slice, and at a FIB resolution of 50 nm, the images obtained by FIB-SEM are of the size 13.6 $\mu m \times 11$ $\mu m \times 17.5$ μm.

All three materials are depicted in Figure 2 with images obtained by FIB-SEM. Note that the difference in resolutions along the image axes stems from the fact that the layers abraded by the ion beam are thicker than the SEM resolution.



(a) (b) (c)

(d) (e) (f)

Figure 2: Visualisations of FIB-SEM images' subvolumes of *AMC17* (a),(d): *AMC17xxfine*. (b),(e): *AMC17xfine*. (c),(f): *AMC17xe*. Top: SEM slices were of the transverse section, Bottom: SEM slices were of the longitudinal section. The extrusion axes are parallel to the red lines. The visualisation was done in MAVI, [MAV05].

Chapter 1

Theoretical foundations of spatial statistics and stochastic geometry

The following is cited from [SW08,CSKM13] (Random closed sets, point processes to Laguerre tessellations), [OS09] (measures on convex sets), [Lau07] (random Laguerre tessellations), [OM00] (stereology) and [TY09] (digital images and their analysis). Another source on point processes is [BR04]. All figures in the chapter were generated using MAVI, [MAV05], unless explicitly stated otherwise.

1.1 Basic notions in $\mathbb{R}^d$ and set theory

In the following, we will introduce the relevant concepts of stochastic geometry in the Euclidean vector space $\mathbb{R}^d$ of dimension d. Vectors and points in $\mathbb{R}^d$ are displayed in bold font, while their coordinates are displayed non-bold. Note that in this space, the scalar product $\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + \ldots + x_d y_d$ induces the Euclidean norm $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ and the distance

$$\mathrm{d}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$$

We denote the origin by $\mathbf{0}$. Additionally, we can define the distance of a point $\mathbf{x}$ and a set A by $dist(\mathbf{x}, A) = \inf_{\mathbf{y} \in A} \mathrm{d}(\mathbf{x}, \mathbf{y})$, so the distance between a point $\mathbf{x}$ and a set A is the distance between the point $\mathbf{x}$ and the point $\mathbf{y}$ in A that is closest to $\mathbf{x}$. Then, we can also define the distance between two sets A and B by the so-called *Hausdorff distance*:

$$d_H(A, B) = \max \left\{ \sup_{\mathbf{x} \in B} dist(\mathbf{x}, A), \sup_{\mathbf{y} \in A} dist(\mathbf{y}, B) \right\}, \text{ see [BR04].}$$

For a general set A, its *interior* $\mathring{A}$ is the union of all open sets in A. The *closure* $\bar{A}$ of A is the intersection of all closed sets containing A. For a finite set A, by $\sharp A$ we denote the number of elements of that set.

The *Minkowski addition* $A \oplus B$ is defined as

$$A \oplus B = \bigcup_{b \in B} \{a + b | a \in A\}$$

and the *Minkowski subtraction* $A \ominus B$ as

$$A \ominus B = \bigcap_{b \in B} \{a + b | a \in A\},$$

see [CSKM13].

1.2 Random Closed Sets

Random Closed Sets (RACS) provide the basis for many stochastic geometry models. For example, if we want to model grains of sand, we would like to model the random shape and size of a single grain as well. This cannot be easily achieved using standard random variables in $\mathbb{R}$. Although certain properties of this grain of sand might be distributed like a random variable in $\mathbb{R}$, they do not give a complete and general description of all possible randomly shaped objects.

1.2.1 Basic definition

A random closed set is a set-valued random variable. These sets are chosen as closed subsets of $\mathbb{R}^d$. The proper definition requires a σ-algebra $\mathcal{F}$ for measurability of the defining mapping $\Xi : (\Omega, \mathcal{A}, \mathbb{P}) \to (\mathbb{F}, \mathcal{F})$, from a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to $\mathbb{F}$, denoting the family of all closed subsets of $\mathbb{R}^d$. The σ-algebra $\mathcal{F}$ is taken as the smallest σ-algebra generated by the Fell topology on $\mathbb{F}$, see [CSKM13]. This means $\mathcal{F}$ is the smallest σ-algebra that contains all the *hitting sets* $\mathbb{F}_K$,

$$\mathbb{F}_K = \{F \in \mathbb{F} : F \cap K \neq \emptyset\}, \text{ for } K \in \mathbb{K}$$

with $\mathbb{K}$ denoting the compact subsets of $\mathbb{R}^d$. Then $\mathbb{F}$ is equipped with the above mentioned Fell topology, also called the *topology of closed convergence*, see [SW08]. This topology is generated by the set system

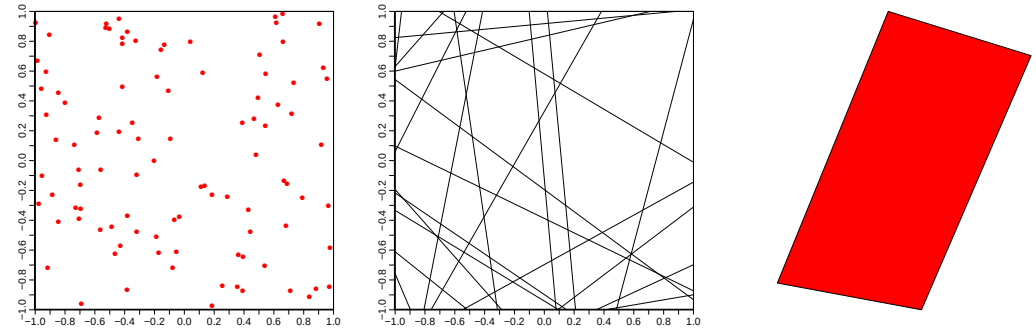
$$\{\mathbb{F}^K : K \in \mathbb{K}\} \cup \{\mathbb{F}_O : O \in \mathbb{O}\},$$

with the system of open subsets $\mathbb{O}$ in $\mathbb{R}^d$. The hitting set system $\mathbb{F}_O$ is generated analogously to $\mathbb{F}_K$ (see above) while $\mathbb{F}^K = \{F \in \mathbb{F} : F \cap K = \emptyset\}$ denotes the missing set system.

1.2.2 Distribution function on $\mathbb{F}$

If Ξ is an $\mathbb{F}$-valued random variable, its distribution $\mathbb{P}_\Xi$ is the image measure of $\mathbb{P}$ under Ξ. The equivalent to a distribution function of a random closed set Ξ is defined in terms of the *capacity functional* $T_\Xi(K)$ via

$$\mathbb{P}_\Xi(\mathbb{F}_K) = T_\Xi(K) = \Pr(\Xi \cap K \neq \emptyset)$$



(a) (b) (c)

Figure 1.1: Realisations of (a): Random point process. (b): Random line process. (c): Random polygon. The plots were generated using R, [R D08]

for a compact set $K \subset \mathbb{R}^d$. This capacity functional is a *Choquet capacity*, this means it has the following properties, see for exampe [OS09, SW08].

1. $0 \leq T_\Xi \leq 1, T_\Xi(\emptyset) = 0$
2. for compact sets $K_i, K \in \mathbb{K}$ with $K_{i+1} \subset K_i, \quad \bigcap_i K_i = K$, we have $T_\Xi(K_i) \to T_\Xi(K)$ (continuity from the right)
3. for compact sets $K, K_0, \ldots, K_n \in \mathbb{K}, n \in \mathbb{N}$ we define

$$\begin{aligned} S_0(K) &:= 1 - T_\Xi(K) \\ S_n(K_0; K_1, \ldots, K_n) &:= S_{n-1}(K_0; K_1, \ldots K_{n-1}) \\ &\quad - S_{n-1}(K_0 \cup K_n; K_1, \ldots, K_{n-1}). \end{aligned}$$

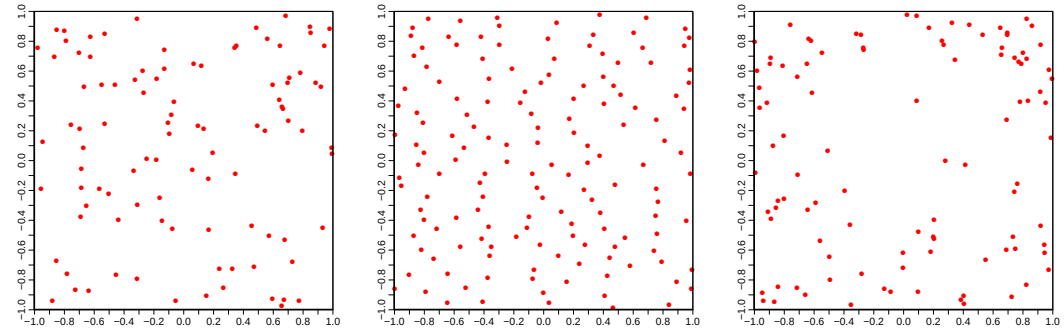
   Then it holds true that $S_n(K_0; K_1, \ldots, K_n) \geq 0$ (monotonicity)

These three properties ensure that $T_\Xi(K)$ behaves as we would expect it for a distribution function.

Additionally, the capacity functional T_Ξ characterizes the random closed set in distribution: For two random closed sets Ξ_1 and Ξ_2 with $T_{\Xi_1} = T_{\Xi_2}$, it follows $\Xi_1 \stackrel{d}{=} \Xi_2$, see [SW08], Theorem 2.1.3.

1.2.3 Examples

One example for the application of random closed sets, the stochastic modeling of the shape of sand grains, was introduced already as motivation for random closed sets. However, there is a large variety of random closed sets. Typical examples are random points, random lines and random polygons. Examples are depicted in Figure 1.1. Among these, random points form a special case: They are not only closed sets in $\mathbb{R}^d$, but they can also be interpreted in a measure-theoretic sense as *random counting measures*, which has many advantages, see Section 1.3. Therefore, we will introduce them again in the following.



(a) (b) (c)

Figure 1.2: (a): Isotropic and stationary point process. (b): Stationary but anisotropic point process, the distances between points in x-direction are larger than in y-direction. (c): Isotropic but not stationary point process (the intensity depends only on the distance from $\mathbf{0}$). The plots were generated using R, [R D08].

1.2.4 Stationarity and isotropy

A random closed set Ξ is *stationary* if $\Xi + \mathbf{x} \stackrel{d}{=} \Xi$ for any $\mathbf{x} \in \mathbb{R}^d$, which means, if its capacity functional (and therefore its distribution) is translation-invariant. It is *isotropic* if $\theta\Xi \stackrel{d}{=} \Xi$ for any rotation $\theta \in SO_d$, which means, if its capacity functional (and therefore its distribution) is invariant under rotation, see Figure 1.2 for examples. When Ξ is stationary and isotropic, it is called *homogeneous*.

1.3 Point processes

Point processes are the fundamental ingredient of stochastic geometry and the respective models. They allow us to generate random locations in space and time that can serve as model, for example for locations of trees in a forest, arrival times of random events, scattered positrons in PET imaging and many more, see [MW04], Chapter 1.

In order to be able to use these models it is necessary to first understand what a point process actually is, in a stochastic sense. A typical viewpoint is to assume point processes as a special form of random measures, namely *random counting measures*, see [SW08], Chapter 3.1. Based on that chapter, we will give a short introduction to point processes in the following.

1.3.1 Basic definition

In order to properly define point processes, we will assume that we have a locally compact space E with a countable base, which is equipped with a Borel σ-algrebra, $\mathcal{B}(E)$. An easy example of such a locally compact space is the Euclidean space $\mathbb{R}^d$, see [Rud87], page 42.

We denote the set of locally finite Borel measures on E by $M = M(E)$. A measure μ is *locally finite* if $\mu(B) < \infty$ for bounded sets $B \in \mathcal{B}(E)$. Then, we set $N = \{\mu \in M : \mu(B) \in \mathbb{N} \cup \{0, \infty\} \text{ for } B \in \mathcal{B}\}$. Then N is the set of locally finite *counting* measures. Looking at the evaluation maps defined by $f_B : \mu \mapsto \mu(B), B \in \mathcal{B}$, we define $\mathcal{M}$ as the smallest σ-algebra such that the f_B are measurable for $B \in \mathcal{B}$. For counting measures N, we denote the trace of this σ-algebra on N by $\mathcal{N} = \{A \cap N : A \in \mathcal{M}\}$.

Based on the σ-algebras $\mathcal{M}$ and $\mathcal{N}$, we can now define *random measures* and *point processes*: For a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, a measurable mapping $\Psi : (\Omega, \mathcal{A}, \mathbb{P}) \to (M, \mathcal{M})$ is called a random measure, and $\Phi : (\Omega, \mathcal{A}, \mathbb{P}) \to (N, \mathcal{N})$ is called a point process. This definition is analogous to the standard definition of a random variable, with the difference that the random element is now measure-valued.

The reason why the Φ are called point processes is that for simple counting measures (that is, measures with $\mu(\{\mathbf{x}\}) \leq 1$ for $\mathbf{x} \in E$), the measure is usually identified with its support, which consists of single points $\mathbf{x}$ for which $\mu(\{\mathbf{x}\}) = 1$.

Based on this, $\mathbb{P}_\Phi$ is the *distribution* of the random point process (analogous for random measures), and $\Lambda(B) = \mathbb{E}\Phi(B), B \in \mathcal{B}$ is called the *intensity measure*. For a simple point process $\Lambda(B)$ would denote the mean number of points in B.

1.3.2 Isotropy and stationarity

As in the case for general RACS, a point process is called *stationary* if its distribution does not change under translations. Similarly, it is called *isotropic* if its distribution does not change under rotations. If a point process is both isotropic and stationary, the intensity measure is invariant under rigid motions. If the point process is stationary, the intensity measure can be decomposed with respect to the d-dimensional Lebesgue measure ν_d. We obtain $\Lambda(B) = \lambda\nu_d(B)$ for $\lambda \in \mathbb{R}$ denoting the mean number of points within a unit cube.

There are various ways to analyse point patterns. From *Choquet's theorem* we know that it is sufficient to know the *void probabilities* $\Pr(\Phi \cap B = \emptyset)$ for $B \in \mathcal{B}$ to completely describe the point process Φ, see [CSKM13]. However, this approach is not very practical for everyday applications. Instead, we aim to gain information on point processes using suitable statistics. If we assume that we observe the stationary random point process Φ on a bounded compact window $W \subset \mathbb{R}^d$, then (see [CSKM13], Chapter 4.7), an estimate of the intensity λ is

$$\hat{\lambda} = \frac{\Phi(W)}{\nu_d(W)}.$$

1.3.3 Poisson point process

Based on the concepts introduced above, we can define the *Poisson point process (ppp)* Φ in the following way:

1. Φ is a simple point process.

2. For $B \in \mathcal{B}$ with $\Lambda(B) = \mathbb{E}(\Phi(B)) < \infty$, $\Phi(B)$ is Poisson-distributed with parameter $\Lambda(B)$, which means $\mathbb{P}(\Phi(B) = k) = \frac{\Lambda(B)^k}{k!} \exp(-\Lambda(B))$, with $k \in \mathbb{N}$.

3. For pairwise disjoint sets $B_1, \ldots, B_n \in \mathcal{B}$, $\Phi(B_1), \ldots, \Phi(B_n)$ are independent. In particular, the points of the point process are independent.

The Poisson point process is often used for stochastic models, due to its flexibility and good theoretical properties. It is also generally quite easy to generate and often the starting point for more sophisticated models. For example, using dependent thinning on a Poisson point process, we can obtain a Matern hardcore point process.

1.3.4 Marked point process

The idea of the *marked point process* is to generalize the point process to $\mathbb{R}^d \times \mathbb{W}$ for a locally compact space $\mathbb{W}$ with countable basis, see [OS09]. The elements of $\mathbb{W}$ are then called *marks*. For example, when analysing the distribution of trees in a forest, one might be interested in their species. Then, $\mathbb{W}$ is a finite set and the point process under investigation consists of points $(\mathbf{x}, s)$, where $\mathbf{x}$ denotes the location in space and s denotes the tree's species.

Formally, a marked point process is defined as follows:

We call the mark space $\mathbb{W}$ and its corresponding σ-algebra $\mathcal{W}$, see [CSKM13], Chapter 7.1.4. Then, a *marked point process* Φ is a point process on $E = \mathbb{R}^d \times \mathbb{W}$, satisfying $\Lambda(K \times \mathbb{W}) < \infty$ for compact $K \in \mathbb{K}$, see [SW08].

We define translations of the marked point process by translations of the underlying *unmarked process*, defined by the projection $(\mathbf{x}, m) \mapsto \mathbf{x}$. This means, when we translate a marked point process $\Phi = \{(\mathbf{x}_1, m_1), \ldots\}$ by $\mathbf{y}$, the translated process $\Phi_\mathbf{y}$ is given by $\Phi_\mathbf{y} = \{(\mathbf{x}_1 + \mathbf{y}, m_1), \ldots\}$. For the rotation around the origin we proceed analogously.

A popular example for marked point processes are the germ-grain model and various kinds of hardcore packings, which will be presented in the following. In these cases, $\mathbb{W} \subset \mathbb{K}$ comprises compact sets.

1.3.5 Particle process and germ-grain models

In a *germ-grain model* we use random non-empty compact sets $\{\Xi_i\}_{i \in \mathbb{N}}$, taken from the space of non-empty compact sets denoted as $\mathbb{K}'$, with centres $c(\Xi_i) = \mathbf{0}$ for $i \in \mathbb{N}$, as marks for a point process $\Phi = \{(\mathbf{x}_i)\}_{i \in \mathbb{N}}$, see [CSKM13], Chapter 6.5 for an overview. There are various possible centre function $c : \mathbb{K}' \to \mathbb{R}^d$, for example the *centre of mass* or the *circumcentre* (that is the centre of the smallest ball containing Ξ_i), see [SW08]. In the following, the space of non-empty compact sets with centre $\mathbf{0}$ will be denoted by $\mathbb{K}'_\mathbf{0}$.

We denote the resulting marked point process by $\Psi = \{(\mathbf{x}_i, \Xi_i)\}_{i\in\mathbb{N}}$. Then the point process Φ, which is the unmarked version of Ψ, constitutes the *germs*, while $\{\Xi_i\}_{i\in\mathbb{N}}$ are the *grains*.

The *germ-grain model* is then the union Ξ over all grains that are added to the germs:

$$\Xi = \bigcup_{i\in\mathbb{N}} (\Xi_i + \mathbf{x}_i)$$

The $(\Xi_i + \mathbf{x}_i)$ constitute a point process, called a *particle process*, in $\mathcal{F}' = \mathcal{F} \setminus \emptyset$ with intensity measure Θ and $\Theta(\mathcal{F}' \setminus \mathbb{K}') = 0$, see [SW08], Chapter 4. Following that chapter, we get the following result: When the particle process $\{(\Xi_i + \mathbf{x}_i)\}_{i\in\mathbb{N}}$ is stationary, Θ can be decomposed as

$$\Theta = \lambda f(\nu_d \otimes \mathbb{Q})$$

with the homeomorphism

$$\begin{aligned} f : \mathbb{R}^d \times \mathbb{K}'_{\mathbf{0}} &\to \mathbb{K}' \\ (\mathbf{x}_i, \Xi_i) &\mapsto \Xi_i + \mathbf{x}_i. \end{aligned}$$

Then the *typical grain* Ξ_0 is defined as the random closed set in $\mathbb{K}'$ with distribution $\mathbb{Q}$. The grains $\Xi_i, i \in \mathbb{N}$ are independently identically distributed as Ξ_0, and λ is the intensity of the underlying point process $\{(\mathbf{x}_i)\}_{i\in\mathbb{N}}$, see also [CSKM13].

The Boolean model

The *Boolean model* is a special case of the germ-grain model, where the underlying point process Φ is a Poisson point process, and the marks are independently identically distributed. The grains Ξ_i are additionally chosen independently from Φ.

Under the condition that $\mathbb{E}(\nu_d(\Xi_0 \oplus K)) < \infty, K \in \mathbb{K}$, we denote the resulting marked point process by $\Psi = \{(\mathbf{x}_i, \Xi_i)\}_{i\in\mathbb{N}}$, see [CSKM13], Chapter 3 for an overview. As stated there, the condition $\mathbb{E}(\nu_d(\Xi_0 \oplus K)) < \infty$ is necessary so that the union

$$\Xi = \bigcup_{i\in\mathbb{N}} (\Xi_i + \mathbf{x}_i)$$

is closed as well.

This model is very well-known due to the many theoretical results on its properties, which are rooted in the properties of the Poisson point process. Its Choquet capacity is given by

$$T_\Xi(K) = 1 - \exp(-\lambda \mathbb{E}(\nu_d(\check{\Xi}_0 \oplus K)))$$

for stationary Φ and $\check{A} = \{-\mathbf{x} | \mathbf{x} \in A\}$. Similarly, the *volume covering fraction*, that is the mean fraction of volume covered by Ξ is given by

$$p = 1 - \exp(-\lambda \mathbb{E}(\nu_d(\Xi_0))).$$

Hardcore packings

Hardcore packings are also germ-grain models which have non-empty compact sets $\{\Xi_i\}_{i\in\mathbb{N}}$ as marks. However, the underlying point process $\Phi = \{(\mathbf{x}_i)\}_{i\in\mathbb{N}}$ is generated in a way that the compact sets $(\Xi_i + \mathbf{x}_i)$ do not intersect.

There are many different algorithms to achieve this, most notable are the *global rearrangement* algorithms and the *sequential adsorption* algorithms: Global rearrangement algorithms start with configurations that allow the intersection of the particles, and globally move the points in Φ until no more overlap occurs. The most famous of these global rearrangement algorithms is the so-called *force-biased algorithm* for balls and spheroids, see [BBS02, BS06]. Sequential adsorption algorithms start with one marked point $(\mathbf{x}_1, \Xi_1)$ and accept additional particles $(\mathbf{x}_i, \Xi_i)$ only when they do not overlap with previously accepted points $(\mathbf{x}_j, \Xi_j)_{j<i}$. Examples are *random sequential adsorption* and sedimentation algorithms, see for example [CSKM13], Chapter 6.5.3.

1.3.6 Cox processes

In real-life applications the stationary and isotropic Poisson point process is rarely a sufficient model: Often, the intensity λ of a Poisson point process is not a constant, but rather a function of the location $\mathbf{x}$. This leads to a local intensity function $\lambda(\mathbf{x}) = \lim_{|\mathrm{d}\mathbf{x}|\to 0} \frac{\mathbb{E}(\Phi(\mathrm{d}\mathbf{x}))}{|\mathrm{d}\mathbf{x}|}$. If the support of the intensity function is also random, we obtain another common generalisation of the Poisson point process, the *doubly stochastic Poisson process*, also called *Cox process*, see [CSKM13], Chapter 5.2.

As the attribute *doubly stochastic* suggests, these are stochastic processes, for which the intensity measure itself is random. The process is a Poisson process conditional on the realisation of the random intensity. The formal definition of this process is presented in the following:

The Cox process Φ with driving random measure Θ, which itself has a distribution Q on $(M, \mathcal{M})$, has the distribution

$$\mathbb{P}_\Phi(Y) = \int \mathbb{P}_{\Psi_\Lambda}(Y) Q(\mathrm{d}\Lambda) \quad \text{for } Y \in \mathcal{N}$$

with a Poisson process Ψ_Λ with intensity measure Λ.

Examples of such a process are the *mixed Poisson process*, which is a Poisson point process with randomised intensity parameter, for example $\lambda \sim Exp(\mu)$; the $\pi(\mathbf{x})$-*thinning of a Poisson process* Φ_b with driving random measure

$$\Psi(B) = \int_B \pi(\mathbf{x}) \Lambda_b(\mathrm{d}\mathbf{x}) \quad \text{for Borel } B \text{ and original intensity } \Lambda_b$$

and a random field π; or the *random-set-generated Cox process*, for example, when the Cox process is lying on the surface or inside a random set. For example, in

Chapter 3 we use a Cox process lying on the facets of random polygons as a model for Al_2Cu precipitations.

All these examples where taken from [CSKM13] as well.

1.3.7 Spherical contact distribution and moment measures

In real-life applications, the points of a point process are often not independent: For example, trees need a minimal amount of free space around them for their roots. This means locations of trees in a forest cannot be independent of each other. There are various approaches to analyze this kind of spatial dependence. When we denote the closed d-dimensional unit ball of radius r centred in $\mathbf{x}$ by $B_r^d(\mathbf{x})$, the *spherical contact distribution* of a point process Φ is defined as

$$H_S(r) = 1 - \mathbb{P}(\Phi(B_r^d(\mathbf{0})) = 0), \quad \text{for } r \geq 0.$$

It gives the distribution function of the distance from $\mathbf{0}$ to the closest point of Φ, see [CSKM13], Chapter 4.1.7. When Φ is stationary, the spherical contact distribution is even the distribution from an arbitrary test point $\mathbf{x} \in \mathbb{R}^d$ to its nearest neighbour in Φ, since by stationarity, moving $\mathbf{0}$ to that point $\mathbf{x}$ does not change the distribution of the point process.

The spherical contact distribution can be estimated using

$$\widehat{H}_S(r) = \frac{\nu_d(W_{\ominus r} \cap \bigcup_{\mathbf{x}\in\Phi} B_r^d(\mathbf{x}))}{\nu_d(W_{\ominus r})}.$$

Here, we investigate only the points lying in the smaller window $W_{\ominus r} = W \ominus B_r^d(\mathbf{0})$. For points lying closer to the edge of W, including them into the estimation would introduce a bias, since their actual closest point might lie outside the observation window. If we instead assume their closest point to lie in the window, the contact distribution would get overestimated.

1.4 Hausdorff measure

In the analysis of random closed sets, not only the full-dimensional feature given by the volume is interesting, but also lower dimensional features such as surface area or curvature. For this reason, many formulas for Random Closed Sets are based on the notion of the *Hausdorff measure*. This measure is used to determine the k-dimensional volume of a k-dimensional set A embedded in the d-dimensional $\mathbb{R}^d$ (meaning $k < d$). In the following, we will use the the definitions proposed in [Ved98], Chapter 2:

Assuming ω_k as the k-dimensional Lebesgue measure of the unit ball in $\mathbb{R}^k$, we define the *sphere measure* by

$$\mathcal{H}_d^k(A, \epsilon) = \inf\left\{ \sum_j \omega_k \left(\frac{\operatorname{diam} A_j}{2} \right)^k : A \subset \bigcup_j A_j, \operatorname{diam} A_j \leq \epsilon, \forall j \right\}$$

with $A \subset \mathbb{R}^d$, $\epsilon > 0$ and $\operatorname{diam} A_j = \sup\{\|\mathbf{x} - \mathbf{x}'\| : \mathbf{x} \in A_j, \mathbf{x}' \in A_j\}$. This means we cover A by sets of diameter less than ϵ and take the infimum over the sums of the ball volumes as measure for A. The *Hausdorff measure* $\mathcal{H}_d^k$ is the limit of the sphere measure when $\epsilon \to 0$:

$$\mathcal{H}_d^k(A) = \lim_{\epsilon \to 0} \mathcal{H}_d^k(A, \epsilon)$$

The Hausdorff measure has the nice property that if A is a subset of a k-dimensional affine subspace of $\mathbb{R}^d$, the Hausdorff measure is equal to the k-dimensional Lebesgue measure on this affine subspace, see also [Ved98]. An affine subspaces L of $\mathbb{R}^d$ is a space that contains the whole line through $\mathbf{x}$ and $\mathbf{y}$ if $\mathbf{x}, \mathbf{y} \in L$.

1.5 Measures on convex sets

Often the random closed sets used for modeling particle structures are not only closed, but also *convex* or *polyconvex*: A set $K \subset \mathbb{R}^d$ is called convex if for any $\mathbf{x}$, $\mathbf{y}$ in K it follows that $c\mathbf{x} + (1-c)\mathbf{y} \in K$ as well, with $0 \leq c \leq 1$, see [CSKM13], Chapter 1.6. We denote the space of compact convex sets by $\mathcal{K}$. Polyconvex sets are defined as the finite union of convex sets.

The *convex hull* of a set X, denoted by $\operatorname{conv}(X)$, is the smallest convex set Y, so that $X \subset Y$. For convex sets X it obviously follows that $X = \operatorname{conv}(X)$. Typical examples of convex sets are the unit ball $B_1^d(\mathbf{0})$, polytopes and affine linear subspaces of $\mathbb{R}^d$, for example the flat $\left\{\mathbf{x} \in \mathbb{R}^3 : x_3 = 1\right\}$ is an affine linear subspace of $\mathbb{R}^3$, when x_1, x_2, x_3 denote the coordinates of a vector $\mathbf{x} \in \mathbb{R}^3$. The compact convex subsets of $\mathbb{R}^d$ are generally called *convex bodies*.

Polytopes are defined as the convex hull of finitely many points $\{\mathbf{x}_1, \dots \mathbf{x}_N\}$. A polytope P can be represented by a system of M linear inequalities with $M < \infty$ that are fulfilled for $\mathbf{x} \in P$, which means that it is the intersection of the closed halfspaces $\{H_i^+\}_{i=1,\dots,M}$ that fulfill these equations, see [SW08], Chapter 10.1:

$$P = \bigcap_{i=1}^{M} H_i^+$$

The *faces* of the polytope P are the intersections of P with its supporting hyperplanes, see [SW08], Chapter 10.1. The *dimension* of the face is the dimension of the isomorphic Euclidean space. For example, for a three-dimensional polytope $P = \operatorname{conv}(\{\mathbf{x}_1, \dots, \mathbf{x}_N\})$ with minimal N, we have the faces:

$\emptyset$ of dimension -1; $\mathbf{x}_1, \dots, \mathbf{x}_N$, the *vertices* (of dimension 0); $l_1, \dots, l_m$, the *edges* (of dimension 1); $f_1, \dots, f_n$, the *facets* (of dimension 2) and P, the polytope itself. In the following, the sets of faces of dimension k are denoted by $\mathcal{F}^k(P)$ for a polytope P.

1.5.1 Intrinsic volumes

Following [CSKM13], Chapter 1.6, a *convex body functional* $h(K), K \in \mathcal{K}'$ is a function from the set of convex bodies to the real numbers $\mathbb{R}$. Examples are the length $l(K)$ if K is of dimension 1, the volume $V(K)$ and the surface area $S(K)$ if K is of dimension 3, etc. These functionals are $\mathcal{K}$-additive, monotone, continuous and invariant under rigid motion, see [SW08], Chapter 14.

This means that they additionally have the property that

$$\begin{aligned} h(K_1 \cup K_2) &= h(K_1) + h(K_2) - h(K_1 \cap K_2) \text{ ($\mathcal{K}$-additivity for } K_1, K_2 \in \mathcal{K}) \\ h(K_1) &\leq h(K_2) \text{ if } K_1 \subset K_2 \text{ (monotonicity)} \\ h(K_n) &\to h(K) \text{ for a sequence } K_n \in \mathcal{K} \text{ with } K_n \to K \in \mathcal{K} \\ &\quad \text{w.r.t. the Hausdorff distance (continuity)} \\ h(\theta K + \mathbf{x}) &= h(K) \text{ (invariance under rotation by } \theta \text{ and translation by } \mathbf{x}) \end{aligned}$$

These properties make them useful measures to characterize convex sets. In the following, we will introduce the *intrinsic volumes*, following [SW08], Chapter 5, as the functionals that fulfil these properties.

The *intrinsic volumes* are defined in terms of the *Steiner formula*. If $V_d = \nu_d$ is the d-dimensional volume, then the volume of $K \oplus B_r^d(\mathbf{0})$ is a polynomial in r and

$$V_d(K \oplus B_r^d(\mathbf{0})) = \sum_{k=0}^{d} r^{d-k} \omega_{d-k} V_k(K), \quad r \geq 0, K \in \mathcal{K}.$$

The set $K \oplus B_r^d(\mathbf{0}) = \{\mathbf{x} \in \mathbb{R}^d : d(\mathbf{x}, K) \leq r\}$ is also called the *parallel set* of K. $V_k,\ k = \{0, \ldots, d\}$, are called the *intrinsic volumes*.

An important result regarding intrinsic volumes is the *Hadwiger characterisation theorem*. It states that any $\mathcal{K}$-additive, monotone, continuous and motion-invariant convex body functional $h(K) \to \mathbb{R}, K \in \mathcal{K}'$ is a linear combination of the intrinsic volumes $V_k(K)$ of K. This means that it does not make sense to consider other functionals with these properties besides the intrinsic volumes when characterizing convex bodies.

Following [OS09], for $d = 3$, the intrinsic volumes are:

$$\begin{aligned} V_3 &= V, \text{ the volume} \\ 2V_2 &= S, \text{ the surface area} \\ \pi V_1 &= M, \text{ the integral of mean curvature} \\ V_0 &= \chi, \text{ the } \textit{Euler characteristic} \end{aligned}$$

with

$$V(X) = \nu_3(X)$$
$$S(X) = \int_{\partial X} \mathrm{d}\mathcal{H}_3^2(s)$$
$$M(X) = \int_{\partial X} H_1(s)\mathrm{d}\mathcal{H}_3^2(s)$$

where $\partial X = \bar{X} \setminus \mathring{X}$ denotes the *boundary* of X and $H_1(\mathbf{x}) = \frac{\kappa_1(\mathbf{x})+\kappa_2(\mathbf{x})}{2}$ the mean curvature at $\mathbf{x} \in \partial X$, with κ_1 and κ_2 denoting the principal curvatures.

The *Euler characteristic* is constant $\chi(K) = 1$ for $K \in \mathcal{K}'$ and integer-valued for other types of sets. For polytopes P with k-dimensional facets $\mathcal{F}^k(P)$, the *Euler-Poincaré formula* holds, see [OS09], Chapter 2:

$$\chi(P) = \sum_{k=0}^{d} (-1)^k \sharp \mathcal{F}^k(P).$$

Additionally, it holds that $\frac{1}{2}V_1 = \bar{b}$, the mean breadth. The breadth $b_\theta(K)$ is the minimal distance between two $(d-1)$-dimensional hyperplanes with normal direction $\theta \in S^{d-1}$ that enclose K between them. The mean breadth is the mean of $b_\theta(K)$ over all directions. In the case of convex polytopes the mean breadth can be calculated very easily:

$$\bar{b}(P) = \frac{1}{4\pi} \sum_{i=1}^{n} l_i \gamma_i$$

with P being a convex polytope, l_i being the length of the i-th edge and γ_i the angle between the surface normals of the polytope facets that form the edge, see [OM00].

Using the *Crofton formula*, we can determine the intrinsic volumes by integrating over lower-dimensional intrinsic volumes on subspaces, see [OS09]. For $L \in \mathcal{L}^k$, the space of k-dimensional linear subspaces of $\mathbb{R}^d$, and L's orthogonal complement ${}^\perp L \in \mathcal{L}^{d-k}$, this gives

$$V_{d+j-k}(X) = \frac{\Gamma\left(\frac{j+1}{2}\right)\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{k+1}{2}\right)\Gamma\left(\frac{d+j-k+1}{2}\right)} \int_{\mathcal{L}^k} \int_{{}^\perp L} V_j(X \cap (L + \mathbf{x}))\nu_{{}^\perp L}(\mathrm{d}\mathbf{x})\mu(\mathrm{d}L),$$

for $k = 1, \ldots, d-1$ and $j \leq k$. Here, $\nu_{{}^\perp L}$ is the Lebesgue measure on ${}^\perp L$ and μ the rotation invariant probability measure on $\mathcal{L}^k$. In Chapter 2, we will introduce how this formula can be used to estimate intrinsic volumes based on digital image data, following [OS09], Chapters 3 and 5.

Intrinsic volume densities

When analysing random sets, it cannot be assumed that they are bounded. For example, a stationary random closed set Ξ is unbounded due to $\Xi + \mathbf{x} \stackrel{d}{=} \Xi$. In these cases, it is sensible to use the intrinsic volume densities. For a stationary, locally polyconvex random closed set Ξ with $\mathbb{E}2^{N(\Xi \cap [0,1]^d)} < \infty$ observed in the d-dimensional non-empty window W with $V_d(W) < \infty$ and
$N(\Xi \cap [0,1]^d) = \min\{m | \Xi \cap [0,1]^d = \bigcup_{i=1,\dots,m} K_i, K_i \in \mathcal{K}'\}$, they are defined as:

$$V_{V,d} = \frac{\mathbb{E}(V_d(\Xi \cap W))}{V_d(W)}$$
$$V_{V,k} = \lim_{\alpha \to \infty} \frac{\mathbb{E}(V_k(\Xi \cap \alpha W))}{V_d(\alpha W)} \quad \text{for } k = 0, \dots, d-1,$$

see [OS09], Chapter 2.3.2. In the case of $d = 3$ we obtain

$$V_{V,3} = V_V, \text{ the volume density}$$
$$2V_{V,2} = S_V, \text{ the surface density}$$
$$\pi V_{V,1} = M_V, \text{ the density of the integral of mean curvature}$$
$$V_{V,0} = \chi_V, \text{ the density of the Euler number.}$$

In the case that Ξ is a stationary germ-grain-model comprised of non-intersecting convex sets, $\chi_V = N_V = \lambda$, the mean number of sets per unit volume is also the *intensity*.

In Chapter 2, we will explain in detail how one can estimate the intrinsic volumes based on digital image data.

1.6 Random Laguerre tessellations

A *random Laguerre tessellation* is a subdivision of $\mathbb{R}^d$, based on a marked point process $\Phi \in \mathbb{R}^d \times \mathbb{W} \subset \mathbb{R}^d \times \mathbb{R}$. In the following we denote the points of Φ by $\{(\mathbf{x}_i, w_i)\}_{i \in \mathbb{N}}$.

Then the cells of the Laguerre tessellation are defined by

$$C(\mathbf{x}_i, w_i) = \bigcap_{(\mathbf{x}_j, w_j) \in \Phi} \{\mathbf{y} \in \mathbb{R}^d : \|\mathbf{y} - \mathbf{x}_i\|^2 - w_i \leq \|\mathbf{y} - \mathbf{x}_j\|^2 - w_j\}$$

It can be shown that every point in $\mathbb{R}^d$ lies almost surely within exactly one cell, see [Lau07]. The special case when all weights are the same leads to the well-known *Voronoi tessellation*. However, unlike the Voronoi tessellation, a Laguerre tessellation does not uniquely define Φ: due to the definition of the cell system, $\Phi' = \{(\mathbf{x}_i, w_i + a)\}_{i \in \mathbb{N}}$ generates exactly the same tessellation for any constant $a \in \mathbb{R}$. Note that this kind of translation is not the same as the translation of a marked point process, which is only defined on the underlying point process. Additionally, some cells $C(\mathbf{x}_i, w_i)$ of the tessellation might be empty or might not

include their generating points $\mathbf{x}_i$. As a remedy, random Laguerre tessellations are often generated based on very specific marked point processes Φ: To ensure that there are no empty cells and each cell $C(\mathbf{x}_i, w_i)$ contains its generating point $\mathbf{x}_i$, random Laguerre tessellations are often generated based on *packed spheres*. Then, the centres of the spheres generate the process $\Phi_0 = (\mathbf{x}_i)_{i\in\mathbb{N}}$ and the radii generate the marks by setting $w_i = r_i^2$, for $i \in \mathbb{N}$. As the packed spheres do not intersect each other by definition, now each cell $C(\mathbf{x}_i, r_i^2)$ contains its generating point, which allows to control the number of cells generated within a window. Additionally, due to the power distance, each cell $C(\mathbf{x}_i, r_i^2)$ contains the ball $B^d_{r_i}(\mathbf{x}_i)$.

1.7 Digital images and their analysis

In the general sense, an image is simply a function $f : X \to Z$ from a closed set $X \subset \mathbb{R}^d$, the *canvas*, to a set Z, the *gray value space*. An example of this would be a classical two-dimensional photograph, where X is a closed set in $\mathbb{R}^2$ and Z is a subset of $\mathbb{R}$. However, it is also possible to obtain more complicated images. For example Z could also be vector- or matrix-valued, which is the case for colour images or EBSD data, see [Hum04].

Nowadays, most imaging systems are computerized and generate digital images. In digital images, the *canvas* is also called a *grid* and consists of discrete points in $\mathbb{R}^d$. Generally, the distances between the grid points are the same along the axes of the image, so X can be considered as subset of

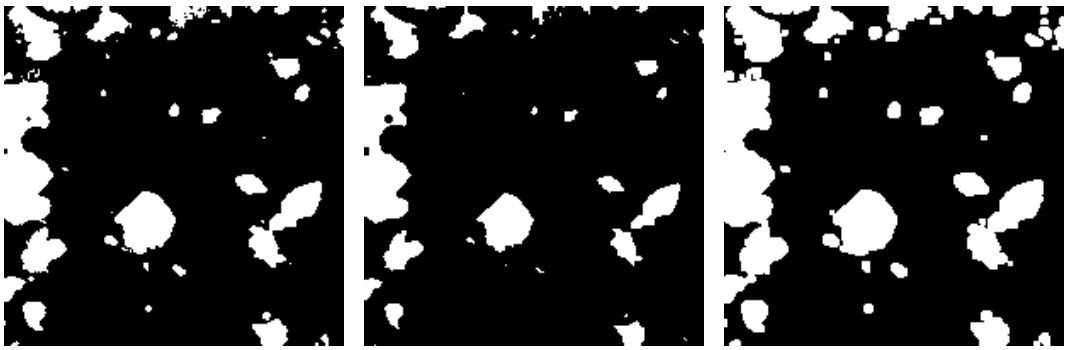
$$\mathbb{L}^d = \left\{ \mathbf{x} \in \mathbb{R}^d | \mathbf{x} = \sum_{i=1}^{d} a_i \mathbf{u}_i \text{ with } a_i \in \mathbb{Z} \right\}$$

with $\mathbf{u}_1, \ldots, \mathbf{u}_d$ forming a basis of $\mathbb{R}^d$, see [OS09], Chapter 3.

Usually, the lattice $\mathbb{L}^d$ is a rectangular lattice, so $\mathbb{L}^d = u_1\mathbb{Z} \times \ldots \times u_d\mathbb{Z} \subset \mathbb{R}^d, u_i \in \mathbb{R}$. The values $u_1, \ldots, u_d$ are denoted the *grid sizes* or *resolution*. A single grid point is called a *pixel* for $d = 2$, or *voxel* in the case of $d = 3$. Similarly to the canvas, the values at these pixels are often discretized as well. Typically, $Z \subset \mathbb{Z}$, or even $Z \subset [0, \ldots, 255]$. If Z only attains the values 0 or 1, the image is a binary image. The canvas of a digital image is a digitised window, $X = \mathbb{L} \cap W$, for a compact $W \subset \mathbb{R}^d$.

In the following, we will denote the space of images by $\mathcal{P}$. The special cases of binary images and gray value images are denoted by $\mathcal{P}_B$ and $\mathcal{P}_G$, respectively. Usually, this means $Z = \{0, \ldots, 255\}$ (8-bit image) or $Z = \{0, \ldots, 65535\}$ (16-bit image). In the case that $Z = \mathbb{R}$, the image is denoted as *float* image. Within each image, we will start the coordinate system at $\mathbf{0}$ and denote pixels by $(i, j) \in \mathbb{N}^2$, voxels by $(i, j, k) \in \mathbb{N}^3$. For an arbitrary image f, its value at (i, j, k) is denoted by f_{ijk}.

Operations on images are mappings from one domain in $\mathcal{P}^n = \mathcal{P} \times \ldots \times \mathcal{P}$ to the range $\mathcal{G} \subset \mathcal{P}$ see [TY09]. Important special cases are the *unary operations*, for which $n = 1$ and the *binary operations* with $n = 2$. For example, with input image f and



(a) (b) (c)

Figure 1.3: (a): Original image f with set $A = \{x|f(x) = 1\}$. (b): $\epsilon_S(A)$ with $S = [-1, 1] \times [-1, 1]$. (c): $\delta_S(A)$ with $S = [-1, 1] \times [-1, 1]$.

output image h, a unary gray value thresholding $\mathcal{O}_t : \mathcal{P}_G \to \mathcal{P}_B$ is given by

$$h_{ij} = \begin{cases} 1 & \text{if } f_{ij} > t \\ 0 & \text{else.} \end{cases}$$

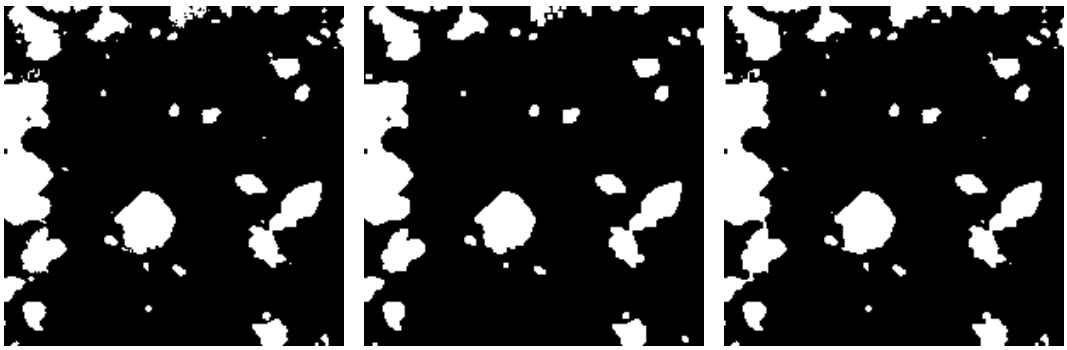
Binary operations are often used as a means to combine the information of two images. For example, a very common binary operation is the *mask*, which can be defined for input images $f \in \mathcal{P}_G$, $g \in \mathcal{P}_B$ and output image $h \in \mathcal{P}_G$ via $\mathcal{O} : \mathcal{P}_G \times \mathcal{P}_B \to \mathcal{P}_G$ by

$$h_{ij} = \begin{cases} f_{ij} & \text{if } g_{ij} = 1 \\ 0 & \text{else.} \end{cases}$$

1.8 Morphological operations

For binary images the values can often be directly associated with the presence or absence of a material in a specimen: For a set $\Xi \subset \mathbb{R}^d$, we can obtain a digital image f by setting the foreground pixels to $f(\Xi \sqcap \mathbb{L}^d \cap W) = 1$ with observation window W. By $\cdot \sqcap \mathbb{L}$ we denote the discretisation based on an adequate adjacency system $\mathbf{F}$, which will be introduced in detail in Chapter 2. Based on the size and structure of Ξ and the lattice $\mathbb{L}^d$, measurements on $\Xi \sqcap \mathbb{L}^d \cap W$ are good approximations for properties of Ξ. In the following, we will assume that the lattice has a sufficient resolution to resolve the relevant structures of Ξ. A thorough introduction in the sampling problem can be found for example in [Lyo04].

Morphological image analysis uses set operations on the voxels of an image with canvas $\mathbb{R}^d$, a d-dimensional Euclidean vector space. All morphological operations are based on the Minkowski addition $A \oplus S$ and the Minkowski subtraction $A \ominus S$ for the sets $S, A \subset \mathbb{R}^d$, see [SW08, Soi99]. Using the notation $\check{S} = \{-s|s \in S\}$, we define the basic operations of morphological image analysis, *erosion* ϵ_S and *dilation* δ_S as:



(a) (b) (c)

Figure 1.4: (a): Original image f with set $A = \{\mathbf{x}|f(\mathbf{x}) = 1\}$. (b): $\gamma_S(A)$ with $S = [-1, 1] \times [-1, 1]$. (c): $\phi_S(A)$ with $S = [-1, 1] \times [-1, 1]$.

$$\epsilon_S(A) = A \ominus \check{S}$$
$$\delta_S(A) = A \oplus \check{S}$$

Applying an erosion to a set A removes all structures that cannot contain the *structuring element* S (see [Soi99], Chapter 4), but it also shrinks the other structures. On the other hand, the dilation of the set A is the same as an erosion of the set A^c, it removes structures in the complement, thereby increasing the size of A and changing its shape. Examples of the effect of dilation and erosion can be seen in Figure 1.3, where the square structuring element $S = [-1, 1] \times [-1, 1]$ has been used and A is the set of all white pixels in the original image.

By combining erosion and dilation, we obtain new morphological operations:

$$\gamma_S = \delta_S \circ \epsilon_S, \text{ the opening}$$
$$\phi_S = \epsilon_S \circ \delta_S, \text{ the closure.}$$

When applying an opening to a set A, first all structures that cannot contain the *structuring element* S are removed, and the remaining structures of the set A are shrunk. The following dilation by the same structuring element counters this effect and recovers the remaining structures of the set A "as much as possible" [Soi99]. This means the opening erases structures smaller than S but changes the rest of the eroded set only slightly. This is useful when small error structures are present in the images, which can be removed. Another application of openings is the *granulometry*: for this, openings are used to filter the structures by size - when applying openings with increasing structuring elements S, combined with measurement regarding how large the retained sets are (for example by counting the number of voxels), we can obtain a size distribution of the structures in the image.

The closure ϕ_S is the complement of the opening, in the sense that $\phi_S(X) = (\gamma_S(X^c))^c$, see [Soi99], with $X^c = \mathbb{R}^d \setminus X$. First, all structures in the image are slightly enlarged using the structuring element S. This process can fill little holes

within or gaps between the structures. The following erosion partially reverses this effect. The structures return roughly to the previous size, but the gap-closing and hole-filling properties of the dilation are retained. This method is especially useful when there is dark noise present or parts of the structures could not be recorded completely. Examples of the effect of opening and closure can be seen in Figure 1.4, where the square structuring element $S = [-1, 1] \times [-1, 1]$ has been used and A is the set of all white pixels in the original image.

Chapter 2

Image analysis

This chapter presents methods to prepare the images of the *AMC17* samples in a way that allows for a statistical analysis of these structures. Parts of it are submitted to appear in [LSB$^+$ed]. In the context of stochastic geometry, we assume that the SiC particles, the Al grains and the Al_2Cu precipitations are stationary random closed sets generated by germ-grain models or random tessellations with *typical grains* or typical cells $X_{0,\mathrm{SiC}}$, $X_{0,\mathrm{Al}}$ and $X_{0,\mathrm{Al_2Cu}}$.

As a first step, it is necessary to correctly identify the phases within the images. Once this is obtained, we have to identify the grains (in the stochastic geometry sense) within each phase. Once the grains are identified, the corresponding pixels are given a label $n_i \in \mathbb{N}$ in a way that the grain X_i can be analysed based on the n_i-set, with $f(X_i \sqcap \mathbb{L} \cap W) = n_i$. Each phase X_{SiC}, X_{Al} and $X_{\mathrm{Al_2Cu}}$ consists of a union of grains via

$$X_\cdot = \bigcup_{\mathbf{x}_{i,\cdot} \in \Phi_\cdot} (X_{i,\cdot} + \mathbf{x}_{i,\cdot})$$

for a point process $\Phi_\cdot \in \{\Phi_{\mathrm{SiC}}, \Phi_{\mathrm{Al}}, \Phi_{\mathrm{Al_2Cu}}\}$.

Given that the grains $X_{i,\mathrm{SiC}}$, $X_{i,\mathrm{Al}}$ and $X_{i,\mathrm{Al_2Cu}}$ forming the phases are independently and identically distributed as the typical grains $X_{0,\mathrm{SiC}}$, $X_{0,\mathrm{Al}}$ and $X_{0,\mathrm{Al_2Cu}}$, we estimate the distributions $\hat{\mathbb{Q}}_{\mathrm{SiC}}$, $\hat{\mathbb{Q}}_{\mathrm{Al}}$, $\hat{\mathbb{Q}}_{\mathrm{Al_2Cu}}$ of $X_{0,\mathrm{SiC}}$, $X_{0,\mathrm{Al}}$ and $X_{0,\mathrm{Al_2Cu}}$ by statistics on the $X_{i,\mathrm{SiC}}$, $X_{i,\mathrm{Al}}$ and $X_{i,\mathrm{Al_2Cu}}$ within each phase.

The separation of the different phases can be obtained by grey value clustering or binarisation. The separation of the grains within each phase can be achieved by using their mostly convex shapes or other local properties. For example, the Al grains can be separated based on the assumption that Al_2Cu precipitations occur on the grain boundaries, while the SiC grains can be identified by their mostly convex shapes.

In the following, we will first introduce the specificities of our image data and some simple filters to account for noise within the images. Then, we will introduce various binarisation methods which can be used to separate the phases from each other. As a next step, we will present a method to identify the grains within a phase. Then,

we will show how to estimate the intrinsic volumes based on digital image data. Finally, we will show how these methods can be used to segment the phases and grains within our data sets.

The methods presented in this chapter are well-known and taken from [TY09, OS09]. This chapter will only give a very short introduction to the methods used. For a more in-depth background we refer to the cited books. All images processing steps presented in this chapter and all figures were generated using MAVI, [MAV05], unless stated otherwise.

2.1 Input Images

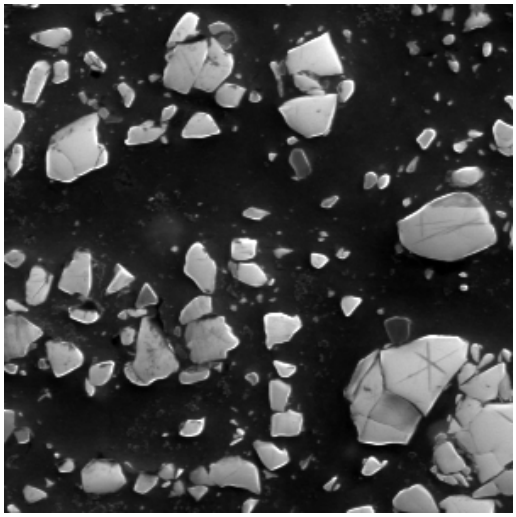
As introduced in the introduction, the data on *AMC17* are available in the form of 2D SEM images and 3D FIB-SEM images. Several examples of data are displayed in Figure 2.1.

The SEM samples were prepared and recorded by various people over the length of five years. This means the available data are very heterogeneous in terms of lighting, contrast and quality of the preparation. The largest difficulties in processing the SEM images are posed by errors in the polishing of the surface: In preparation of the SEM imaging, the surface area of the sample has to be polished manually. Due to the higher hardness of the SiC particles compared to the Al, the Al is abraded faster. This results in SiC particles partially sticking out of the surface or even falling out of the matrix. In the images, this results in some very bright edges on SiC particles or scratches on the surface. Apart from these errors, it is still possible to obtain images with even illumination of the SiC particles.

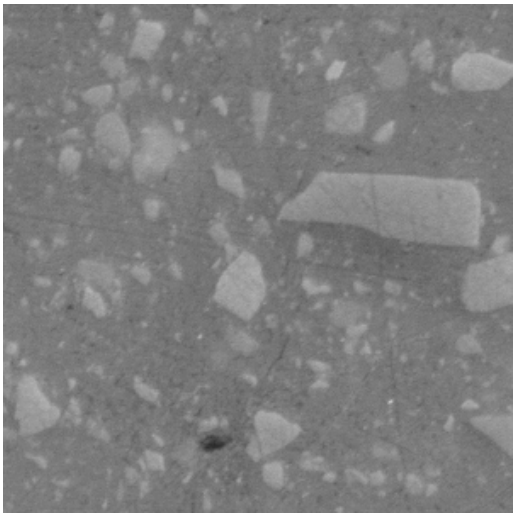
In contrast, the FIB-SEM images were all prepared and recorded by one person over the course of two years. Therefore, the data are very homogeneous in their quality. In FIB-SEM imaging, the polishing is obtained by a focused ion beam, which is a scratch-free procedure. However, the particles are often not evenly illuminated. Instead, the images contain certain typical FIB-SEM artefacts, namely the curtaining effect and bright areas which stem from the accretion of abraded material. Even when these effects have been corrected by a decurtaining algorithm, the resulting images still have an uneven illumination and lack sharpness. There are not only SiC particles and matrix visible in these data sets, but also Al grains and bright Al_2Cu precipitations on grain boundaries. Due to the uneven illumination it often becomes very difficult to distinguish between Al grains and SiC particles. Both SEM and FIB-SEM images contain a slight amount of noise.

2.2 Filtering

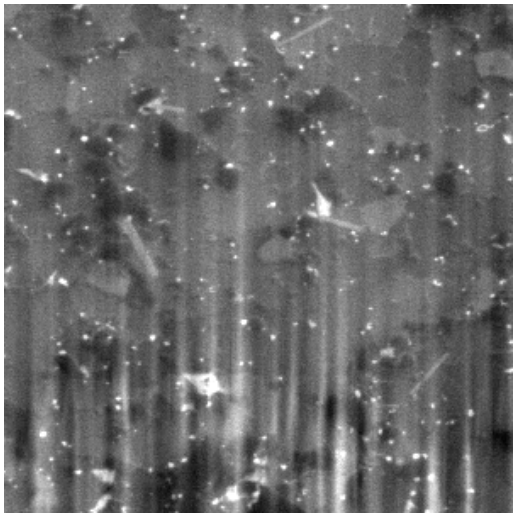
In order to eliminate image artefacts such as noise, which is often present in digital images, as a first step the newly acquired images are *filtered*. A *filtering* is a function from the space of grey value images to the space of grey value images. Usually, only local functions are considered. This means the value at voxel (i, j, k) of the resulting



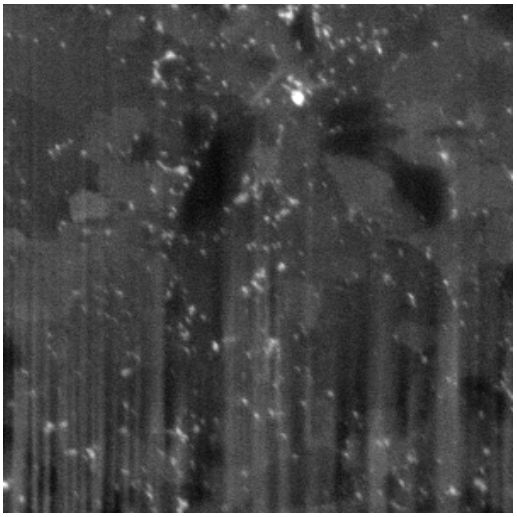
(a) AMC17xe



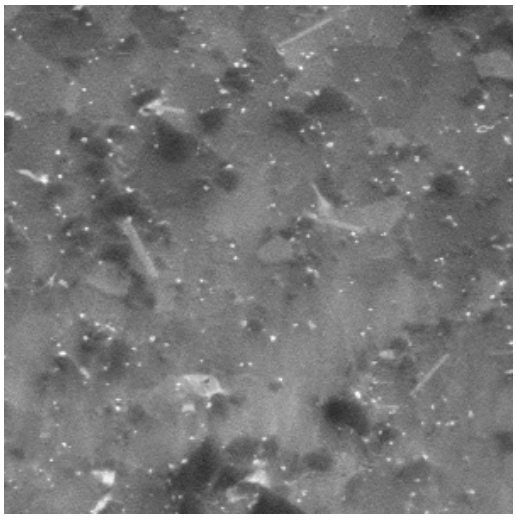
(b) AMC17xfine



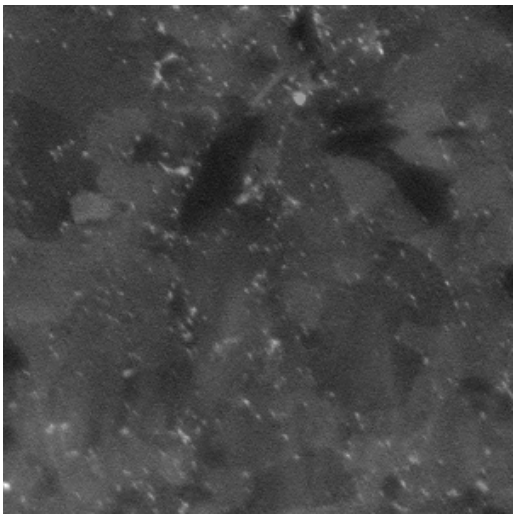
(c) AMC17xxfine



(d) AMC17xfine

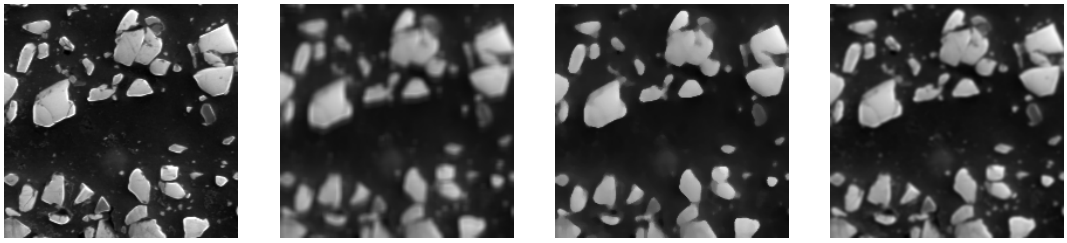


(e) AMC17xxfine



(f) AMC17xfine

Figure 2.1: (a), (b): SEM images. (c), (d): Slices of FIB-SEM images. (e), (f): Slices of FIB-SEM images after removal of the curtaining effect. Displayed are subsections of size 300×300 of the images that were chosen so that the curtaining effect is clearly visible.



(a) Original (b) Mean (c) Median (d) Gauss

Figure 2.2: Smoothing filters, computed on a square mask of size 5×5 pixels

image h depends only on grey values within a small set of voxels lying close to (i, j, k) in the input image f. We will denote this set by $M(i, j, k)$ and call it a *mask*. A typical example for such a mask is $M(i, j, k) = [i - n/2, i + n/2] \times [j - n/2, j + n/2] \times [k - n/2, k + n/2] \cap \mathbb{L}^d$ for even $n \in \mathbb{N}$, see [TY09], Chapter 3. In this example, the *size of the mask* is $n + 1$.

The most common filters are the *mean filter*, the *median filter* and the *Gaussian filter*, which will be presented in the following.

2.2.1 Mean filter

For the input image f and the output image h, the mean filter is defined by:

$$h_{ijk} = \frac{1}{\#M(i,j,k)} \sum_{(p,q,r) \in M(i,j,k)} f_{pqr}$$

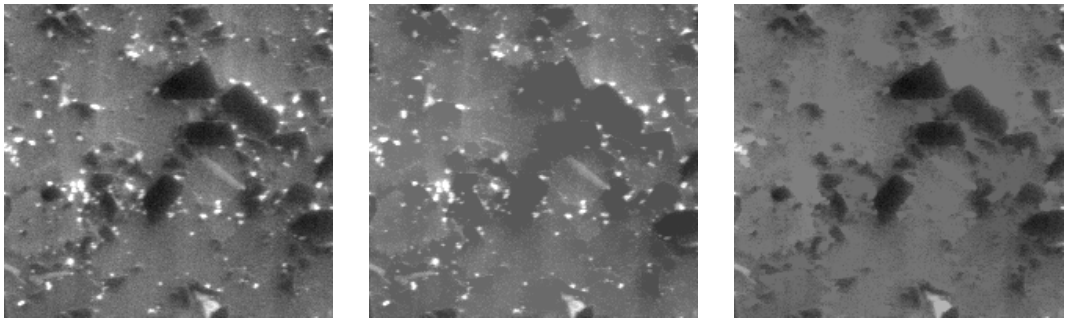
with $M(i, j, k)$ denoting the mask centred at the pixel (i, j, k). Due to the linearity of the filter the computation is very fast. However, this filter is not robust against outliers and smoothens edges within the image.

2.2.2 Median filter

For the input image f and the output image h, the median filter is defined by:

$$h_{ijk} = median\{f_{pqr} | (p, q, r) \in M(i, j, k)\}$$

with $M(i, j, k)$ denoting the mask centred at the pixel (i, j, k). For the computation of the median the voxel values within the mask $M(i, j, k)$ have to be sorted at each voxel. This makes the calculation of the median filter computationally expensive, especially for large masks. The advantage of the median filter are its robustness against outliers and the preservation of edges within the image.



(a) Original (b) Fillhole (c) Cuthill

Figure 2.3: Cuthill and fillhole, computed on an image of size 300×300 pixels.

2.2.3 Gaussian filter

For the input image f and the output image h, the Gaussian filter is defined by:

$$h_{ijk} = \frac{1}{\sum w_{i-p,j-q,k-r}} \sum_{(p,q,r)\in M(i,j,k)} w_{i-p,j-q,k-r} f_{pqr}$$

with $M(i,j,k)$ denoting the mask centred at the pixel (i,j,k) and Gaussian weights $w_{l,m,n}$ taken from a 3-dimensional Gaussian probability density function with mean $\mu = 0$ and standard deviation $\sigma > 0$. In Matlab, version R2016b, the relationship between σ and mask size $N+1$ is implemented as $\sigma = \frac{N}{4}$, for even $N \in \mathbb{N}$, see [mat17]. In order to restrict the weights to the bounded mask, they are usually approximated by binomial coefficients, see [OS09], Chapter 4. The Gaussian filter can be decomposed along the image dimensions. Due to this separability, the filtered image is fast to compute.

Results of all three filters applied to a grey value image are depicted in Figure 2.2.

2.2.4 Cuthill and Fillhole

The *fillhole* is a morphological operation on images, which "fills" all minima in the image that do not lie on the image border, see [Soi99]. This means local minima are eliminated by this filter. For grey value images, the erosion is defined as $\epsilon_S(f_{ijk}) = \min_{(p,q,r)\in S} f_{i-p,j-q,k-r}$ with structuring element S. The *fillhole* is defined for grey-valued images in terms of the *geodesic erosion* $\epsilon^1_g(f_{ijk}) = \max\{\epsilon^1(f_{ijk}), g_{ijk}\}$ for a marker image g of the same grid size and dimensions. Here, ϵ^1 refers to the *elementary erosion*, that is an erosion by a 3 voxel wide structuring element, consisting of its central voxel $\mathbf{0}$ and all adjacent voxels, depending on the chosen neighbourhood, see 2.2.5 for details on neighbourhoods and adjacency.

In order to fill the minima of an image f, we have to choose f as a marker image in

the definition of the fillhole, and construct another image f':

$$f'(i,j,k) = \begin{cases} f_{ijk} & \text{if } (i,j,k) \text{ is on the image border} \\ \max f & \text{else.} \end{cases}$$

The fillhole is then the repeated geodesic erosion of f' with the marker image f until stability, see also [Soi99]. This means in this case, the geodesic erosion is taken as $\epsilon_f^1(f'_{ijk})$. Note that due to the maximisation within the geodesic erosion, repeated geodesic erosions reach stability after a finite number of applications. The *cuthill* is obtained by a fillhole on the complement $f^c = \max(f) - f$ of the image, it is defined as $(\text{fillhole}(f^c))^c$.

Since fillhole and cuthill eliminate local minima and maxima, respectively, they can be used to eliminate noise within images or extract the local extrema. Examples of the application of fillhole and cuthill can be seen in Figure 2.3.

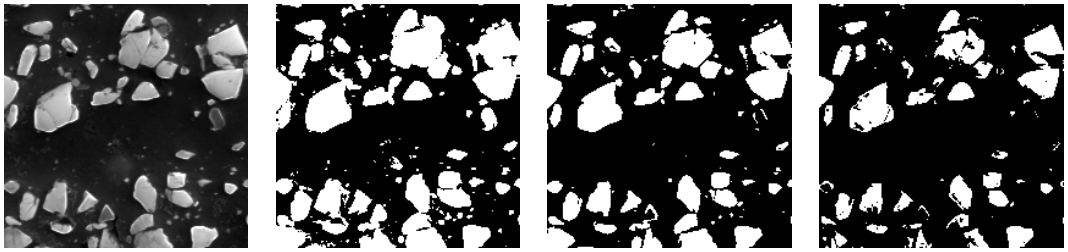
2.2.5 Neighbourhoods and Adjacency

When segmenting particles, we would usually assume that voxels $x = (i,j,k)$ and $x' = (i',j',k')$ with $f(x) = f(x') = 1$ that lie next to each other in the binary images should belong to the same particle. In order to determine which voxels constitute a particle and for many image analysis algorithms, it is necessary to choose the adjacency of the images. In 2-D, these adjacencies are defined by the neighbourhoods of the pixels, see [OS09], Chapter 3.3. Typical neighbourhoods in 2D are the 8-neighbourhood $N_8(i,j) = \{(i+p, j+q), p, q \in \{-1,0,1\}\}$ and the 4-neighbourhood $N_4(i,j) = \{(i+p, j+q),$ either $p = 0$ and $q \in \{-1,0,1\}$ or $q = 0$ and $p =\in \{-1,0,1\}\}$.

For higher-dimensional images, the neighbourhood alone is not sufficient to characterise the topology of voxel sets, see [OS09], Chapter 3.3.1. Therefore, adjacency is usually defined in terms of an *adjacency system* consisting of subsets of the vertices $\mathcal{F}^0(C)$ of the unit cell C, see for example [OS09], Chapter 3.2.1: The unit cell C of a lattice $\mathbb{L}^d$ is formed by the basis of the lattice by $C = [\mathbf{0}, \mathbf{u}_1] \oplus \ldots \oplus [\mathbf{0}, \mathbf{u}_d]$, where $[\mathbf{0}, \mathbf{u}_i]$ is a segment between $\mathbf{0}$ and the *lattice point* $\mathbf{u}_i$. Assuming that $d = 3$, this unit cell has 8 vertices, 12 edges and 6 facets and the topology of a scaled cube. A *local adjacency system* $\mathbf{F}_{loc}$ is generated by a subset of all the possible convex hulls of vertex combinations within C, when it has the following properties:

1. $\emptyset \in \mathbf{F}_{loc}$ and $C \in \mathbf{F}_{loc}$
2. if $F \in \mathbf{F}_{loc}$ then so are its faces $\mathcal{F}^i(F)$
3. for two sets $F_i, F_j \in \mathbf{F}_{loc}$, either the convex hull of their union is in $\mathbf{F}_{loc}$ or $F_i \cap F_j, \overline{F_i \setminus F_j}$ and $\overline{F_j \setminus F_i} \in \mathbf{F}_{loc}$
4. for $F_{i_1}, \ldots F_{i_m} \in \mathbf{F}_{loc}, \bigcup_{j=1}^m F_{i_j} \in \mathbf{F}_{loc}$ if it is convex

The (global) *adjacency system* is then generated by $\mathbf{F} = \bigcup_{x \in \mathbb{L}^d} (\mathbf{F}_{loc} + x)$. Typical adjacencies are the 6-adjacency, which is generated by the union of the faces of



(a) Original (b) $t = 50$ (c) Otsu ($t = 100$) (d) $t = 150$

Figure 2.4: Results of global thresholding at different threshold levels t

the unit cell, and the 26-adjacency, which is generated by the power set of the set containing all vertices of the unit cell.

Based on the adjacency system, the discretisation of a compact set $X \subset \mathbb{R}^d$ is given by the elements of the adjacency system that have all their vertices lying in X:

$$X \sqcap \mathbf{F} = \bigcup \{F \in \mathbf{F} | \mathcal{F}^0(F) \subset X\}.$$

As the connectivity of the discretised set is determined by the vertices and the edges lying in the adjacency system, different adjacency systems might generate differently connected discretisations. In order to have consistency of the Euler number, defined by $\chi(X) = (-1)^{d+1}\chi(\bar{X}^c)$, in the discretised case, it is necessary to choose an adequate adjacency system $\mathbf{F}_c$ on the background $\bar{X}^c \cap \mathbb{L}^d$ for the chosen adjacency system $\mathbf{F}$, see [OS09]. In [OS09] it is shown that adjacency systems fulfilling this criterion for $d = 3$ are given by $(\mathbf{F}_6, \mathbf{F}_{26})$, $(\mathbf{F}_{26}, \mathbf{F}_6)$, $(\mathbf{F}_{14.1}, \mathbf{F}_{14.1})$ and $(\mathbf{F}_{14.2}, \mathbf{F}_{14.2})$. For information on $\mathbf{F}_{14.1}$ and $\mathbf{F}_{14.2}$ we refer to [OS09] as well.

In the following, we will always assume that adequate adjacency systems have been chosen to generate the discretisations $X \sqcap \mathbf{F}$ and $X^c \sqcap \mathbf{F}_c$.

2.3 Segmentation methods

Image segmentation or *grey value segmentation* is essentially a method to cluster the pixels of an image by a certain set of rules. For our data, we want to cluster the pixels according to the phase they represent. However, for other applications, the notion of segmentation could have an entirely different meaning.

In the following, we will present several common methods to achieve a segmentation. It has to be noted that there is no optimal method to segment an image, since the effort and the effectiveness of the methods strongly depend on the image quality.

2.3.1 Thresholding

Thresholding is one of the simpler methods to segment an image. A *global thresholding* assigns each voxel to a cluster solely based on its grey value, while the *local thresholding* also takes the grey values of surrounding voxels into account. In the

case that there are only two cluster centres taken into consideration, the resulting image is a binary image and the segmentation is called a *binarisation*.

For cluster centres $C = \{c_0, \dots, c_n\}$, $n \in \mathbb{N}$, input image f and output image h, a global thresholding function can be defined as $\mathcal{O}_{t_0,\dots,t_{n-1}} : \mathcal{P}_G \to \mathcal{P}_C$ with

$$h_{ijk} = \begin{cases} c_0 & \text{if } f_{ijk} < t_0 \\ c_1 & \text{if } t_0 \leq f_{ijk} < t_1 \\ \vdots & \vdots \\ c_n & \text{if } t_{n-1} \leq f_{ijk} \end{cases}$$

where $\mathcal{P}_C$ refers to images of range C.

A *local thresholding* function can be defined by

$$h_{ijk} = \begin{cases} c_0 & \text{if } f_{ijk} < t_0(f_{M(i,j,k)}) \\ c_1 & \text{if } t_0(f_{M(i,j,k)}) \leq f_{ijk} < t_1(f_{M(i,j,k)}) \\ \vdots & \vdots \\ c_n & \text{if } t_{n-1}(f_{M(i,j,k)}) \leq f_{ijk} \end{cases}$$

with $M(i,j,k)$ denoting a mask as introduced above, $f_{M(i,j,k)}$ the image values in the sub-image defined by the mask $M(i,j,k)$ and threshold functions $t_i : \mathcal{P}_G \to G$. This means the thresholds $t_0, \dots t_{n-1}$ depend on the grey values near (i,j,k). Usually, the cluster centres are identified with their index, so that $c_0 = 0$ and $c_1 = 1$.

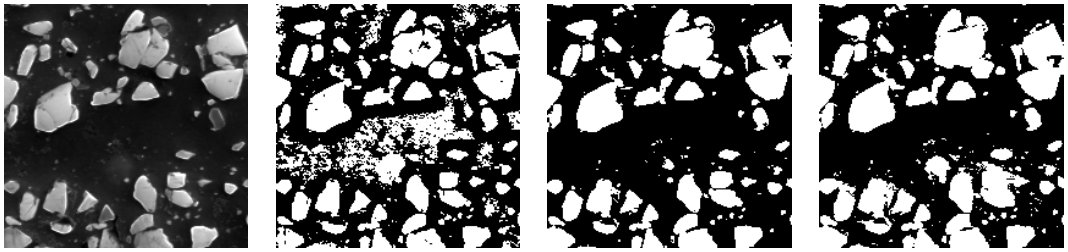
A segmentation can be obtained by applying repeated binarisations. This means, instead of directly clustering the voxels into n clusters, we first identify the set of voxels in the canvas X belonging to c_0, and denote this voxel set by X_{c_0}. As a next step, the remaining voxels $X \setminus X_{c_0}$ are clustered to either belong to the cluster c_1 or not, resulting in the set $X_{c_1} \subset (X \setminus X_{c_0})$, and so on.

When using global binarisation, the necessary threshold t is often obtained using *Otsu's method* [Ots79]: t is set so that the intra-class variance is minimized. In Figure 2.4, the results of global thresholding methods are presented for different threshold levels t. It is obvious from this simple example that for most images global thresholding does not provide good results: low and high thresholds overemphasise grey value fluctuations, and also the seemingly optimal Otsu threshold cannot correctly segment regions with uneven grey values.

A common *local* thresholding method is *Sauvola binarisation*. This method sets the local threshold according to the rule:

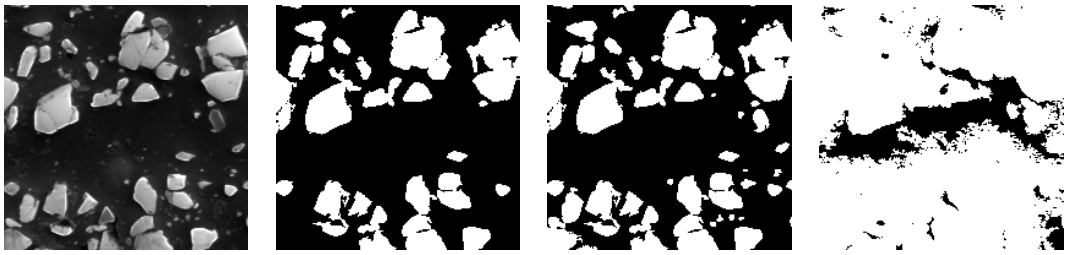
$$t(f_{M(i,j,k)}) = m_{ijk}\left[1 + k\left(\frac{s_{ijk}}{R} - 1\right)\right]$$

(in [SP00], Formula 5). Here, $m_{ijk} = \frac{1}{\sharp M(i,j,k)} \sum_{(p,q,r)\in M(i,j,k)} f_{pqr}$ refers to the mean pixel value within a small mask around the pixel (i,j,k), s_{ijk} refers to the standard deviation within this mask and $R = \frac{\max s}{\min s}$ refers to the *dynamic range* of the standard deviation. In many applications this dynamic range is set to the maximum possible grey value (this means $R = 255$ for 8-bit images or $R = 2^{16} - 1$ for 16-bit images),



(a) Original (b) $k = 0.1, w = 10$ (c) $k = 0.1, w = 50$ (d) $k = 0.5, w = 50$

Figure 2.5: Local thresholds: Sauvola's method applied for different values k, w for square masks



(a) Original (b) $t_l = 50, t_u = 220$ (c) $t_l = 50, t_u = 130$ (d) $t_l = 20, t_u = 190$

Figure 2.6: Local thresholds: Hysteresis binarisation

(Katja Schladitz, personal communication, January 2017). The parameters of this decision are k and the size and shape of the mask $M(i, j, k)$ under consideration. The result of this method is depicted in Figure 2.5 for different values of $k \in \mathbb{R}$ and the side length w of a quadratic window. In comparison with the global threshold method depicted in Figure 2.4, the method by Sauvola is able to capture regions with grey value fluctuations. However, the parameters have to be well-tuned in order to obtain a good binarisation.

2.3.2 Hysteresis Binarisation

Another local thresholding method is the so-called *hysteresis binarisation*. This method uses two different thresholds, the upper threshold t_u and the lower threshold t_l with $t_u > t_l$. Then, for input image f and output image h, in a first step, all pixels (i, j, k) for which f_{ijk} is lying above the upper threshold t_u are assigned to cluster c_0, while the pixels for which f_{ijk} is below the lower threshold t_l are assigned to c_1:

$$h^1_{ijk} = \begin{cases} c_0 & \text{if } f_{ijk} > t_u \\ c_1 & \text{if } f_{ijk} < t_l \\ c_2 & \text{else.} \end{cases}$$

The pixels in cluster c_2 are then recursively reassigned to the clusters c_0 and c_1,

based on whether they lie adjacent to a pixel already lying in cluster c_0:

$$h_{ijk}^{n+1} = \begin{cases} c_0 & \text{if } f_{ijk} \geq t_l \wedge h_{pqr}^n = c_0 \text{ for at least one } (p,q,r) \in N(i,j,k) \\ c_1 & \text{if } f_{ijk} < t_l \\ c_2 & \text{else} \end{cases}$$

until convergence, resulting in an image h_*. Here, $N(i,j,k)$ is the neighbourhood of the voxel (i,j,k) in terms of the chosen adjacency system $\mathbf{F}$. The remaining voxels in cluster c_2, which are not connected to any voxels in $h_*^{-1}(c_0)$, are then assigned to cluster c_1, resulting in the binary image h. This local binarisation often leads to smoother regions than Otsu's or Sauvola's method. Examples for various values of t_l and t_u are presented in Figure 2.6. It is clear from the image that a careful tuning of the parameters is necessary.

2.3.3 Optimising functionals

Another way to obtain a segmentation of images is by minimising certain functions J on the image. An example of this would be the K-means or the fuzzy-c-means functional. By adding a regularisation term it is possible to enforce smooth edges of the clusters. The classical K-means tries to find optimal cluster centres c_l, $l = 1, \ldots, m = K$ that minimize

$$J = \sum_{(i,j,k)\in X} \sum_{l=1}^{m} u_l(i,j,k)|f_{ijk} - c_l|^2$$

with the *label defining matrix* $u_l(i,j,k)$ such that $u_l(i,j,k) = 1$ if $|f_{ijk} - c_l|^2 < |f_{ijk} - c_p|^2$ for $p \neq l$ and the canvas X. The resulting image h is then obtained by

$$h_{ijk} = \sum_{l=1}^{m} \mathbb{1}_{u_l(i,j,k)} c_l$$

or

$$h_{ijk} = \sum_{l=1}^{m} \mathbb{1}_{u_l(i,j,k)} (l-1).$$

For $m = 2$ this is equivalent to Otsu's method, see [LY09]. The k-means clustering has the same problems as Otsu's method, namely that local information of the voxels is not taken into account.

In [SS12], a fuzzy-c-means functional is presented with a regularisation term $\lambda \mathrm{TV}(u)$ that enforces smooth edges of the clusters:

$$J = \sum_{(i,j,k)\in X} \sum_{l=1}^{m} (u_l(i,j,k))^b (f_{ijk} - c_l)^p + \lambda \mathrm{TV}(u)$$

with $b > 1$, $p = 2$ and $\mathrm{TV}(f) = \sum_{ijk} |\nabla f_{ijk}|$, where ∇ denotes the gradient, see [SS12]. Here, $u_l(i,j,k)$ is a *label defining matrix* that assigns each image point

partially to a label c_l, i. e. $\sum_{l=1}^{m} u_l(i,j,k) = 1, u_l \geq 0$ for each voxel (i,j,k). The resulting image h is then obtained by

$$h_{ijk} = \{c_l | l = \arg\max_l u_l(i,j,k)\}.$$

Due to the local information that is used, "good" shapes of the clustered particles are enforced.

2.3.4 Random forest classification

Often, it is very hard to find an automatic method to segment images. For example, if the particles observed within an image vary greatly in their sizes, a regularization might not yield good segmentations of both small and big particles. On the other hand, it could also happen that there are huge grey value fluctuations within the images or between the slices of a three-dimensional image set. In these cases, it is often helpful to use a learning-based method to segment the data that uses example regions selected *manually by the user* to automatically learn rules to separate the different phases.

A program which uses such a segmentation method is *ilastik*, see [SSKH11], which is based on a *random forest classifier*, [Bre01]. The random forest consists of several decision trees that are randomly initialized with subsets of the manually marked pixels and their local properties, such as variance, grey value, derivatives, etc. Based on the informativeness of these features with regards to the labels, each tree learns different "rules" for the labels. By combining these random decision trees, overfitting is suppressed.

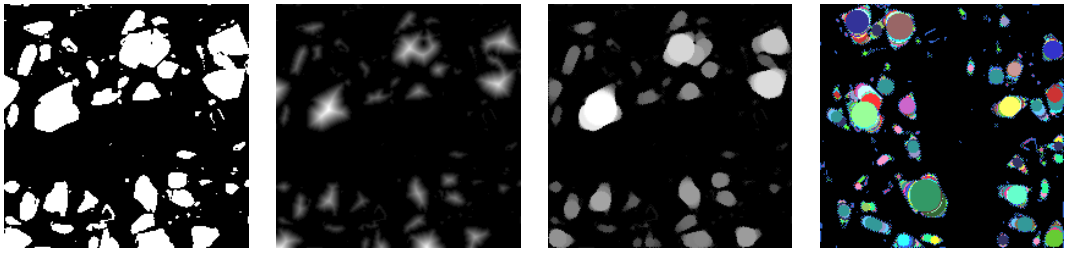
2.4 Particle separation

In terms of stochastic geometry, the *segmentation methods* described so far are used to identify the phases Ξ_{SiC}, Ξ_{Al} and $\Xi_{\mathrm{Al_2Cu}}$ of the material, which themselves can be modeled as germ-grain process, i.e.

$$\begin{aligned}
\Xi_{\mathrm{SiC}} &= \bigcup_{i\in\mathbb{N}} (\Xi_{i,\mathrm{SiC}} + \mathbf{x}_{i,\mathrm{SiC}}) \\
\Xi_{\mathrm{Al_2Cu}} &= \bigcup_{i\in\mathbb{N}} (\Xi_{i,\mathrm{Al_2Cu}} + \mathbf{x}_{i,\mathrm{Al_2Cu}}) \\
\Xi_{\mathrm{Al}} &= \bigcup_{i\in\mathbb{N}} (\Xi_{i,\mathrm{Al}} + \mathbf{x}_{i,\mathrm{Al}})
\end{aligned}$$

for suitable point processes $\{\mathbf{x}_{i,\mathrm{SiC}}\}_i$, $\{\mathbf{x}_{i,\mathrm{Al_2Cu}}\}_i$ and $\{\mathbf{x}_{i,\mathrm{Al}}\}_i$.

In order to estimate the distribution of the typical grains $\Xi_{0,\mathrm{SiC}}$, $\Xi_{0,\mathrm{Al_2Cu}}$ and $\Xi_{0,\mathrm{Al}}$ based on statistics on the observed sets X_{SiC}, X_{Al} and $X_{\mathrm{Al_2Cu}}$, it is therefore necessary to identify the single grains $\{\Xi_{i,\mathrm{SiC}}\}_i$, $\{\Xi_{i,\mathrm{Al_2Cu}}\}_i$ and $\{\Xi_{i,\mathrm{Al}}\}_i$ within the phases. This is obtained by *particle separation*, which is also often called *labelling*. This is done on the segmented images h obtained by clustering.



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

Figure 2.7: The effect of Euclidean distance transform and spherical granulometry on a binary image. (a): input binary image. (b) Euclidean distance transform on the foreground. (c) Spherical granulometry on the foreground. (d) Colored levels of the granulometry image for better visibility

When the particles do not touch each other, they can be identified by labelling the connected components within the segmented image h. The standard method to separate touching objects within a phase is the *watershed algorithm*: Local minima in the image are considered as water sources, and the water level rises according to the grey value gradient at each pixel and the applied adjacency system $\mathbf{F}$. When two water basins touch, they form a *watershed* at the contact. These watersheds are then used as the separating curves within the particle phase to separate said particles. The particle separation is obtained by assigning each minimum a unique label l_n. The labelled image is them $h_{ijk} = l_n$ if (i, j, k) lies within the basin belonging to minimum n.

There are several methods available to generate the local minima needed for the watershed algorithm, which will be presented in the following.

2.4.1 Euclidean distance transform

On a binary image f_{ijk} the *Euclidean distance transform* assigns each pixel of the phase $f^{-1}(1)$ the shortest Euclidean distance to the other phase $f^{-1}(0)$. As our image data is displaying non-intersecting particles that touch each other on a small fraction of their surfaces, this results in local maxima lying at the centres of particles. The transform can be defined as $\mathcal{O} : \mathcal{P}_B \to \mathcal{P}_G$ with input image f and output image h by:

$$h_{ijk} = \begin{cases} \min_{(i',j',k') \text{ st } f_{i'j'k'}=0} ||(i,j,k) - (i',j',k')|| & \text{if } f_{ijk} = 1 \\ 0 & \text{if } f_{ijk} = 0 \end{cases}$$

For example, if f_{ijk} displays a ball of radius r, its centre gets assigned the value r, while the surrounding pixels get assigned lower values, decreasing linearly with their distance from the center. By taking the complement $h^c_{ijk} = \max h - h_{ijk}$, local maxima become the local minima needed for the watershed algorithm. The result of the Euclidean distance transform is depicted in Figure 2.7, (b).

2.4.2 Spherical granulometry

The *spherical granulometry image* of a binary image assigns each foreground voxel (i, j, k) with $f_{ijk} = 1$ the radius r of the largest ball lying completely inside the foreground and covering that voxel. The maxima generated by this transform are flat, while for voxel sets not conforming to a spherical shape, pixels lying at edges get assigned smaller values. This means if f_{ijk} displays a ball of radius r, all circle pixels get assigned the value r. Taking the complement image generates minima which can be used for the watershed algorithm. The effect of the spherical granulometry is depicted in Figure 2.7 (c).

When applying the watershed transform to any of these distance transform images, one has to account for local minima. Often, these local minima were generated by noise or smaller, unimportant structures within the particles and result in an oversegmentation. There are several methods to deal with this:

- Smoothing the distance transform image by one of the filters presented above
- H-Minima transform on the distance transform image
- Manually joining labels after the watershed algorithm
- Preflooded watershed

Using the Euclidean distance transform (EDT) generates many of these lesser local minima generated by small structures or noise. In order to obtain a good labelling, a typical approach is to use a H-Minima transform on top of the complement of the Euclidean distance transform. The spherical granulometry is more robust in that matter, so it might be sufficient to manually correct some oversegmented labels.

2.4.3 H-Minima transform

The *H-Minima transform* removes local minima in an image with a method similar to *fillholes*, see [OS09], Chapter 4. The transform is based on repeated geodesic erosions while using the EDT image f as marker on an image defined by $h_{ijk} = f_{ijk} + d, d \in G$ until stability is reached. This fills all minima of *dynamic* less than d. By the dynamic of a minimum we denote the difference between the value attained at the closest local maximum, here denoted by $f_{max,l}$, and the value attained at the minimum, here denoted by $f_{min,l}$. The affected minima are elevated by d, while the related maxima are elevated by $\max\{d - (f_{max,l} - f_{min,l}), 0\}$.

2.4.4 Preflooded watershed

In the standard watershed algorithm, all basins are filled simultaneously. The *preflooded watershed* eliminates oversegmentation by only admitting watershed basins in the d-dimensional image to be filled when they are above a certain ($(d + 1)$-dimensional) volume threshold t. The basins that do not meet this criterion are

only filled when a neighbouring water basin spills into them, see [OS09], Chapter 4. As mentioned there, the difference between preflooded watershed and H-minima transform lies in the fact that the preflooded watershed is a local operation, while the H-minima transform elevates all minima in the image.

2.5 Estimating the distribution of the typical grain Ξ_0

Based on the labelled images, it is possible to estimate properties such as local Euler number or orientation within the labelled sets and use them as estimates of the intrinsic volumes and orientations of the particles within the sample. The distributions of these properties are then estimates of the related distributions of the typical grain Ξ_0. In the following, we will describe shortly how to arrive at these estimates. The following sections are based on the books on 3D image analysis by [OS09] and [TY09].

2.5.1 Volume V_3

The estimation of the volume of a particle X labelled within the image f by the value c_i is straightforward: Following Cavalieri's principle (see [RSVW14]) we get $\hat{V}_3(X) = \sum \mathbb{1}_{f=c_i} \times V_{vox}$, where $\sum \mathbb{1}_{f=c_i}$ denotes the number of voxels within the label c_i and V_{vox} denotes the physical volume represented by one voxel.

2.5.2 Euler number

Following the *Euler-Poincaré* equality, the Euler number of a set X observed in an image can be estimated based on the faces of the digitized set $X \sqcap \mathbf{F}$:

$$\hat{\chi}(X) = \chi(X \sqcap \mathbf{F}) = \sum_{k=0}^{3} (-1)^k \sharp \mathcal{F}^k(X \sqcap \mathbf{F}),$$

see [OS09] (3.5).

2.5.3 Other intrinsic volumes

There are several methods on how to estimate the other intrinsic volumes based on digital image data:

The naive approach consists in first generating a triangular mesh on the voxel data and then measuring the surface area or mean breadth of this mesh. The problem with this approach is that the generated mesh is not unique and the results would depend on the mesh size.

In order to estimate the surface area and other intrinsic volumes directly from voxel data, one uses the *Crofton formula* instead (see 1.5.1), which relates measurements

of the Euler number in lower dimensions to other intrinsic volumes in higher dimensions. The following considerations are taken from [OS09]: For a polyconvex compact set $X \in \mathbb{R}^d$ and a k-dimensional linear subspace of $\mathbb{R}^d$, denoted by $L \in \mathcal{L}^k$, the space of k-dimensional linear subspaces, and its $(d-k)$-dimensional orthogonal complement $L_\perp$, we have

$$V_{d-k}(X) = \frac{\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{k+1}{2}\right)\Gamma\left(\frac{d-k+1}{2}\right)} \int_{\mathcal{L}^k} \int_{\perp L} \chi(X \cap (L + \mathbf{y}))\nu_{L_\perp}(\mathrm{d}\mathbf{y})\mu(\mathrm{d}L)$$

By $\nu_{L_\perp}$ we denote the Lebesgue measure on $L_\perp$. By μ we denote the rotational invariant measure on $\mathcal{L}^k$. On digitized images, the integration has to be calculated on the discretised set $X \cap (L+\mathbf{y}) \sqcap \mathbf{F}^k$ for discrete $\mathbf{y}$ and an adjacency system $\mathbf{F}^k$ on the section lattice $\mathbb{L}^k$, see [OS09], Chapter 5 and [RSVW14]. We will assume that $d = 3$ and $k = 1, 2$, since the estimation of the volume was already outlined above and the Euler number can be estimated based on the Euler-Poincaré formula. Following the outline of [RSVW14], we replace the integrals by sums in the following way: The directional integral is approximated by using the 13 discrete directions given by the directions of the edges, the face diagonals and the space diagonals of the unit cell of the lattice. The orthogonal complement is replaced by the *translative complement* $\mathbb{L}_T^{3-k}$, which is not necessarily uniquely determined.

The translative integral, here denoted by $p_k(X, L)$, can be estimated in the following:

$$\hat{p}_k(X, L) = \sum_{l=0}^{m_k} \frac{V(C)}{V(C^k)} \chi_0^k(\zeta_l \sqcap \mathbf{F}^k) \sum_{x \in \mathbb{L}^d} \mathbb{1}(\zeta_{m_k - l} + x \subset X^c),$$

where ζ_l are the $m_k + 1$ vertex configurations of the k-dimensional unit cell of the section lattice. In order to apply this estimation not only on configurations in $\mathbb{L}^k$, but on configurations in $\mathbb{L}^d$, a reweighting has to be applied, see [RSVW14] and [OS09]. This results in the following estimation of the intrinsic volumes:

$$\hat{V}_{3-k}(X) = 2 \sum_{i=1}^{13} \gamma_i^{(k)} \hat{p}_k(X, L_i),$$

where the rotational integral is approximated as mentioned above by integrating over the 13 section lattices, where each translative integral is weighted with $\gamma_i^{(k)} > 0$.

2.5.4 Orientation

The intrinsic volumes are invariant under rigid motion. Therefore, they are not suited to estimate the orientation of structures depicted within an image. In the following, we will describe various notions of *orientation* and how to estimate these properties on digital image data. Based on the orientations of the grains $\{\Xi_i\}_i$ observed within the images, it is then possible to estimate the orientation distribution of the typical grain Ξ_0.

Surface orientation

One way to define the direction or orientation of an object is the indirect method of analysing its surface orientations. For an elongated convex object, these orientations concentrate around the plane orthogonal to the direction of elongation on the equator. When the objects under investigation are distributed invariant under rotations around the axis of elongation, this results in a so-called *girdle distribution*. For isotropic convex sets, the surface orientations are distributed uniformly on the sphere.

As [OJG90] have shown, this is not true for non-convex objects: Indeed, they present an example of an elongated object having the same surface distribution as a non-elongated object.

The estimation of object elongation via surface orientation is sensible for specific cases, see for example [COH86] where surface orientation is used as a measure of anisotropy of biological cells.

In practice, the surface orientations of a set $X \sqcap \mathbf{F}$, which can be observed within the digital image, are usually estimated by the gradients $\mathbf{g}$ on the voxel data calculated using grey value differences within a small mask. Note that due to the fact that $f(X) = 1$ and $f(X^c) = 0$, the gradients are equal to $\mathbf{0}$ for all voxels not on the surface.

Chord length distribution

A more direct measure of the orientation or elongation of an object is obtained by analysing the *chord length distribution* for lines in different directions.

Theoretically, the chord length distribution is the distribution of the typical chord obtained by the intersection of a random closed set Ξ with a 1-dimensional linear subspace L, see [OS09], Chapter 5. The *mean chord length* $\bar{l}(\theta)$ is the expectation of the length of the typical chord obtained when intersecting Ξ only with affine subspaces of direction θ, denoted by L_θ. It is estimated by intersecting $\Xi \sqcap \mathbf{F}$ with lines $L_\theta + \mathbf{x}, \mathbf{x} \in L_\perp$, which are usually taken from discrete section lattices $\mathbb{L}^1$.

Elongated objects have a larger mean chord length in direction of their elongation.

Inertia tensor

The *moment of inertia* I_x of a rigid body is physically defined as a resistance this body has against a change in rotation around the x-axis. This relation is given by the equation $E_{rot} = \frac{1}{2} I_x \omega^2$ with ω perpendicular to the x-axis, see [Dem06]. When applied to rotations around all three axes of the cartesian system, centred at the centre of mass of the object under investigation, one obtains a three-dimensional, symmetric *inertia tensor* $\tilde{I}$. Its value depends on the rotation axis and on the shape of the rigid body. It can be computed very easily on voxelated images. Note that the *centre of mass* $\mathbf{x}_m$ of an object $X \sqcap \mathbf{F}$ in a digital image is simply the average

location over all its voxels:

$$\mathbf{x}_m = \begin{pmatrix} x_{m,1} \\ x_{m,2} \\ x_{m,3} \end{pmatrix} = \frac{1}{V(X \sqcap \mathbf{F})} \sum_{(i,j,k) \in X \sqcap \mathbf{F}} \begin{pmatrix} i \\ j \\ k \end{pmatrix},$$

where the voxels (i, j, k) are interpreted as vectors in $\mathbb{R}^3$.

For the symmetric inertia tensor $\tilde{I} \in \mathrm{Mat}(3 \times 3)$ with

$$\tilde{I} = \begin{pmatrix} \tilde{I}_{11} & \tilde{I}_{12} & \tilde{I}_{13} \\ \tilde{I}_{12} & \tilde{I}_{22} & \tilde{I}_{23} \\ \tilde{I}_{13} & \tilde{I}_{23} & \tilde{I}_{33} \end{pmatrix}$$

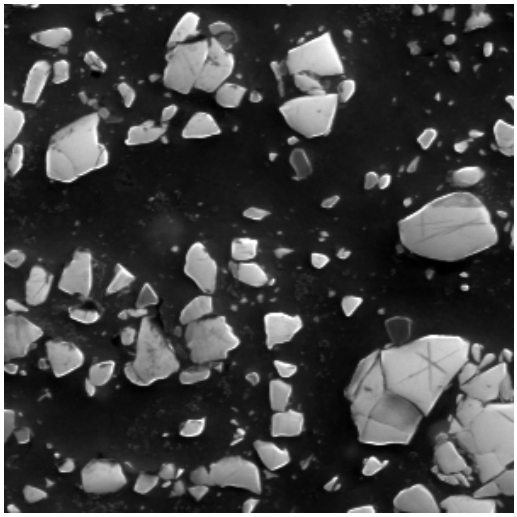
we only have to compute 6 different values, which are given by

$$\begin{aligned}
\tilde{I}_{11} &= \sum_{(i,j,k) \in X \sqcap \mathbf{F}} \left((j - x_{m,2})^2 + (k - x_{m,3})^2 \right) \\
\tilde{I}_{22} &= \sum_{(i,j,k) \in X \sqcap \mathbf{F}} \left((i - x_{m,1})^2 + (k - x_{m,3})^2 \right) \\
\tilde{I}_{33} &= \sum_{(i,j,k) \in X \sqcap \mathbf{F}} \left((i - x_{m,1})^2 + (j - x_{m,2})^2 \right) \\
\tilde{I}_{12} &= - \sum_{(i,j,k) \in X \sqcap \mathbf{F}} (i - x_{m,1})(j - x_{m,2}) \\
\tilde{I}_{13} &= - \sum_{(i,j,k) \in X \sqcap \mathbf{F}} (i - x_{m,1})(k - x_{m,3}) \\
\tilde{I}_{23} &= - \sum_{(i,j,k) \in X \sqcap \mathbf{F}} (j - x_{m,2})(k - x_{m,3}).
\end{aligned}$$

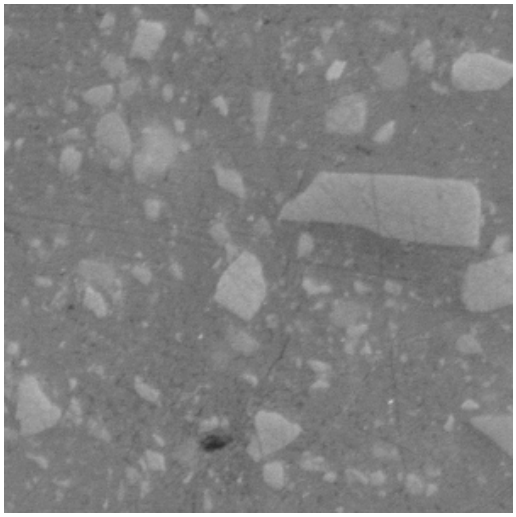
Due to its symmetry, this tensor can be diagonalized. The eigenvectors of the inertia tensor are the three main axes of rotation of the rigid body under investigation. The lowest resistance against rotation is obtained when rotating the body around its longest rotation axis, this axis is given by the eigenvector corresponding to the smallest eigenvalue of the inertia tensor. The inertia tensor was used in [AJ09] for estimating fibre orientations.

2.6 Applications to various data sets

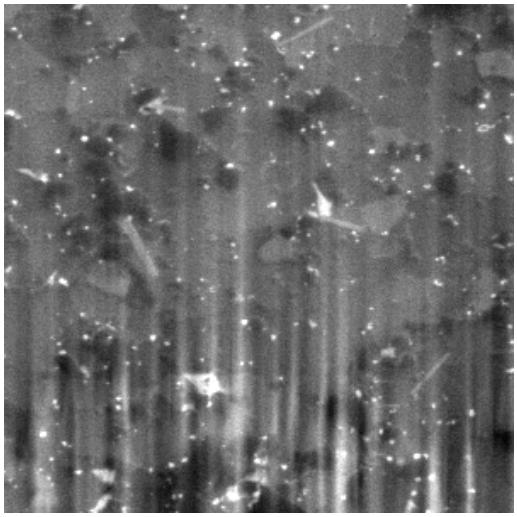
In the following, we will use the methods presented above to analyse the data of *AMC*17. As outlined in the introduction, our data consists of several types of images: Images obtained using SEM and FIB-SEM. The FIB-SEM images are corrupted by the so-called curtaining effect, which leads to striped artefacts parallel to the focused ion beam and to bright laminar artefacts, as displayed in Figure 2.8 (c) and (d). These artefacts can be removed using a decurtaining algorithm, see [FMS], the images displayed in Figure 2.8 (e) and (f) were kindly provided to us by the author. The removal of the curtaining effect leads to a larger volume available for labelling and analysis. However, the images get blurred slightly. Due



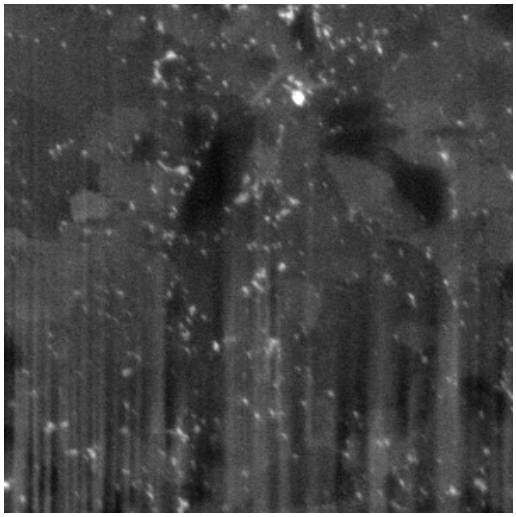
(a) AMC17xe



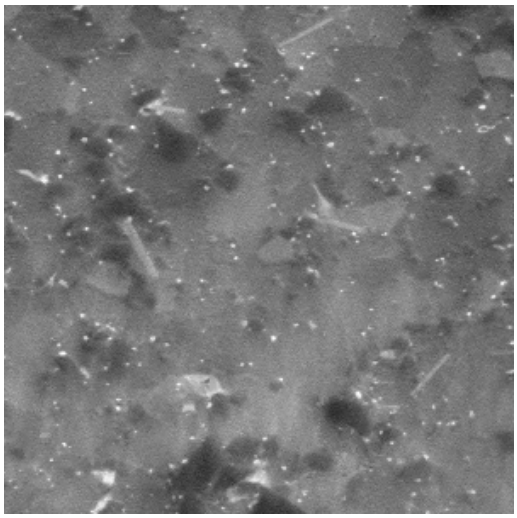
(b) AMC17xfine



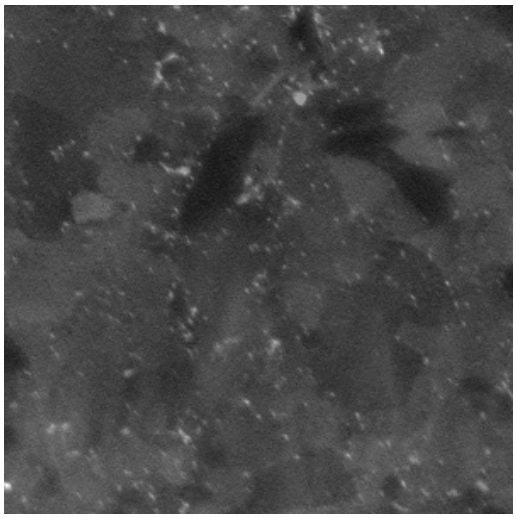
(c) AMC17xxfine



(d) AMC17xfine

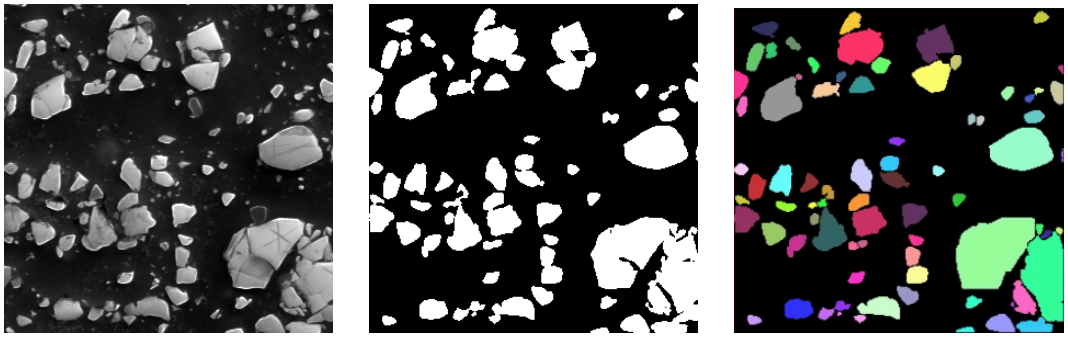


(e) AMC17xxfine



(f) AMC17xfine

Figure 2.8: (a), (b): SEM images. (c), (d): Slices of FIB-SEM images. (e), (f): Slices of FIB-SEM images after removal of the curtaining effect. Displayed are subsections of size 300×300 of the images that were chosen so that the curtaining effect is clearly visible.



(a) Original

(b) Binarisation by fuzzy c-means

(c) Labelled SiC particles

Figure 2.9: Labelling of the SiC particles in *AMC17xe* based on an SEM slice. Displayed is a subsection of size 300×300 pixels of the images.

to the fact that for FIB-SEM imaging the resolution of the FB-SEM images is fixed at approximately $15 \times 15 \times 50$ $(nm)^3$, the curtaining removal is only necessary only for medium-sized particles that are still resolved well with a slight blur. For small particles, the negative effects of the blurring outweigh the benefit of a larger sample volume. Indeed, at an average equivalent diameter of 0.3 μm, the particles in *AMC17xxfine* are so small that a statistically sufficient number of them can be observed on a small sub-volume of the original FIB-SEM data set. The SiC particles within *AMC17xe* are too large to lie entirely within the observation window. For this sample, we relied on SEM images for parameter estimation.

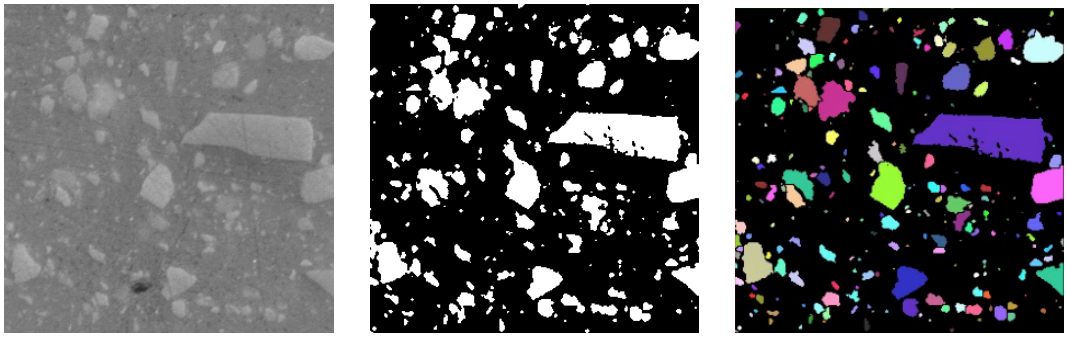
For the SEM images obtained using the scanning electron microscope (without FIB sectioning), there is a wider range of resolutions available, which means that the particles of different sizes are resolved well in the images. However, the images are very different from each other due to the fact that they were obtained by different people over the course of several years, see Figure 2.8. In the following, we will describe methods to binarise these images and label individual particles in a fashion that allows us to infer their distributions and then model the material (Chapter 3).

2.6.1 SEM images

In the SEM data sets, it is not possible to distinguish SiC particles and Al_2Cu precipitations. The Al grains cannot be observed on these images, either. Therefore, in the following we will only show how to segment the SiC particles in these images.

AMC17xe

The SEM image of *AMC17xe* was binarised using the fuzzy-c-means with TV-regularization mentioned above, see [SS12]. The algorithm to minimize this functional is based on first performing some iterations of the fuzzy-c-means clustering algorithm to learn a codebook and then using an ADMM to minimize the functional. We used the implementation in ToolIP [ITW14]. We used 25 iterations of the FCM



(a) Original (b) Binarisation by Otsu (c) Labelled SiC particles

Figure 2.10: Labelling of the SiC particles in *AMC17xfine* based on an SEM slice. Displayed is a subsection of size 300×300 pixels of the images.

and 100 iterations of the ADMM, with $\lambda = 0.1, m = p = 2$ and $\beta = 0.3$, which is a parameter used for the minimisation. The feature mode was set to length 100.

As a next step, we computed the Euclidean distance transform on the SiC label, took the complement to obtain the minima, and computed a preflooded watershed with area threshold $t = 150$ and an 8-neighbourhood on them. By masking the resulting image with the binarised image, we obtained a labelling of the SiC phase. Due to slight over- and undersegmentation of the particles, some particles had to be relabelled manually. In Figure 2.9, the binarisation and label images are displayed.

AMC17xfine

For these images, we first applied a median filter with a square filter mask of size 3. Then, we binarised the image using Otsu's method. We computed the Euclidean distance transform on the foreground (corresponding to the SiC phase) and spread the complement of the float image to 16-bit. On this image, we applied a H-minima transform based on the 8-neighbourhood to eliminate irrelevant local minima. Based on these minima, we computed the watershed image on an 8-neighbourhood. To obtain the segmented SiC particles, we masked with the binary image obtained before. The dynamic d of the H-minima transform was adjusted manually so that over-segmentation was minimized while under-segmentation was kept low. Usually, the dynamic could be chosen between $d = 1000$ and $d = 2500$. Still, there were some particles not segmented correctly, so the over- and undersegmentations were corrected manually, based on the shape of the particles observable in the original image. When large scratches were present in the images, we simply deleted the corresponding labels. An example of the binarisation and labelling results are displayed in Figure 2.10.

2.6.2 FIB-SEM images

On the FIB-SEM images, there are not only SiC particles visible which are quite dark, but also bright spots which are Al_2Cu precipitations. The Al grains have a

(a) Original (b) Binarisation of the SiC (c) Labelling of the SiC

Figure 2.11: Labelling of the SiC particles in *AMC17xfine* based on FIB-SEM. Displayed is one slice of the image.

wide variety of grey values and are therefore not easily distinguishable from the SiC particles, because of the grey value fluctuations.

AMC17xfine

The raw data is corrupted by the so-called curtaining effect which leads to dark and light lines parallel to the focused ion beam, as well as to large bright or dark areas. However, with the method described in [FMS], this effect can be eliminated. We were provided with decurtained images by the authors. The images still have a brightness gradient, so simple thresholding is not a feasible way to obtain a binarisation.

Segmenting the SiC **particles** In all FIB-SEM images not only the material, but also some empty space around it is displayed. Therefore, as a first step we cropped the image. This means we only worked on voxels in $W = [130, 1306] \times [91, 994] \times [0, 421]$ of the original canvas.

As in this case, the SiC particles were darker than the other phases, we started with complementing the image. Then we applied a *mean filter* with a cubic mask of size 3. To binarise the image we used a *Sauvola binarisation* with coefficient $k = -0.1$ and window size $w = 180$. As a next step, we rescaled the image to a resolution of $16.5 \times 16.5 \times 50$ $(nm)^3$, to account for the different resolutions of the focused ion beam cutting and the scanning electron microscope by setting the spacing accordingly. Finally, we performed a *morphological opening* with a cubic structuring element of size 3 to obtain a good binarisation.

On the binarised images, we then computed the *Euclidean distance transform* on the foreground, spread the image to 16-bit and complemented the images. Then we applied a *H-Minima transform* in the 26-adjacency with dynamic $d = 3750$ and a watershed with 26-adjacency. The resulting watershed image was masked by the binary image in order to obtain a labelling of the SiC particles. The intermediate steps of the binarisation and the labelling are displayed in Figure 2.11.

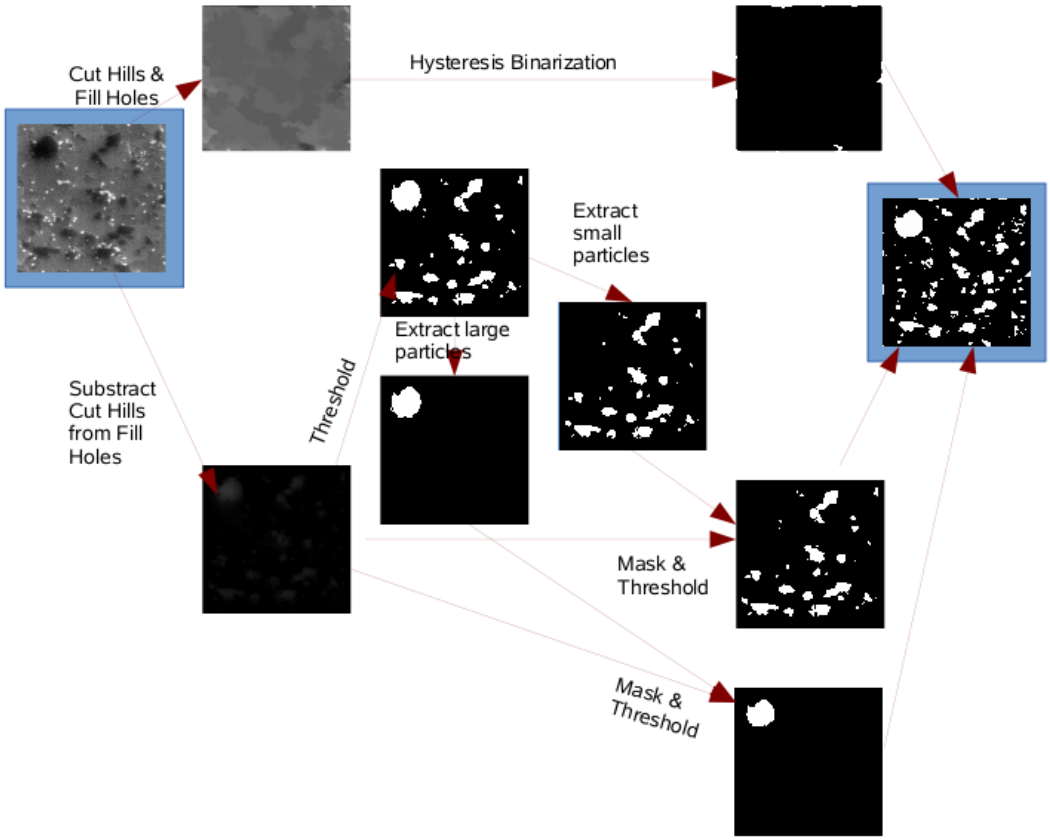


Figure 2.12: Binarisation of the SiC particles in *AMC17xxfine* based on FIB-SEM slices. The blue frames indicate the input and the output image of this process. The smaller images were created using ToolIp, [ITW14], the diagram was generated using GIMP, [tea13].

AMC17xxfine

Segmenting the SiC particles In the case of *AMC17xxfine*, we first aligned the stack using a SIFT alignment implemented in fiji, [SACF+12], restricting the alignment to translational transforms. Then, we binarised each slice of the FIB-SEM stack separately using ToolIp, [ITW14]. The reason for this is that we did not use the decurtained images, to avoid the blurring effect of the decurtaining. However, without the smoothing perpendicular to the SEM slices, which is provided by the decurtaining program, each slice has a slightly different illumination. For each slice, the relevant thresholds were adjusted manually to obtain a good binarisation. Therefore, we chose a small sub-region of size $W = [0, 199] \times [0, 199]$ pixels within each SEM slice.

First, we applied a cuthills and fillholes transform within a 8-neighbourhood to the original slices f to extract the Al phase. Note that minima and maxima connected to the image boundary are not eliminated in that process. These minima and maxima are then binarised using a hysteresis binarisation. Then, the cuthills(f) is substracted from fillholes(f), and the resulting image h is thresholded. This threshold is chosen in order to obtain good masks for large and small particles. Based on the resulting binary image g, we extract a "large particles" image g_L with $g_L = 1$

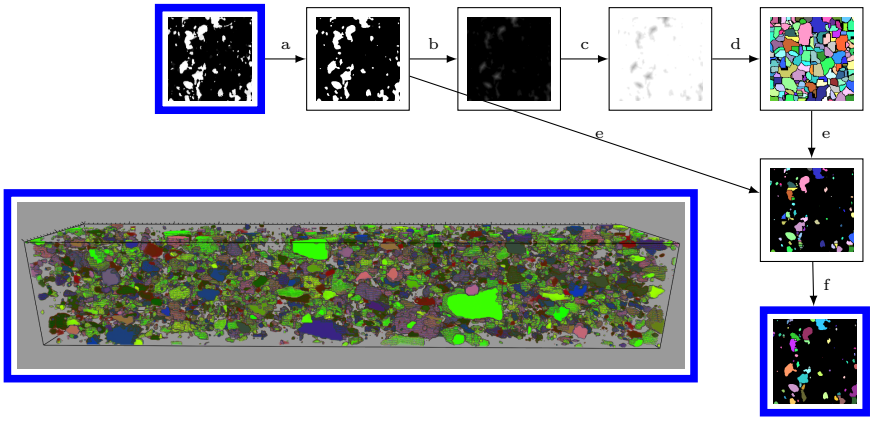


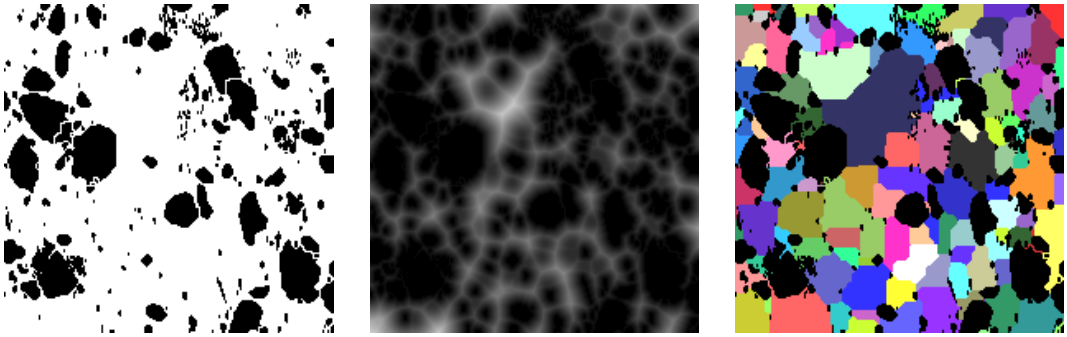
Figure 2.13: Labelling of the SiC particles in *AMC17xxfine* based on FIB-SEM slices, blue frames indicate input and output images of the process.
a: Stack slices and smoothen jumps
b: Euclidean distance transform
c: Complement
d: Watershed
e: Mask watershed image
f: Correct labels.

only for large particles and a "small particles" image g_S with $g_S = 1$ only for small particles. Then, the pixels $g_L^{-1}(1)$ and $g_S^{-1}(1)$ are used as masks for two more thresholdings of h, where only pixels within the masks are thresholded and all other pixels are assigned to the background (with value 0) automatically. This results in the improved binarisations of large and small particles.

The resulting images of the hysteresis binarisation and the binarisations of the large and small particles are combined by taking the pointwise maximum of these three images. For each slice, the thresholds were adjusted manually. A schematic overview of the binarisation process is depicted in Figure 2.12.

In order to label the SiC particles, we stacked the slices to form a 3D-image. As mentioned above, the resolutions of the focused ion beam and the scanning electron microscope are not the same. In this case, the resolution of the FIB was $50nm$ and the resolution of the SEM 15.145 nm. Therefore, we resampled the resulting 3D images to obtain isotropic voxel spacing, using fiji, [SACF+12]. The next steps were all conducted in MAVI, [MAV05]. Since we obtained the binarisations on the slices, we first smoothened the image in the stacking direction by first applying a median filter with square mask of size $w = 3$, then a fillholes in 26-neighbourhood, then a linear closure in stacking direction of length $l = 5$, and finally another median filter with square mask of size $w = 3$. On the resulting image, we computed a Euclidean distance transform image on the foreground. We spread the resulting image to a 16-bit image, took its complement and computed the watershed on it in a 26-neighbourhood. Finally, we masked the watershed image with the binarised image to obtain the labels for the SiC particles.

Due to over- and undersegmetation, we had to correct some labels manually. The labelling process and a volume rendering of the resulting 3D image are depicted in Figure 2.13.



(a) Al phase (b) EDT image (c) Reconstructed grains

Figure 2.14: Labelling of the Al grains based on the assumption that Al_2Cu precipitations occur on grain boundaries. Displayed is one slice of the images. Submitted to appear in [LSB+ed].

Segmenting the Al_2Cu The Al_2Cu precipitations can be easily segmented from the stacked and aligned original images of the image by the application of a Sauvola algorithm with parameter $k = -0.5$ and window size $w = 10$. On the resulting image, we applied a linear opening perpendicular to the slices (i.e. in parallel to the stacking) of width 2. Additionally, the local maxima were extracted from the image stack using the *Extract Hills* algorithm implemented in MAVI [MAV05]. These local maxima were then binarised by a global threshold and masked with the binarisation obtained by the Sauvola method described above. The resulting image was then spread to 8-bit and resampled to isotropy using the bilinear interpolation implemented in fiji [SACF+12]. The resulting image was then binarised by global thresholding. Due to the very small sizes of the precipitations in the images, we abstain from trying to separate them even further and only labelled the connected components.

Segmenting the Al **grains** As mentioned earlier, the Al_2Cu can be assumed to occur on Al grain boundaries. After obtaining a good segmentation of the SiC and the Al_2Cu phases, we automatically have also a segmentation of the Al phase as $X_{\mathrm{Al}} = X \setminus (X_{\mathrm{SiC}} \cup X_{\mathrm{Al_2Cu}})$ for the canvas X. The grains can be easily segmented by first computing an Euclidean distance transform on the binary image f with $f(X_{\mathrm{Al}}) = 1$ and $f(X \setminus X_{\mathrm{Al}}) = 0$, then taking the complement and spreading to 16-bit. Then we took a H-minima transform with dynamic $d = 1000$ and computed the watershed on the remaining minima. By masking the remaining label image with $f^{-1}(1)$, we obtained a label image of the Al grains. The reconstruction of the Al grains is depicted in Figure 2.14.

2.7 Conclusion

We presented a wide variety of segmentation methods and some standard methods for labelling. Due to the nature of the material under investigation and the different circumstances under which the data were imaged, the image data are very

heterogeneous. Therefore, we had to use many different methods to obtain satisfying labellings of the data. Especially in the FIB-SEM images with the fixed resolution, large particles are very hard to segment automatically, small particles can be segmented semi-automatically, while medium-sized particles are easy to segment and label. The SEM images can be mostly segmented and labelled using standard methods, but only because of the gentle grinding applied beforehand. For these images, the preparation before the image acquisition has the largest influence on the quality and facility of the segmentation.

Chapter 3

Statistics and Modeling of Particle Reinforced Aluminium Matrix Composites

In this chapter, we will analyse relevant statistics of the SiC grains, the Al grains and the Al_2Cu which we use to model the material. The analysis is based on the FIB-SEM data set of the *AMCxxfine* and the reconstruction described in Chapter 2. As the intended models are built on random polygons, we use statistics on the convex hulls of the particles, and therefore only model the convex hulls of the SiC particles. This introduces a small error, however, the SiC particles can be expected to be mostly convex. Note that we resampled the data set to obtain isotropy. After the resampling, the size of the data set is $200 \times 200 \times 1178$ voxels at a resolution of 15.145 nm. All figures and minimisations in this chapter were obtained using Matlab [MAT16]. The particle features, namely $\bar{b}$ and V were obtained using the *objectfeatures* in MAVI, [MAV05]. The statistics of these *objectfeatures* were obtained within Matlab. The Laguerre tessellations were generated based on a program written by Claudia Redenbach. All other programs used to analyse 3D image data were programmed using MAVIlib, [Fra11]. The statistical analysis and method to model the SiC particles is submitted to appear in [LSB$^+$ed].

3.1 Proposed Model

We propose to model the *AMC17xxfine* based on a random Laguerre tessellation model for the Al grains, with SiC particles and Al_2Cu precipitations lying on the grain boundaries of the tessellation (and the latter also lying on the surface of the SiC particles). This is an approximation of the real circumstances where the Al grains were formed around the SiC particles. This approximation is necessary in order to obtain a fast-to-compute model that can easily be adapted to other particle sizes. In the following, we will introduce in detail how to model the phases based on the digital image data obtained in FIB-SEM. Since our model is only able to capture convex SiC particles, we analyse the SiC particles based on the convexified

labels within an image f_{conv}. This image was generated from the labelled image f by convexifying the objects in a fashion that no other labels were overwritten.

3.2 Statistics of shapes and sizes of the SiC particles

3.2.1 Correct sampling of the SiC particles

Not all particles within our data set lie completely inside the observation window. This is due to the fact that our data was obtained by FIB-SEM imaging of a small volume of the material (instead of imaging a set of distinct particles). This means we have to account for *boundary effects*:

If we included all particles that were sampled by the window, or if we excluded particles that lie on the boundary of the window, we would introduce a bias into the analysed distributions. This is due to the fact that large particles have a higher probability to be intersected with the observation window's boundary than small particles.

Therefore, it is necessary to use another approach to account for these boundary effects: We analyse only those SiC particles whose midpoints have a larger distance from the boundary than the midpoints of the particles intersecting the boundary of the window. This method is called *reduced sampling*, see [OS09], Chapter 5.2.6. This allows for an unbiased estimation of the particles' volume and shape distributions.

3.2.2 Anisotropy

As it has been shown in [GC05], we generally have to expect anisotropic orientations of the grains due to the manufacturing process: Because of the extrusion process, the particles align in extrusion direction. In Chapter 2 we have presented various methods to investigate anisotropy. In this simple case, we analysed the lengths of the particles' bounding boxes in two directions l_x and l_y perpendicular to the extrusion direction compared to the length l_z parallel to the extrusion direction. Here, x, y and z correspond to the coordinate axes in the analysed image. This yields a simple measure of anisotropy as *elongation* $\mathrm{el} = \frac{2l_z}{l_x+l_y}$.

In general, bounding boxes are largest in the direction of the extrusion. However, this elongation is not independent of particle size. Due to the fact that there are considerably more smaller particles than larger particles, a simple regression fit of the elongation as a function of size would over-emphasize smaller particles. Therefore, we smoothen and resample the scatter of the observations at intervals of 1000 voxels to achieve equidistance of the independent variable *particle size*. As smoothing, a local mean of width 2001 is chosen, so for each resampled volume v_{res}, the corresponding observation is the mean of elongations for the volume v (in voxels) in $[v_{res} - 1000, v_{res} + 1000]$.

The smoothed scatter is then used for a function fit, giving the approximate relation between volume and elongation as

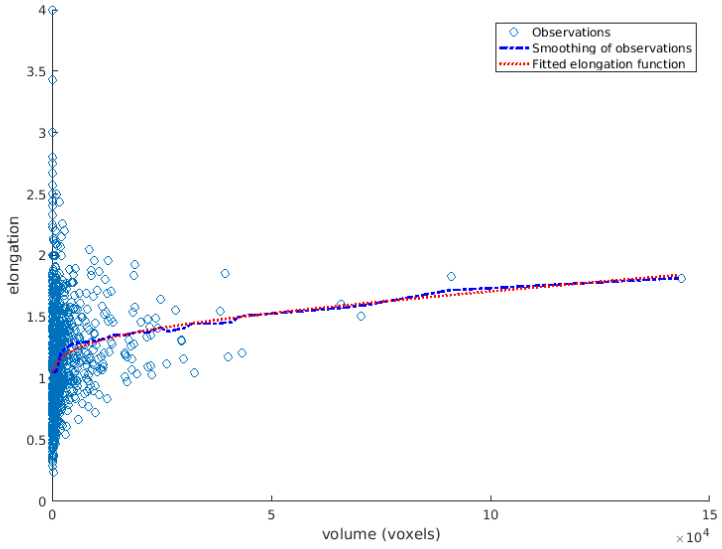


Figure 3.1: Smoothing and function fit of the elongation depending on the volume. On the abscissa the particle volume in voxels, on the ordinate the elongation. Derived from a plot submitted to appear in [LSB$^+$ed].

$$\mathrm{el}(v) = 0.08837 * \ln(170.3v + 14280) + 236500v$$

with an adjusted R^2 of 0.98861.

In order to obtain a more isotropic data set we then rescaled the particles based on their volumes. For the volumes, this rescaling can be directly achieved by dividing the volume of a particle by its elongation. In order to be able to analyse the other intrinsic volumes, we generate a rescaled image $f_{conv,resc}$. We generated this image by the following method: For each label n_i in the image f_{conv}, we estimated the volume $V_3(n_i)$ based on the image data. Based on $V_3(n_i)$, we then determined the relevant rescaling factor $resc_{V_3(n_i)} = \frac{1}{\mathbf{el}(V_3(n_i))}$.

For the rescaling, we extracted the label's bounding box $\mathrm{bbx}(n_i) := [\min x_i, \max x_i] \times [\min y_i, \max y_i] \times [\min z_i, \max z_i]$. Then, for the output image $f_{conv,resc}$, we apply the following mapping $\mathcal{O} : \mathcal{P}_G \to \mathcal{P}_G$:

$$f_{conv,resc}(x,y,z) = f_{conv}\left(x, y, resc_{V_3(n_i)}\left(z - \frac{\min z_i + \max z_i}{2}\right) + \frac{\min z_i + \max z_i}{2}\right)$$

if $\left(x, y, resc_{V_3(n_i)}\left(z - \frac{\min z_i + \max z_i}{2}\right) + \frac{\min z_i + \max z_i}{2}\right) \in W$, the observation window. In other cases, the voxel gets projected to the border of W. This rescaling ensures that the centre of mass of the particles is not changed for particles lying inside of W and that particles lying on the boundary of W are still connected to the boundary. The further analysis is based on this image of rescaled particles.

The correction of these orientations and elongations is useful because it allows us to analyse the underlying particle shapes in terms of sphericity. If we did not correct for the elongation, we would confound deviations from the sphere due to elongation

with deviations from the sphere due to sharp edges. Additionally, we would have to generate anisotropic tessellations to fit the typical grain of the SiC.

3.2.3 Volume distribution of SiC particles

In order to obtain a model that is easy to handle and allows for straight-forward stereological estimation, we strive to find a parametric model to fit to the volume distribution of the SiC particles. It has been shown in previous works [FVF88] that typical volume distributions for granular structures are the *logarithmic normal distribution* and the *gamma distribution*, see [Wal] for an introduction on these distributions, with the densities f_{logn} and f_Γ, respectively:

$$f_{logn}(x) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{(\ln(x)-\mu)^2}{2\sigma^2}\right) & \text{if } x > 0 \\ 0 & \text{else} \end{cases}$$

$$f_\Gamma(x) = \begin{cases} \frac{a^b}{\Gamma(b)} x^{b-1} \exp(-ax) & \text{if } x > 0 \\ 0 & \text{else,} \end{cases}$$

with parameters $\mu \in \mathbb{R}, \sigma > 0$ and $a, b > 0$ and the gamma function $\Gamma(p)$. Note that by definition, the logarithm of a log-normally distributed random variable is normally distributed with parameters μ and σ.

In order to find the distribution of the SiC volumes, we analysed the volume distribution of SiC particles based on the convex hulls of the particles, f_{conv}. Indeed, the volume (measured in voxels) of the particles is best fit by a log-normal distribution with $\mu = 4.5210$ and $\sigma = 2.2688$. At a resolution $(r\text{u})^3$ per voxel, where u refers to the unit of measurement (m, nm, etc), this transforms to $\mu_r = \mu + 3\ln r, \sigma_r = \sigma$. At a resolution of $(15.145\ nm)^3$, we get $\mu_{15.145} = 12.674$ for measurements in nm, the mean particle volume is 4188017.6 $(nm)^3$, which corresponds to an equivalent spherical diameter of 200 nm. It has to be noted, however, that a *Lilliefors test* on a subvolume of size $200 \times 200 \times 300$ with 993 particles fails to reject the null hypothesis of normal distribution of the log-volumes at $p = 0.3771$, while the same test on a set of 3064 particles rejects the normal distribution at minimal $p = 0.001$. Figure 3.2, left, shows the corresponding histogram of the log-volumes with a normal fit. Note that the lack of negative values stems from the fact that each particle has at least 1 pixel.

The coefficient of variation of the volume, measured directly on the image data, is 4.4892. This deviates largely from the theoretical coefficient of variation, obtained when using the fitted parameters,

$$\sqrt{\exp(\sigma^2) - 1} = 13.1.$$

This deviation is due to the fact that the coefficient of variation is a very sensitive measure: Even for a sample of 10000 lognormally distributed random variables with parameters $\mu = 4.521$ and $\sigma = 2.26882$, we only measure a coefficient of variation of 6.9993.

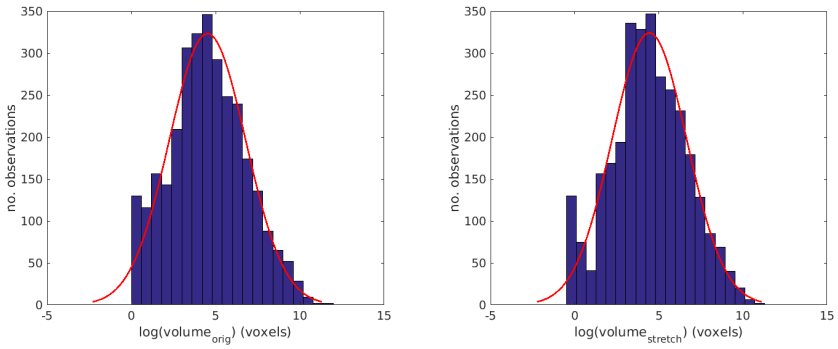


Figure 3.2: Histogram fits of the logarithm of the SiC volume distribution. Left: Histogram fit of the original volume distribution. Right: Histogram fit after rescaling the particles according to the elongation function. Submitted to appear in [LSB$^+$ed].

We therefore propose to use the *quartile dispersion coefficient* v_Q, see [wik], as a measure for the variability of the data and to check the fit of the parameters. It is defined as

$$v_Q = \frac{q_{0.75} - q_{0.25}}{q_{0.5}},$$

where q_x is the x-quantile of the data, this means for example the median is $q_{0.5}$. The quartile dispersion coefficient of the volumes of our data is 4.6897, the theoretical quartile dispersion coefficient of the fitted values is 4.4030. We conclude that the quartile dispersion coefficient is an adequate measure for variability in our data. When we rescale the particles to obtain isotropy as described above, the volume distribution preserves its overall log-normal shape, see Figure 3.2, right. The parameters of the rescaled distribution are $\mu_{resc} = 4.4412$ and $\sigma_{resc} = 2.2252$.

3.2.4 Analysis of shape factors

In many applications the reinforcement particles have a certain, deterministic shape, given by the manufacturer. In our material this is not the case: although SiC forms many different crystal lattices, the particles reinforcing the Al within the material under investigation do not have a crystal shape. Instead, the shapes of the particles are randomly distributed. They have sharp edges and are mostly convex. In the following, we will analyse the shape distribution of the SiC particles on the image $f_{conv,resc}$, that is after their anisotropy has been corrected.

When analysing the shapes, it is sensible to only consider labels that consist of at least 400 voxels. This is due to the fact that it is impossible to get a notion of "shape" for smaller voxel sets, see [Vec14], Chapter 2. A shape factor is highly dependent on measurement of surfaces and mean curvature. When using too few voxels per label, an error of one or two pixels could already largely change the perceived shape of a particle.

We can now observe the so-called *isoperimetric shapefactors*. They describe the deviation of the particle shapes from the ideal shape of a ball. This deviation is

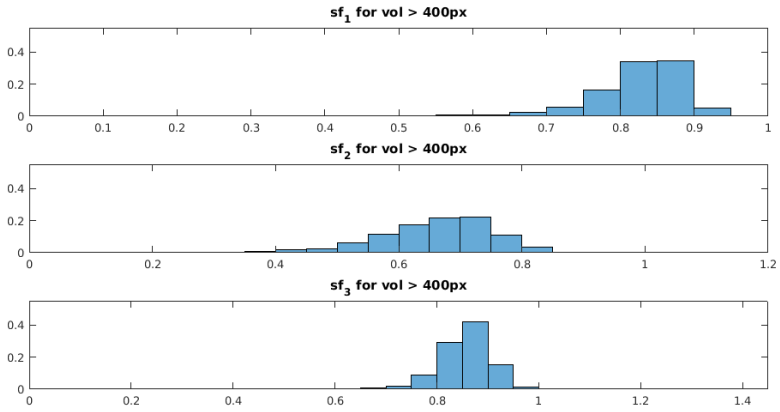


Figure 3.3: Histogram of various shapefactors for particles with more than 400 voxels. Derived from a plot submitted to appear in [LSB$^+$ed].

measured in terms of relations of intrinsic volumes, which have known relationships for balls. See Chapter 1 for the definition of the intrinsic volumes.

The intrinsic volumes can be combined to the following shape factors:

$$sf_1 = \frac{6\sqrt{\pi}V}{\sqrt{S^3}}$$
$$sf_2 = \frac{48\pi^2 V}{M^3}$$
$$sf_3 = \frac{4\pi S}{M^2}$$

These shape factors are 1 for the ball, smaller than 1 for convex objects. For polyconvex objects, sf_2 and sf_3 can be larger than 1, see [Vec14], Chapter 2.

3.2.5 Shapefactor distribution

The histograms of the shapefactors measured on $f_{conv,resc}$, sf_1, sf_2 and sf_3 are displayed in Figure 3.3.

Some particles are not convex, but rather polyconvex, which can be seen easily by the fact that they obtain shapefactors sf_2, sf_3 larger than 1. The reason for this is that the image f_{conv} was obtained under the condition that only the background was overwritten, so that touching particles do not overwrite each other. When restricting ourselves to the convex shapes, we get the following characteristic properties, as displayed in Table 3.1.

We have to consider that we corrected for particle elongation before analysing the shapes. Correcting an ellipsoid's elongation would render a sphere. However, the truncated histograms displayed in Figure 3.4 lack a shape factor clustering near the value 1. We can conclude that the particles are definitely not spherical or ellipsoidal.

	mean	stddev	median	v_q
sf_1	0.8260	0.0617	0.8373	0.0822
sf_2	0.66612	0.0943	0.6696	0.1813
sf_3	0.8573	0.0480	0.8630	0.0741

Table 3.1: Statistics of the shape factors, based on values in $[0, 1]$ and with $vol > 400$ voxels. Submitted to appear in [LSB$^+$ed].

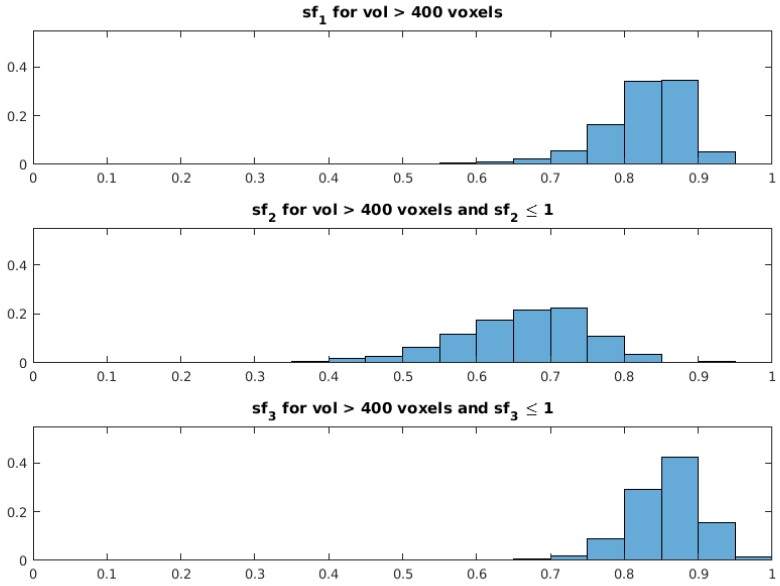


Figure 3.4: Histogram of various shapefactors for particles with more than 400 voxels, truncated at 1. Derived from a plot submitted to appear in [LSB$^+$ed].

3.3 Distribution of the typical grain of SiC particles

Based on the investigations in the previous sections, we propose to model the SiC particles' shape based on the typical cell of a random Laguerre tessellation that was generated on densely packed balls. The Laguerre tessellation has been introduced in Chapter 1.

The shapes and sizes of the tessellation's cells can only be influenced indirectly by the parameters of the underlying ball packing, as was shown in [Red09]. In this paper, they fitted polynomials of degree 3 to model the relationship of the coefficients of variation of the Laguerre tessellation, cv_{Tess}, and the coefficients of variation of the underlying ball packing, cv_{Balls}. Although the polynomial fit is well-suited for their application, it fails to provide sensible extrapolations for our application, especially if we consider generating a ball packing that is able to generate a Laguerre tessellation of $cv = 13.1$. Therefore, based on the data in [Red09], we propose to fit the relationship by:

$$cv_{Tess,1} = a \cdot cv_{Balls}^{b}$$
$$\text{or}$$
$$cv_{Tess,2} = \frac{p_1 \cdot cv_{Balls} + p_2}{cv_{Balls} + q_1}.$$

In the exemplary case that the ball packing achieves the volume fraction $V_V = 0.6$, which results in the strongest correlation of cv_{Tess} and cv_{Balls}, we obtain the parameters $a = 0.7554, b = 0.9108$ and $p_1 = 10.15, p_2 = 0.04613, q_1 = 12.41$. These fitted functions have the advantage that cv_{Tess} is monotonically increasing for increasing cv_{Balls}, which is not the case for the cubic polynomial fit provided in [Red09].

However, for $cv_{Balls} = 10$ and $V_V = 0.6$, based on a simulation on 1904 balls, we can only attain $cv_{Tess} = 3.691$ as estimated on the simulated data, while the theoretical values obtained with our fits would propose $cv_{Tess,1} = 6.15$ or $cv_{Tess,2} = 4.53$.

At this coefficient of variation and packing density, the resulting cell shapes differ greatly from the shape factor distributions of the observed particles. This stems from the fact that in order to obtain the largest coefficient of variation possible, it is necessary to pack the balls very densely. However, the large coefficient of variation means that the probability of extreme cell volumes is high. These large cells are then much rounder than the particles observed in the images.

We conclude that it is not sensible to try to fit the volume distribution and the shape distributions at the same time. Therefore, we only aim at modeling the shapes of the particles by a Random Laguerre distribution. The correct volume distribution has to be attained by isotropic rescaling of the polygons. In order to model the typical SiC particle, we propose the following three-step approach:

- First, fit the empirical shape factor distributions of the rescaled particles by

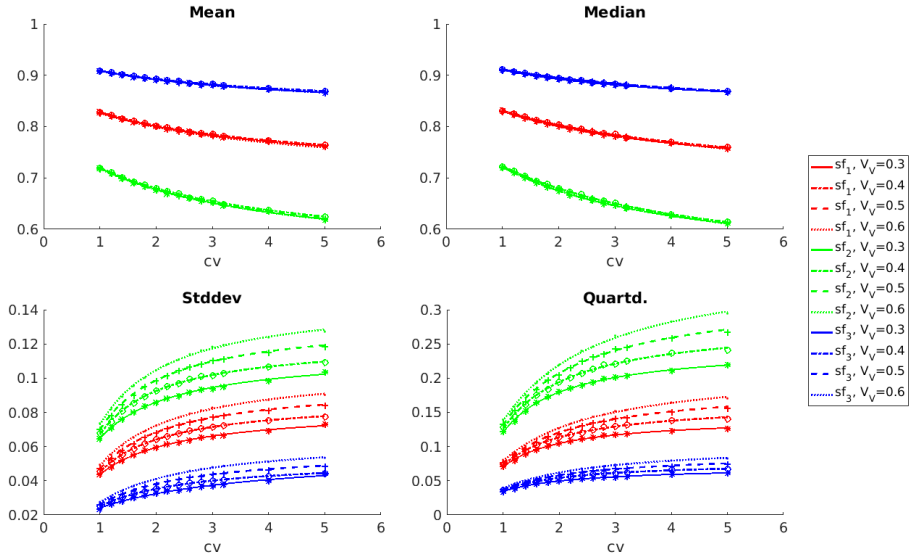


Figure 3.5: Data points for the statistics of sf_1, sf_2, sf_3 for $cv_{Balls} \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 4.0, 5.0\}$ and $V_V \in \{0.3, 0.4, 0.5, 0.6\}$ with function fits. To appear in [LSB$^+$ed] (submitted).

the cells of a random Laguerre tessellation, neglecting the volume distribution

- Secondly, sample cells from the fitted Laguerre tessellation and isotropically rescale them to obtain the desired volume distribution without changing the isoperimetric shape factors of the cells.

- Rescale these polygons anisotropically according to the relationship of volume and anisotropy to obtain elongated polygons with volume distribution parameters $\mu = 4.5210$ and $\sigma = 2.2688$.

In the following we will describe this process in detail.

3.3.1 Fitting the SiC shapes by a random Laguerre tessellation

As described above, we will generate the random Laguerre tessellation based on densely packed balls. Experience has shown that log-normally distributed ball volumes give good fitting results for the resulting random Laguerre tessellation when modeling engineering materials, especially when considering shapes, see [Red09]. This means the parameters to be fitted are the coefficient of variation of the ball volumes, denoted by cv_{Balls} and the ball packing fraction, denoted by V_V. Since we do not want to fit the SiC volume distribution at this point of the modeling, the mean volume of the Laguerre cells is not important. In order to fit the shape factors we proceed the same way as proposed in [Red09] for fitting characteristics of aluminium foams:

Based on a grid for the parameters (cv_{Balls}, V_V), we fit for each volume fraction V_V and for each of the parameters in Table 3.1 a rational function in the coefficient of

variation cv_{Balls} of the underlying ball packing. This means for each volume fraction V_V we obtain 12 different functions in cv_{Balls}, denoted by $f^j_{x,y}(cv_{Balls})$.

We start with a preliminary study to determine the relationships between cv, V_V and the shape factor parameters on a larger scale. Therefore, we generate tessellation models for $cv_{Balls} = \{1, 2, 5, 10\}$ and $V_V = \{0.3, 0.4, 0.5, 0.6\}$. These parameters cover a wider range than [Red09]. We find that a rational function of the form

$$f^j_{x,y}(cv_{Balls}) = \frac{p^j_{1,x,y} \cdot cv_{Balls} + p^j_{2,x,y}}{cv_{Balls} + q^j_{1,x,y}}$$

provides a good fit for the different values of V_V, indexed in the function by j, and all investigated parameters x of the shape distributions y, depending on the shape factor cv_{Balls}. For each shape factor $y = \{sf_1, sf_2, sf_3\}$ and each volume fraction $j = \{0.3, 0.4, 0.5, 0.6\}$ we get four different functions, namely $f^j_{\text{mean},y}$,$f^j_{\text{stddev},y}$,$f^j_{\text{median},y}$ and $f^j_{\text{interq.},y}$, see Tables A.1 to A.4 in Annex A.1. To find the optimal parameters ${V_V}^{**}$ and cv^{**} for our random tessellation model, we minimize the relative differences between the data and the fitted functions $f^j_{x,y}(cv_{Balls})$.

So the optimal coefficient of variation cv^{**}_j for each volume fraction j is obtained by

$$cv^{**}_j = \arg\min_c \sqrt{\sum_{x,y} \left(\frac{f^j_{x,y}(c) - m_{x,y}}{m_{x,y}} \right)^2},$$

where $\cdot_{x,y}$ refers to the x-statistic (mean, median, standard deviation or quartile dispersion coefficient) of the y-shape factor (sf_1, sf_2 or sf_3) estimated from simulated realisations of the models. The statistics of the rescaled particles are referred to by m. The optimal values cv^{**} and ${V_V}^{**}$ are then obtained at the cv^{**}_j with the minimal value of the difference function.

By minimizing the sum of the squared relative differences between the fitted functions and the moments of the data, we obtain a first approximate 1.4068 and ${V_V}^* = 0.6$.

In order to be able to generate statistics with a smaller variance for each Laguerre tessellation, we sample 5 random Laguerre tessellations based on force-biased ball packings, each consisting of 10000 balls. Additionally, we refine the values of cv to investigate

$$cv = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 4.0, 5.0\}$$

in order to increase the quality of the fits. Due to the fact that mean and median are correlated with a correlation coefficient > 0.996 for all three shape factors, we only use the mean, standard deviation and quartile dispersion coefficient for this refined fit. The minimal value of the error function is 0.5017. It is attained at $cv^* = 1.4505$ and ${V_V}^* = 0.6$. The parameters of the fitted curves are displayed in Table 3.2. The data points and the function fits are displayed in Figure 3.5. The statistics for a realisation of the Laguerre tessellation on 10000 balls packed with parameters $cv = 1.4505$ and $V_V = 0.6$ are displayed in Table 3.3.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.6}$	0.6983	0.5313	0.8242
$p_{2,\text{mean},0.6}$	2.305	2.043	2.778
$q_{1,\text{mean},0.6}$	2.634	2.586	2.964
$adjR^2_{\text{mean},0.6}$	0.9993	0.9993	0.9992
$p_{1,\text{median},0.6}$	0.6608	0.4756	0.8083
$p_{2,\text{median},0.6}$	3.636	3.012	4.206
$q_{1,\text{median},0.6}$	4.176	3.83	4.501
$adjR^2_{\text{median},0.6}$	0.9991	0.9991	0.9990
$p_{1,\text{stddev},0.6}$	0.1062	0.1467	0.06515
$p_{2,\text{stddev},0.6}$	-0.03574	-0.05112	-0.01823
$q_{1,\text{stddev},0.6}$	0.4445	0.3158	0.7238
$adjR^2_{\text{stddev},0.6}$	0.9983	0.9979	0.9988
$p_{1,v_Q,0.6}$	0.2151	0.3716	0.104
$p_{2,v_Q,0.6}$	-0.073025	-0.1189	-0.03168
$q_{1,v_Q,0.6}$	0.8084	0.8483	0.8471
$adjR^2_{v_Q,0.6}$	0.9974	0.9981	0.9980

Table 3.2: Parameters of the fitted curves for $V_V = 0.6$ with their adjusted R^2 goodness-of-fit statistic. Based on 5 random Laguerre tessellations based on force-biased ball packings, each consisting of 10000 balls for $V_V = \{0.3, 0.4, 0.5, 0.6\}$ and $cv = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 4.0, 5.0\}$. To appear in [LSB$^+$ed] (submitted).

	stat. on sim.	stat.*	error
mean(sf_1)	0.8132	0.8260	-1.5519%
mean(sf_2)	0.6984	0.6612	5.6246%
mean(sf_3)	0.9007	0.8573	5.0582%
stddev(sf_1)	0.0617	0.0608	1.3628%
stddev(sf_2)	0.0907	0.0943	-3.8386%
stddev(sf_3)	0.0348	0.0480	-27.5404%
$v_q(sf_1)$	0.1036	0.0822	26.11559%
$v_q(sf_2)$	0.1810	0.1813	-0.1325%
$v_q(sf_3)$	0.0514	0.0741	-30.5770%

Table 3.3: Comparison of statistics of the shape factor obtained by a realisation of a Laguerre tessellation based on a ball packing with $cv = 1.4505$, $V_V = 0.6$ with the statistics stat.* obtained on the $f_{conv,resc}$ for $sf_{2,3} \leq 1$ and $vol > 400$ voxels. To appear in [LSB$^+$ed] (submitted).

3.3.2 Sampling of the cells and rescaling

By choosing a random cell c_i of the fitted Laguerre tessellation uniformly, we sample the typical cell of the tessellation, which has the desired shape distribution. However, as was pointed out above, this cell does not have the same volume distribution as the SiC particles. Therefore, we sample a log-normally distributed random variable v_i (for the fitted parameters $\hat{\mu}$ and $\hat{\sigma}$). The sampled cell is then isotropically rescaled so it has volume v_i: Each vertex $X = (x, y, z)$ can be identified with the vector $\vec{X}$. The isotropic rescaling is achieved by multiplication with a suitable scalar ν.

In order to achieve the correct anisotropy, for each cell the z-component (which is parallel to the extrusion direction by definition) is multiplied by an anisotropy factor based on an approximate inverse of the fitted elongation function. This factor f_{sc} is obtained based on an initial estimation

$$f_{sc} \approx 7.117 \times 10^{-6} v_{iso} + 1.292$$

for the isotropic volume v_{iso} which is followed by a fixed-point search to find the exact factor, see [LSB$^+$ed] (submitted).

The rescaled cell $c_i^{'}$ then has the same distribution as the particles we investigated above with respect to the estimated statistics.

3.4 Statistics of the Al grains

We analyse the Al grains based on the reconstructed grain structure, see Chapter 2. In contrast to the previous section where we analysed the typical grain of a germ-grain model, the Al grain structure is a tessellation. Typical characteristics of such granular structures, which we want to model by cells of a random Laguerre tessellation, are the mean number of facets per cell, $\bar{F_C}$; the standard deviation of the number of facets per cell, stdF_C; the mean number of edges per facet, $\bar{E_F}$; and its standard deviation stdE_F; the mean cell volume $\bar{V}$ and its standard deviation std$V$; the mean surface area $\bar{S}$ and its standard deviation std$S$ and the mean mean width $\bar{\bar{b}}$ and its standard deviation std$\bar{b}$. Due to the reconstruction of the grains, which was based on the assumption that Al_2Cu precipitations occur on grain boundaries, the surface areas of the grains are very rough. The grains are often non-convex. Earlier results for cellular structures have shown that the fit of the surface area does not give good results even for labellings with smoother label boundaries, see [Red09]. Therefore, we exclude the surface area related statistics from the modeling. The mean number of facets per cell is usually estimated by the average over the number of neighbours of each cell, the number of edges per facet can be estimated under the assumption of normality, see [Lau07]: Under this assumption, edges are contained in the intersection of three facets.

However, in our data set, there are not only other grains that can be neighbours of a cell, but the neighbour could also be a SiC grain. This means additionally that earlier results for foams in [Red09] cannot be applied to our data set. For our analysis, we included the SiC particles into the count of neighbours, but excluded grains that

were intersected by the image border, this means we applied a minus-sampling. For consistency, we also used minus-sampling for the other statistics of the Al phase. The minus-sampling was implemented using the Miles-Lantejoul [CSKM13] weights output by MAVI [MAV05]. For a space-filling tessellation, the mean cell volume is inversely related to the number of cells per unit volume, $N_V = \frac{\nu_3(W)}{V}$, which can be easily estimated on the images by

$$\widehat{N_V} = \frac{\sharp(\text{particles} \cap W) - \frac{\sharp(\text{particles} \cap \partial W)}{2}}{\nu_3(W)}$$

based on the observation window W, see for example [BVJ05], Chapter 2. On the window of size $[0, 199] \times [0, 199] \times [0, 1177]$ voxels, we observe 8871 grains, covering a volume fraction of 87%. Using the Miles-Lantejoul sampling, we measure a mean grain volume of 5278.1 voxels (18.34×10^6 $(nm)^3$) and the standard deviation of the grain volume is 10887 voxels (37.82×10^6 $(nm)^3$), the mean diameter is 23.0598 units (349.2412 nm), the standard deviation of the diameter is 22.2731 units (337.3263 nm). The average number of facets per cell is 17.5856, its standard deviation is 10.9805. The average number of edges per facet is 5.7029, its standard deviation is 3.5609.

3.5 Statistics of the Al_2Cu precipitations

In an observation window of size $[0, 199] \times [0, 199] \times [0, 1177]$ voxels there are 797675 connected components consisting of Al_2Cu, using an unbiased sampling rule. The average volume of a component is 59.0724 voxels, or 20521 $(nm)^3$. The average diameter of the corresponding ball of a precipitation is 3.4972 units. The volume fraction of the Al_2Cu phase is at 2.86%. The standard deviation of the volume is 388.5858 voxels (1349900 $(nm)^3$). In general, the precipitations are too small to analyse any other intrinsic volumes without introducing a big error.

3.6 Fitting a tessellation to the Al grains by modeling the complete particle system

As before with the parameter-fitting of the SiC particles, we want to generate the random Laguerre tessellation based on densely packed balls. The parameter-fitting process is essentially the same as the fitting process for SiC, however, since our model assumes that the SiC particles lie on the facet system of the tessellation modeling the Al grains, and we can not observe the whole Al tessellation, we generate realisations of the SiC phase on top of the Laguerre tessellations.

For various values of cv and V_V we generate force-biased packed balls. The intensity N_V of the balls can be estimated from the number of Al grains within the observation window W. Experiments show that under the model assumptions, only a small percentage of the modeled grains are completely covered by the polygons modeling the SiC particles. Therefore, we use the intensity estimate $\widehat{N_V}$ for the Al grains without a correction.

Based on these densely packed balls, a Laguerre tessellation is generated. Under the model assumption, the SiC particles must lie on the facets of the Al tessellation. We therefore generate a SiC model on this facet system as described in 3.6.1 below. In order to fit the parameters of the Laguerre tessellation $\mathcal{T}_{\mathrm{Al}}$ used for modeling the Al, we use only statistics on $\mathcal{T}_{\mathrm{Al}} \setminus P_{\mathrm{SiC}}$, with P_{SiC} denoting the polygons modeling the SiC. Based on the statistics on this model, we can fit functions $g_x^j(cv)$ in cv, as it was done in [Red09], and minimize the mean error:

$$cv_j^{**} = \arg\min_c \sqrt{\sum_x \left(\frac{g_x^j(c) - m_x}{m_x} \right)^2},$$

for each volume fraction $j \in \{0.3, 0.4, 0.5, 0.6\}$ and the x-statistic ($\bar{F_C}$, stdF_C, $\bar{E_F}$, stdE_F, $\bar{\bar{b}}$, std$\bar{b}$ and std$V$) estimated for these parameters, and the x-statistics of the analysed phases, m_x. For the function fits, we used rational functions of the form $g = \frac{p_1 \cdot cv + p_2}{cv + q}$ or cubic polynomials. Again, the optimal values cv_{**} and V_V^{**} are obtained at the cv_j^{**} with the minimal value of the difference function.

3.6.1 Modeling the SiC

In terms of the modeling of the SiC phase of the AMCxxfine material, we have so far described how to fit the (anisotropic) grain distribution. The underlying point process, constituting the *germs* of the germ-grain model, is assumed to be stationary within the observation window generated by the FIB-SEM imaging.

Additionally, we can observe that the SiC particles are often lying very close to each other. Based on these observations we propose the *random sequential adsorption + move* algorithm, which is an adaptation of the algorithm used in [Esc12] to model concrete.

The essential idea of this algorithm is that the polygons which model the SiC particles are added one by one, based on a Poisson point process on the Al grain facets. As the Al tessellation is stationary, the Poisson point process of constant intensity λ is, too. Whenever a newly added particle intersects with a previously added particle, the newly added particle is moved until no more overlap occurs. Only then another particle is added. The exact and complete algorithm is described in the following:

- Input: Library of 10000 grains based on a random Laguerre tessellation generated on densely packed balls with log-normal volume distribution and coefficient of variation $cv_{Balls} = 1.4505$ and volume fraction $V_{V,Balls} = 0.6$, window W, volume fraction $V_V = 17\%$ of the SiC phase.
- Based on the parameters μ and σ and the rescaling factors, calculate the mean volume v of the grains.
- Set the intensity of the germ process to $\lambda = \frac{V_V}{v}$.
- Determine the number of germs N to lie within the window W by $N\ Pois(\lambda * \nu_3(W))$.

- Sample N grains from the library, now denoted by $(g_i)_{i=1,\dots,N}$.
- Sort (g_i) by descending size.
- For each $i \in \{1, \dots, N\}$, generate a point x_i within the window W lying on a facet of the tessellation modeling the Al, weighted by the facet area. This means x_i is uniformly distributed on the facet system.
- Starting with g_1, insert the grains centred in x_i into the window W:
  - If g_n overlaps with an already inserted g_m, move g_n into a random direction within the chosen facet until no overlap occurs. If an edge of the facet is reached, sample another possible location x_i.
  - Use periodic boundary conditions in case a grain cuts the window boundary.

Note that there are several aspects as how to implement this algorithm efficiently:

Usually, the grains are considerably smaller than the observation window W. Therefore, it is sufficient to check for intersection only in a small neighbourhood. Its size is given by the largest diameter of the largest particle, ie, the particle with the widest length in z-direction. Practically, we implement this the following way:

Assuming that the largest particle has a length of d, we subdivide the window W into cubes of side length at least d; specifically, the side length l of the cube is chosen as

$$l \geq d$$

so that l is a factor of the side lengths of the (cuboid) window W. Usually, we chose a cubic window with side length $n * 100$ for $n \in \mathbb{N}$. In case this largest particle is longer than the window width, the particles are resampled.

For a particle to be inserted into a cube, possible intersections are only tested for that cube and the 26 surrounding cubes. As in our case the particles are longest in the z direction, we can also adapt the cubes to cuboid size in order to incorporate this knowledge. In most of the cases this is, however, not necessary.

Additionally, since the grains are convex polygons, we can easily check for intersection by using the method described in [Ebe08]. Additionally, this check for intersection can be computed in parallel to increase the speed of the algorithm.

3.6.2 Parameter fit for Al grain tessellation

As described above, we used the statistics on $\mathcal{T}_{\text{Al}} \setminus P_{\text{SiC}}$, with P_{SiC} denoting the polygons modeling the SiC particles, to fit the parameters of the random Laguerre distribution. The simulations are based on Laguerre tessellations of 12539 cells, modeling the Al grains within a window of size $[400 \times 400 \times 400]$ voxels. For each parameter set (cv_{Balls}, V_V) we generated two realisations.

The data points and the function fits are displayed in Figure 3.6. It is clear from the data that there are huge fluctuations in the statistics, making it hard to identify

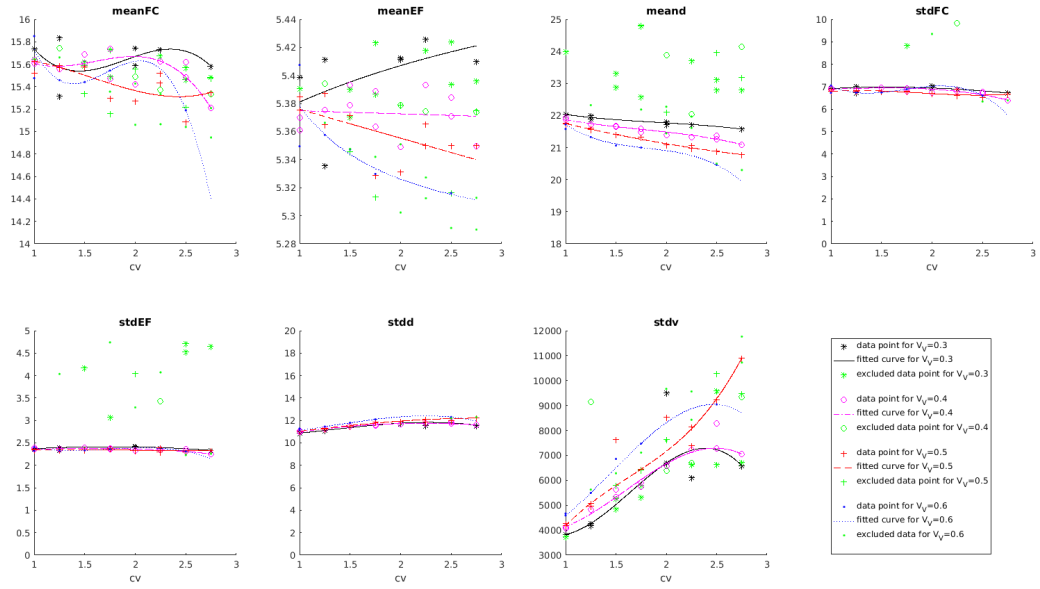


Figure 3.6: Data points for the statistics of F_C, E_F, $\bar{b}$ and std$v$ for $cv_{Balls} \in \{1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75\}$ and $V_V \in \{0.3, 0.4, 0.5, 0.6\}$ with function fits.

actual relationships between the input parameters and the statistics. Therefore, we excluded outlier observations obs_i that deviate from the data median by more than 3 times the median distance to the median of all observations, $3 \cdot \text{median}(|obs_i - \text{median}(obs)|)$. Additionally, we used the Matlab option for robust fitting, LAR [MAT16].

When including all seven proposed statistics, the optimal values for the ball packing are $cv^{**} = 2.2103$, $V_V^{**} = 0.6$ at a value of the error function of 0.7073. Tables containing the parameters of the fitted functions and their goodness-of-fit can be found in Appendix A.2.

3.7 Modeling of the Al_2Cu precipitations

Based on the observations, we suggest to model the Al_2Cu precipitations as balls of radius R, centred on a Cox process. We suggest to generate this Cox process as stationary Poisson point process with parameter λ with respect to the surfaces of the SiC phase as well as the surfaces of the Al phase.

Assuming the low volume fractions as stated above, we ignore intersections of balls on the edges of two facets. Then the balls generate a two-dimensional Boolean model on each facet. Within each facet, the surface fraction s_s covered by this two-dimensional Boolean model is given by

$$s_s = 1 - \exp(-\lambda \pi R^2).$$

The total amount of surface area per volume covered by the two-dimensional Boolean

model is given by $S_V \times s_s$, where S_V denotes the surface area per unit volume of the entire surface system of the SiC phase and the Al phase combined. On a plane parallel to the facet at distance t, the area fraction covered by the Boolean model is

$$s_s = 1 - \exp(-\lambda\pi(R^2 - t^2)).$$

By integration, it follows that the volume fraction p of the balls centred at the Cox process described above is given by

$$\begin{aligned} p &= S_V \int_{-R}^{R} (1 - \exp(-\lambda\pi(R^2 - t^2))\mathrm{d}t \\ &= S_V \left(2R - \exp(-\lambda\pi R^2) \frac{\sqrt{\pi}\,\mathrm{erfi}(\sqrt{\pi\lambda}x)}{\sqrt{\pi\lambda}} \Big|_0^R \right) \end{aligned}$$

By erfi we denote the complex error function. For low values of p, it is often sufficient to use the approximation

$$p \approx S_V \frac{4}{3}\pi R^3 \lambda.$$

This approximation assumes that the intensity λ of the balls on the facet system is so low that the probability that two balls intersect at all is negligible.

3.8 Discussion and Conclusion

In this sections we will discuss alternatives to the chosen statistics and models and analyse the quality of the proposed model.

3.8.1 Model fit

SiC **particles**

A comparison of the statistics of the random Laguerre tessellation used to model the SiC particles and the statistics on the image $f_{conv,resc}$ is displayed in Table 3.3. The statistics for the model were based on 10000 Laguerre cells. With the exception of $v_q(sf_1)$, the statistics of the fitted model had the smallest error for the statistics of sf_1. The shape factor sf_3 is not reproduced well by the fitted parameters.

In order to get a notion of the quality of the model fit of the SiC particles that is independent from the statistics used in the fit, we used the chord length distribution, which was introduced in Chapter 1, as a measure of the agreement between model and data set. The input images were of the size $[0, 199] \times [0, 199] \times [0, 1177]$ voxels (data set) and $[0, 299] \times [0, 299] \times [0, 299]$ voxels (model) with resolution $15.145 \times 15.145 \times 15.145\ nm^3$. For this analysis, the SiC particles were not placed conditionally on the Al grain boundaries. The results are depicted in Figure 3.7 and show a good agreement between model and data set.

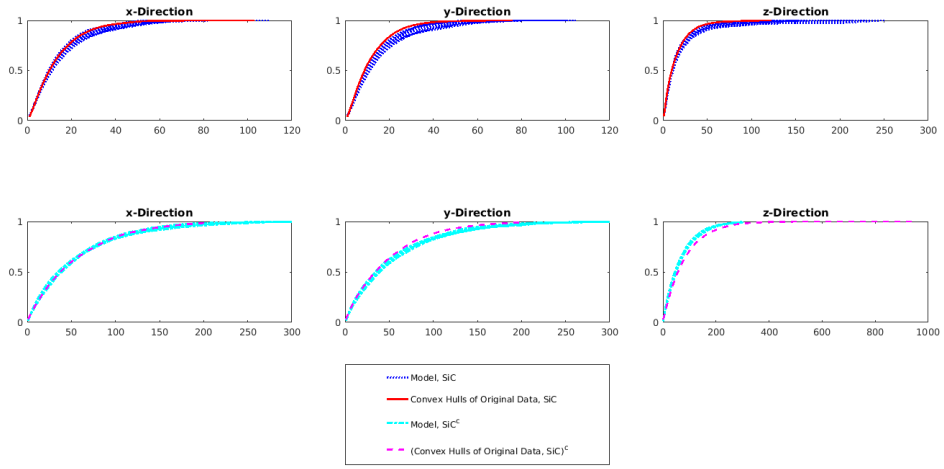


Figure 3.7: Plots of the chord length distributions for the three main axes, based on anisotropies modeled by elongation.

Al **grains**

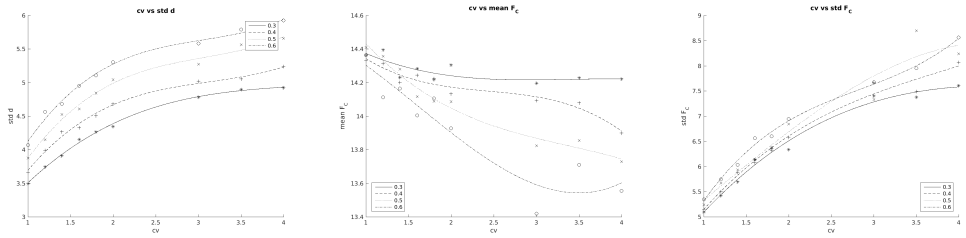
The fit based on simulations does give clear relationships between the parameters of the models used for simulation and the statistics obtained on the simulations only when outliers have been excluded. Some earlier fits based on a simulation with 10000 Laguerre cells and the parameters $cv_{Balls} = 1.456$, $V_V = 0.6$ for the ball packings used to generate the shapes of the SiC particles are displayed in Figure 3.8. In this case, there are obvious relationships visible for the relationship between cv_{Balls} and the statistics. We conclude that the simulation based modeling should be repeated for a large volume and that outlier treatment should be employed carefully. A visual comparison between a model based on the fits and a volume rendering of an FIB-SEM image of the *AMC17xxfine* is presented in Figure 3.9.

3.8.2 Orientation distribution for SiC anisotropy

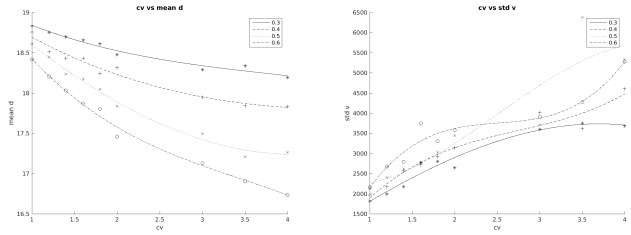
We incorporated the anisotropy of the SiC grains by a simple elongation factor. As introduced in Chapter 2, there are also other methods to analyse and model anisotropic grain distributions. One of them is the analysis of the main axis of orientation given by the moment of inertia tensor. For each normalized eigenvector, we calculate its representation in spherical coordinates, (ϕ, θ). In Figure 3.10 (a), we plotted the data points, together with a spherical density estimate of bandwidth $h = 0.1$ and a Gaussian kernel.

A typical model for directional distributions is the *Schladitz distribution*, [FRZ16, SPRB$^+$06] with density

$$f_{Schlad}(\theta, \phi) = \frac{1}{4\pi} \frac{\beta \sin\theta}{(1 + (\beta^2 - 1)\cos^2\theta)^{3/2}}$$



(a) cv vs. std($\bar{b}$) (b) cv vs. F_C (c) cv vs. std(F_C)



(d) cv vs. mean($\bar{b}$) (e) cv vs. std(V)

Figure 3.8: Data points and function fits for
$cv \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 3.0, 3.5, 4.0\}$ and $V_V \in \{0.3, 0.4, 0.5, 0.6\}$.

for $\theta \in [0, \pi)$ and $\phi \in [0, 2\pi)$ and $\beta \in \mathbb{R}^+$. This distribution is symmetrical around the direction $(0, 0, 1)$. The parameter β indicates whether the density is concentrated on the poles (for $\beta < 1$) or on the equator (for $\beta > 1$). For $\beta = 1$, the distribution is isotropic. The Schladitz distribution can be generalized to other symmetry axes (θ_0, ϕ_0) in spherical coordinates by

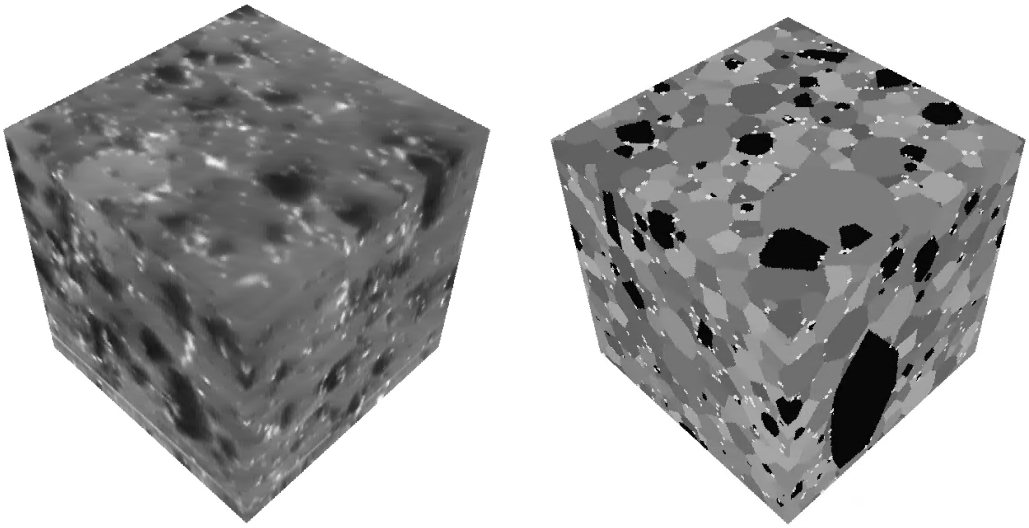
$$f_{Schlad,\theta_0,\phi_0}(\theta, \phi) = \frac{1}{4\pi} \frac{\beta \sin\theta}{(1 + (\beta^2 - 1)(\sin\theta_0 \sin\theta \cos(\phi_0 - \phi) + \cos\theta_0 \cos\theta)^{3/2}},$$

see [Zha13], Chapter 2.3. As laid out in detail in [FRZ16], this distribution is an angular central Gaussian distribution and can be further generalised to mixture distributions, defined by

$$f(\theta, \phi) = \sum_{k=1}^{n} \pi_k p(\theta, \phi, \alpha_k), \pi_k > 0, \sum \pi_k = 1.$$

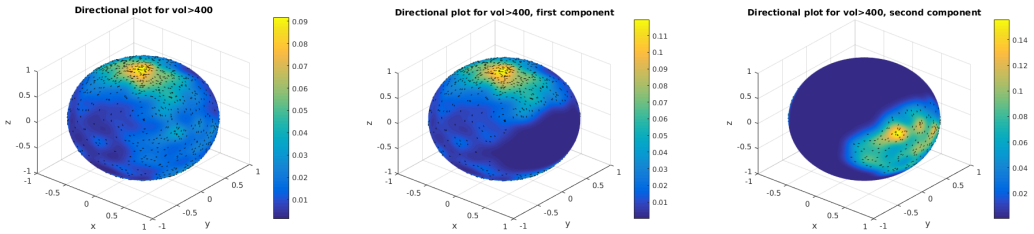
In order to estimate the parameters of the underlying distribution of the direction under the assumption of symmetry around $(0, 0, 1)$, we have to verify that ϕ is indeed uniformly distributed in $[0, 2\pi]$. A Kolmogorov-Smirnoff test rejects this hypothesis at $p < 0.001$. This means the observed directional distribution is not symmetric around the direction $(0, 0, 1)$. Based on the algorithms presented in [FRZ16], we unmixed the distribution of the directions, using the R code kindly provided by the authors. We identified two clusters:

30.48% of the directions lie in a cluster with $\hat{\beta}_1 = 0.5227$ and mean direction $\hat{\mathbf{x}}_1 = (0.9838, -0.1691, 0.0594)$, 69.52% of directions are in the cluster with $\hat{\beta}_2 = 0.37106$



(a) (b)

Figure 3.9: (a) Volume rendering of *AMC17xxfine* based on FIB-SEM images. (b) Volume rendering of the model based on $cv_{\mathrm{SiC},Balls} = 1.4505$, $V_{V_{\mathrm{SiC},Balls}} = 0.6$ and $cv_{\mathrm{Al},Balls} = 2.2103$, $V_{V_{\mathrm{Al},Balls}} = 0.6$. The volume fraction of Al_2Cu is 2.8%. The visualisation was generated in MAVI, [MAV05].



(a) (b) (c)

Figure 3.10: (a) Distribution of (θ, ϕ) with kernel density estimate. (b) First component. (c) Second component.

and mean direction $\hat{\mathbf{x}}_2 = (-0.0105, -0.0930, 0.9956)$. The clusters are depicted in Figure 3.10 (b,c). For these figures, we assigned each data point to a cluster according to its probability to lie in that cluster.

Based on the directional distribution, another possible model for the SiC phase would consist in rotating polygons of adequate shape so that their main axes are distributed according to the fitted distributions. This model is obtained by sampling from a large library of polygons according to the distribution of the shapefactor sf_1, for which a β-distribution with parameters $s_1 = 46.7228, s_2 = 9.7728$ was fitted, determining the main axis of the polygon and rotating it according to the fitted directional distribution. However, as this model does not incorporate the elongation of the particles, the model is too isotropic, as can be seen in the chord length distributions, displayed in Figure 3.11.

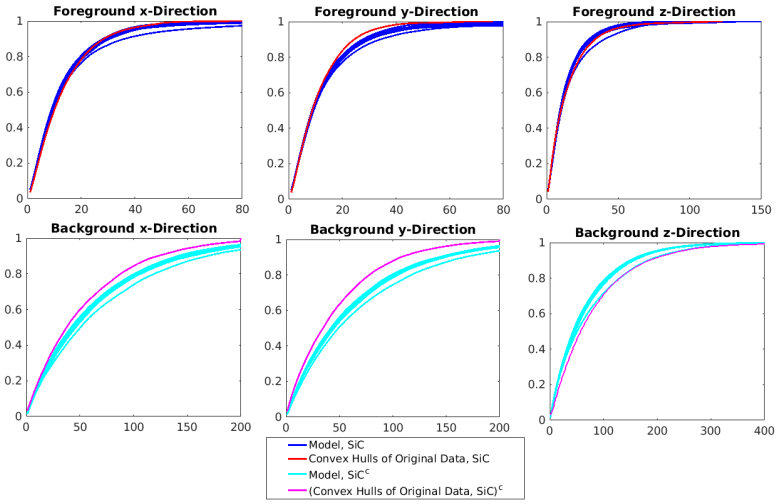


Figure 3.11: Plots of the chord length distributions for the three main axes, based on anisotropies modeled by Schladitz-distributed orientations, based on images of size $[400 \times 400 \times 400]$ voxels.

3.8.3 Other models for Al grains

We modeled the Al grains by cells of a random Laguerre tessellation, with polygonal SiC grains on the facets. This has the advantage that the model is fast to compute while still being a good approximation of the underlying structure. Also, the modeling of the Al_2Cu grain becomes very easy when we can assume that the particles lie on flats.

The disadvantage of the model is that it is physically not justified: The Al grains form around the SiC grains, and their surfaces are not flat. A physically more justified model might be achieved by a Johnson-Mehl tessellation, which is often used to model such grain structures.

Chapter 4

Stereology

In many cases, when investigating a structure embedded in the d-dimensional Euclidean space, there are only lower-dimensional samples available. The method of inferring higher or full-dimensional information from lower-dimensional samples is called *stereology*. It has its roots in geology, where the composition of rocks is of interest, which can often be investigated only on the surfaces or by drilling holes. Similarly, the standard method to investigate biological samples is to image slices of the sample under a light microscope and use stereological techniques to estimate cell volumes.

In the case of the material *AMC17xe*, with nominal SiC particle diameter of 3.0 μm, many SiC particles are too large to be included in a FIB-SEM sample. At the same time, they are too small to be resolved by a CT scan. Therefore, we propose to estimate the particle distribution based on 2-dimensional SEM slices. In the following, we will give a short introduction into the basic principles of stereology and show how the parameters μ and σ can be estimated when we assume that the volumes of the SiC particles are log-normally distributed with parameters μ and σ.

4.1 Classical stereology

Classical stereology, as introduced in [BVJ05] is interested in estimating the intrinsic volume densities of the form V_{V,k_V}, as introduced in Chapter 1 based on intrinsic volume densities at lower dimensions, here denoted by A_{A,k_A}, L_{L,k_L} and P_{P,k_P} with

$$A_{A,2_A} = \frac{\mathbb{E}(V_2(\Xi \cap W))}{V_2(W)}$$
$$A_{A,k_A} = \lim_{\alpha \to \infty} \frac{\mathbb{E}(V_k(\Xi \cap \alpha W))}{V_2(\alpha W)} \text{ for } k = 0, 1$$

for a 2-dimensional observation window W. The L_{L,k_L} and P_{P,k_P} are defined analogously.

The intrinsic volume densities have the advantage that they can be easily estimated

from lower-dimensional sections using Cavalieri's principle, under the assumption that either X is random and homogeneous or the method of sectioning is sufficiently randomised, see [BVJ05], Chapter 3, using the relationships:

$$\begin{aligned} V_{V,3} = A_{A,2} = L_{L,1} = P_{P,0}, \\ 2V_{V,2} = S_V = \frac{4}{\pi} B_A = 2I_L, \\ L_V = 2Q_A, \end{aligned}$$

where B_A denotes the boundary length per unit area and I_L denotes the number of intersection points per unit length of a line probe, which is a 1-dimensional observation window. Similarly, Q_A is the number of intersections of a line per unit area of a 2-dimensional observation window.

This means the volume fraction can be estimated on 2D sections by measuring the area fraction A_A of the phase X under investigation.

Note however that this classic method does not provide estimates one might be interested in when observing particles. For example, one might be interested in the mean number of particles per unit volume, N_V. However, this fraction cannot be easily estimated from sections alone: The probability of a particle to be hit by an observing section is proportional to its breadth H vertical to the cutting plane, see [BVJ05], Chapter 2. We obtain for the number of section profiles N_A:

$$N_A = N_V \mathbb{E}(H)$$

The breadth perpendicular to the cutting plane cannot be estimated within the plane without further assumptions on the volume distribution of the particles and their shapes.

4.2 Stereological estimation under a parametric model

When we model 3D particle systems, we can often assume that their volume distribution follows a parametric distribution. In this case, it is often possible to estimate the distribution parameters from 2D sections:

For example, on uniform and isotropic random sections, which are sections generated by planes with uniformly distributed reference point and isotropically distributed normals, see [BVJ05], Chapter 5, we can estimate the moments of the volume-weighted distribution, see [GJ85]. These moments can then be used to estimate the moments and the parameters of the (number-weighted) volume distribution, using the relationships established in [YLS98].

On the other hand, it is also possible to estimate the parameters based on simulations and a cost function based on two-dimensional section properties. An example of this approach was presented in [Lie14], where random Laguerre tessellations were fitted to sections with simulated annealing.

4.2.1 Estimation of the moments of the volume-weighted distribution

One approach to estimate the moments of the volume-weighted volume distribution was presented in [GJ85]. It works the following way:

A grid of sample points is overlayed with the observation window. When a sample point lies within a particle, this particle is sampled with a line probe, which means a line of random direction is laid through the sampling point. In case the particle is not convex, it might happen that there are more than one intersection chords. The length of the intersection with the particle, which contains the sampling point, is then denoted by l_0, all other intersection lengths are denoted by l_1, l_2, etc. For convex particles, there is only one intersection length l_0.

The volume-weighted mean volume is then $\mu_{1,v} = \frac{\pi}{3}\mathbb{E}(l_0^3 + 2\sum_{i>0} l_i^3)$. It is estimated by the sample mean. The second moment of the volume-weighted distribution can be estimated by randomly choosing two more points per sample point, which have to lie uniformly distributed within the same particle. If Δ denotes the area of the resulting triangle and a denotes the total area of the particle section, the second moment of the volume-weighted distribution is $\mu_{2,v} = 2\pi\mathbb{E}(a^2\Delta)$. Again, it is estimated by the sample mean. Under the assumption that the distribution of the particle volumes is parametric and known, the parameters of the number-weighted volume distribution can then be estimated based on the estimated parameters of the volume-weighted distribution. Under the assumption that the number-weighted volume distribution is a log-normal distribution with parameters μ and σ, the resulting estimates are

$$\hat{\mu} = 4\ln\hat{\mu}_{1,v} - 1.5\ln\hat{\mu}_{2,v}$$

and

$$\hat{\sigma} = \sqrt{\ln\left(\frac{\hat{\mu_{2,v}}}{\hat{\mu}_{1,v}}\right)},$$

see [YLS98].

4.2.2 Simulation-based parameter estimation

Another approach to estimate the parameters $\mu_1 \ldots, \mu_n$ of the distribution of three-dimensional particles based on two-dimensional samples is the direct optimisation based on simulation, using a cost function

$$c(\mu_1, \ldots \mu_n) = \sum_j \left(\frac{\hat{m}_j(\mu_1, \ldots \mu_n) - \hat{m}_j}{\hat{m}_j}\right)^2,$$

where the $\hat{m}_j$ are statistics on the two-dimensional samples and $\hat{m}_j(\mu_1, \ldots \mu_n)$ are statistics on two-dimensional sections of the model generated with parameters $\mu_1, \ldots \mu_n$. As mentioned above, in [Lie14] random Laguerre tessellations were fitted to sections using *simulated annealing*, which is a stochastic optimisation method that accepts

non-optimal solutions depending on the current "temperature" t, which decreases over the course of the optimisation, see [KGV83]. The algorithm has the advantage that local minima can be easily overcome, but it depends strongly on the temperature decrease. Additionally, due to the stochastic nature of the optimisation method, it might take many iterations (and therefore also many simulations of the model) to converge. In the following, we will present the *Nelder-Mead* algorithm, which has the advantage that it also easily overcomes local minima, is easy to implement and does not require a lot of parameter-tuning. The disadvantage of the Nelder-Mead algorithm is its slow convergence.

Nelder-Mead algorithm

The *Nelder-Mead algorithm* is a direct search algorithm that does not need any derivatives. It is based on reflecting, expanding and contracting a simplex of $(N+1)$ nodes in the parameter space (of dimension N). This is done in a fashion that for each iteration step, the simplex node θ_i with the worst (that is, highest) cost function value $c(\theta_i)$ is replaced by a better node. The original method has the parameters α, β and γ. It works the following way, see [NM65]:

1. Generate the initial simplex in the parameter space, with vertices $V = \{\theta_0, \dots \theta_N\}$. This can be done by choosing one vertex $\theta_0 \in \mathbb{R}^N$ and varying this initial vector in each component independently. GoTo 2.

2. Now, the function values $c(\theta_i)$ at the vertices are evaluated and the vertices are sorted in a fashion that $c(\theta_0) \leq c(\theta_1) \leq \ldots \leq c(\theta_N)$. GoTo 3.

3. Among the N best values $\theta_0, \dots, \theta_{N-1}$, the mean θ_m is generated. GoTo 4.

4. Based on this mean θ_m, the worst vertex θ_N is reflected using the reflection parameter α, to generate the reflected vertex

$$\theta_r = (1+\alpha)\theta_m - \alpha\theta_N.$$

   GoTo 5.

5. If the reflected vertex θ_r is even better than the best value of the former simplex, θ_0,

   - we try to further improve the value by *expanding* the vertex even further, generating the expanded vertex

$$\theta_e = (1+\gamma)\theta_m - \gamma\theta_N.$$

     The vertex set is then updated to

$$(V \setminus \theta_N) \cup \arg\min \{c(\theta_e), c(\theta_r)\}.$$

     GoTo 2.
   - Else GoTo 6.

6. However, if the reflected vertex θ_r is only better than θ_{N-1},

- update the vertex set according to

$$(V \setminus \theta_N) \cup \theta_r$$

GoTo2.

- Else GoTo7.

7. If θ_r is not better than θ_{N-1},

- generate the *contracted* vertex

$$\theta_c = \beta\theta_h + (1-\beta)\theta_m,$$

with

$$\theta_h = \arg\min\{c(\theta_N), c(\theta_r)\}.$$

GoTo 8.

8. If $c(\theta_c) < c(\theta_N)$,

- update the vertex set to

$$(V \setminus \theta_N) \cup \theta_c.$$

GoTo 2.

- Else: *contract* the simplex by replacing all vertices $\theta_i \in V \setminus \theta_0$ (that is, all vertices apart from the best) by $\sigma\theta_0 + (1-\sigma)\theta_i$. The new vertex set is then

$$\{\theta_0, \sigma\theta_0 + (1-\sigma)\theta_1, \ldots, \sigma\theta_0 + (1-\sigma)\theta_N\}.$$

GoTo 2.

Convergence is reached when the pairwise distances between the vertices of the parameter simplices are all smaller than a threshold ϵ. Following [NM65], the parameters are chosen as $\alpha = 1$, $\beta = 0.5$ and $\gamma = 2$. Stereological estimation using the Nelder-Mead algorithm was for example done by [RAB+12] for granular structures.

4.3 Stereological estimation in the non-parametric case

In case that it is not reasonable to assume a parametric model distribution of particles, stereological estimation can obtained by solving the *stereological unfolding problem*, which relates the size distribution of particles in 3D to the size distribution observed in 2D, under the assumption that the particles are homogeneously and isotropically distributed within the sample, see [OM00]. Typical "sizes" under consideration are the (equivalent spherical) diameter or the mean breadth. The following introduction into this approach to stereology is based on the very detailed Chapters 6 to 8 of [OM00]:

The easiest stereological problem is the so-called *Wicksell problem*, which considers spherical particles and their diameter distribution $F_V(u)$. Then, the number of 2D

section profiles N_A and the distribution of the circle diameters $F_A(u)$ are related to the number of spheres in 3D, N_V and their size distribution via:

$$N_A(1 - F_A(s)) = N_V \int_0^\infty u(1 - G_u(s))\mathrm{d}F_V(u)$$

with the *section circle diameter distribution function* $G_u(s) = 1 - \sqrt{1 - s^2/u^2}$ under the condition that the random sphere diameter is u, see [OM00]. This equation can be solved analytically, but for real-life applications, it is generally solved numerically. This numerical solution is based on the approximation of the stereological integral equation by a linear equation system

$$y = P\theta,$$

precisely

$$y_k = \sum_{i=1}^{n} p_{ki}\theta_i, \quad k = 1, \ldots, n$$

under the assumption that the sizes U are discrete random variables with values u_i and $F_A(s)$ is approximated by a piecewise constant function with value a_i for $(i-1)\Delta < u \leq i\Delta, \quad k = 1, \ldots, n$ with interval width Δ. The kernel $p = u(1 - G_u(s)) = \sqrt{u^2 - s^2}$ is approximated by $p_{ki} = p(i\Delta, (k-1)\Delta) - p(i\Delta, k\Delta), i \leq k$. The approximation equation is then achieved using $\theta_i = N_V(a_i - a_{i-1})$ and $y_k = N_A(F_A(s_k) - F_A(s_{k-1}))$.

For easy problems such as the Wicksell problem, p_{ki} can be determined analytically, while for more complicated problems it has to be calculated based on simulations.

These more complicated problems include for example the analysis of spheroidal or polygonal particles. In these cases, a single size parameter is usually not sufficient to characterize the structure. Therefore, the *size-shape distributions* $F_V(u, v) = \mathbb{P}(U \leq u, V \leq v)$ and $F_A(s, t) = \mathbb{P}(S \leq s, T \leq t)$ are taken into consideration, with U and S denoting the size in 3D or in 2D sections, respectively, and V and T denoting the shapes in 3D and in 2D sections. Again, there are various notions of "shape", typical examples are the relation of longest to shortest diameter or the number of vertices (in the case of polygons). The size s and shape t in 2D and size u and shape v in 3D are related via

$$N_A(1 - F_A(\infty, t) - F_A(s, 1) + F_A(s, t)) = N_V \int_s^\infty \int_t^1 p(u, v, s, t)\mathrm{d}F_V(u, v),$$

with $p(u, v, s, t) = u\mathrm{Pr}(S > s, T > t | U = u, V = v, X_0 \uparrow E)$, where $X_0 \uparrow E$ means that the structure was intersected by the observation plane. Again, when assuming discrete sizes u_i and shapes v_j, the stereological integral equation can be approximated by a linear equation system of the form

$$y_{kl} = \sum_{i=k}^{m} \sum_{j=l}^{n} p_{ijkl}\theta_{ij}, \quad k = 1, \ldots, m;\ l = 1, \ldots, n.$$

This equation can be simplified by using a logarithmic discretisation, so that $u_i = a^i$ and $s_k = a^k$. Then, we obtain the equation

$$y_{kl} = \sum_{i=k}^{\infty} \sum_{j=l}^{n} \pi_{j,(k-i),l}\Theta_{ij}, \quad k = \ldots, -1, 0, 1, \ldots;\ l = 1, \ldots, n$$

(a) (b) (c)

Figure 4.1: (a): Section of a model containing SiC particles. (b): Labelled SEM image of *AMC17xe*, perpendicular to the extrusion direction, particles are convexified. (c): Labelled SEM image of *AMC17xfine*, perpendicular to the extrusion direction, particles are convexified. Displayed is a 300×300 pixel region of the images. The resolution of (a) is 15.145×15.145 $(nm)^2$, the resolution of (b) and (c) is 25×25 $(nm)^2$. The images were generated with MAVI, [MAV05].

with $\Theta_{ij} = a^i \bar{b}_j \theta_{ij}$ with the mean breath $\bar{b}_j$ of particle j, and $\pi_{jkl} = \frac{\bar{\bar{b}}}{\bar{b}_j} \Pr(s_{k-1} \leq S < s_k, t_{l-1} \leq T < t_l | U = 1, V = v_j, X_0 \uparrow E)$.

In order to solve this linear equation system one usually uses the EM algorithm proposed in [SJW90]. Under the assumption that the particle centres are Poisson distributed with constant intensity, the algorithm is based on the following steps:

$$\hat{P}_{j,k-i,l} = \frac{\pi_{j,k-i,l}}{\sum_{i=k}^{\infty} \sum_{j=1}^{n} \pi_{j,k-i,l} \Theta_{ij}^{\lambda}} \Theta_{ij}^{\lambda} y_{kl} \text{ (expectation)}$$

$$\Theta_{ij}^{\lambda+1} = \frac{\sum_{l=1}^{n} \sum_{k=-\infty}^{i} \hat{P}_{j,k-i,l}}{\sum_{k=-\infty}^{0} \sum_{k=1}^{n} \pi_{jkl}} \text{ (maximisation)}$$

with $i = \ldots, -1, 0, 1, \ldots$ and $j = 1, \ldots, n$. The initial value Θ_{ij}^0 is chosen as $\Theta_{ij}^0 = y_{kl}$. Both steps can be combined to one equation:

$$\Theta_{ij}^{\lambda+1} = \frac{\Theta_{ij}^{\lambda}}{q_j} \sum_{l=1}^{n} \sum_{k=-\infty}^{i} \frac{\pi_{j,k-i,l} y_{kl}}{\sum_{i=k}^{\infty} \sum_{j=1}^{n} \pi_{j,k-i,l} \Theta_{ij}^{\lambda}}$$

4.4 Application: Stereological estimation for samples of AMCxe and AMCxfine

In the following, we will present a method to estimate the parameters μ and σ of the volume distribution of particles under the assumption that the volume is log-normally distributed. The method is applied to labelled SEM images of the materials *AMC17xe* and *AMC17xfine* with convexified labels, and to a random plane sample from an SiC model generated using the parameters fitted in Chapter 3 of

size $[0, 400] \times [0, 400]$ pixels of resolution 15.145×15.145 $(nm)^2$. However, for this test case, the SiC particles were not placed conditional on a Laguerre tessellation in order to increase the speed of simulations.

4.4.1 Estimating the volume fraction using classic stereology

Based on the results of classic stereology, we can estimate the volume fraction V_V of the particle phase by the area fraction A_A, the length fraction L_L for a set of test lines, or the point fraction P_P for a set of test points.

This means the volume fraction of the SiC phase can be estimated on the images by the fraction of pixels within the observation window that belong to the SiC phase.

The area fraction in the section is 0.1533. The volume fraction of the model is 0.1683. This sets the relative error at 8.9%.

4.4.2 Estimation of parameters based on volume-weighted moments

A preliminary study on $N = 2500$ balls with log-normally distributed volumes shows that the estimation of the parameters μ and σ based on the estimation of volume-weighted moments is not applicable in our case, as μ is overestimated and σ is underestimated for $cv > 1.0$. The reason lies probably in the high coefficient of variation and the relatively small observation window (Luis Cruz-Orive, personal communication, September 2016). In case of *AMC17xxfine*, we fitted a volume distribution with $cv = 13.1$ to the data. It is to be expected that the related materials might have a similarly high coefficient of variation.

4.4.3 Estimation of parameters based on simulations

Since the direct estimation of the parameters via the moments of the volume-weighted distribution is not feasible, we propose to estimate the parameters of the volume distribution based on simulations.

This is done in the following way: Based on 2D images, we can get information on the moments of the area distribution. Additionally, the area fraction A_A is an unbiased estimate of the volume fraction V_V. We have shown above that the error introduced by using this estimate of V_V is only small. Based on the model for *AMC17xxfine*, we assume the 3D volume distribution to be log-normal with parameters μ, σ. As a first step, we will additionally assume the shapes of the SiC particles in *AMC17xfine* and AMCxe to be similarly distributed as the SiC particles within the *AMC17xxfine*. For the optimisation, we use the cost function

$$c(\mu_i, \sigma_i) = \left(\frac{A_{A,i} - A_A}{A_A} \right)^2 + \sum_{j=1}^{2} \left(\frac{\hat{m}_i^j - \hat{m}^j}{\hat{m}^j} \right)^2 ,$$

where $\hat{m}^j$ refers to the jth moment of the area distribution of the data set and $\hat{m}_i^j$ is the jth moment of a cut through the model generated by parameters μ_i, σ_i. By $A_{A,i}$ we denote the area fraction of a section through the model generated by parameters μ_i, σ_i, while A_A is the area fraction of the 2-D sample. To evaluate the cost function at μ_i, σ_i, we generate a model based on the Laguerre cells fitted in Chapter 3 with $\text{Logn}(\mu_i, \sigma_i)$ distributed sizes. This model is then randomly cut and the generated 2D area distribution is evaluated.

There are two aspects of variance we have to account for:

1. Sample variance due to the sampling of a random model with the parameter set μ, σ
2. Stereological variance due to randomisation in the location of the section plane.

The method to overcome these problems is to repeatedly sample the model, and to use several independent stereological samples. Especially the latter is generally not possible, as in many real-world applications there are only few sections available. Studies done in [Dob10] regarding the use of the Nelder-Mead algorithm for parameter estimation of several spatial germ-grain processes suggest that the number of samples generated in each update step of the Nelder-Mead algorithm should be linear in the iteration number k. In [Dob10], the relationship

$$N(k) = \text{round}\left(\frac{17}{49}k + \frac{130}{49}\right)$$

is used.

Due to the fact that the variance introduced due stereology is very large, this kind of variance reduction becomes unnecessary: For example, for two random cuts through the same model we observe a deviation in the average area of section profiles of 22.8%, the standard deviations differ by 9.7%.

Minimisation based on grid search

Based on an initial estimate $(\hat{\mu}_0, \hat{\sigma}_0)$ using the volume-weighted moments of the volume distribution, we generate a random grid within $(0, a] \times (0, \sqrt{2a}]$ with $a = \hat{\mu}_0 + 0.5\hat{\sigma}_0^2$, which is based on the fact that the mean volume $\exp(\hat{\mu}_0 + 0.5\hat{\sigma}_0^2)$ is generally overestimated by the volume-weighted estimation. Based on a grid with 100 grid nodes, we can identify the best parameters $\hat{\mu}_{Grid}, \hat{\sigma}_{Grid}$ by simulating within a window chosen so that the expected number of particles lying in the window is at least 5000.

The precision of the grid search method for stereological estimation of particle sizes was evaluated by estimating the parameters on an artificially generated model. All parameters of the log-normal distribution refer to volumes measured in voxels. The spacing was 15.145×15.145 (nm^2), respectively $15.145 \times 15.145 \times 15.145$ (nm^3). The models for the function evaluation were the standard *RSA*. The results are displayed in Table 4.1.

Method	μ	σ	$\hat{\mu}_{Grid}$	$\hat{\sigma}_{Grid}$	$c(\hat{\mu}_{Grid}, \hat{\sigma}_{Grid})$	$d(\hat{\mu}_{Grid})$	$d(\hat{\sigma}_{Grid})$
RSA	4.5210	2.2688	5.0849	2.1177	1.1691	12.47%	−6.66%

Table 4.1: Estimation of the parameters μ and σ from one section through an artificial model perpendicular to elongation direction, based on grid search, where d denotes the deviation.

Method	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	$c(\hat{\mu}, \hat{\sigma})$	$d(\hat{\mu})$	$d(\hat{\sigma})$
RSA+move	4.5210	2.2688	5.13612	2.07294	0.0204991	13.61%	−8.63%
RSA	4.5210	2.2688	5.03045	2.12206	1.08487	11.27%	−6.47%

Table 4.2: Estimation of the parameters μ and σ from one section through an artificial model perpendicular to elongation direction, based on the Nelder-Mead simulation with approximately 1000 particles per model. The deviation is denoted by d.

Minimisation based on Nelder-Mead

In order to improve the quality of the fit, we propose to use the minimum $(\hat{\mu}_{Grid}, \hat{\sigma}_{Grid})$, found by grid search as one vertex of the starting vertex of a Nelder-Mead algorithm. The other vertices are chosen randomly as $(\hat{\mu}_{Grid} \pm 1, \hat{\sigma}_{Grid})$ and $(\hat{\mu}_{Grid}, \hat{\sigma}_{Grid} \pm 1)$, with equal probability for the positive and the negative sign. In order to increase the speed of the estimation, we abstain from increasing the number of generated samples. Instead, we chose the sampling window's volume so that the expected number of particles lying in the window is at least 1000.

The precision of the Nelder-Mead method for stereological estimation of particle sizes was evaluated by estimating the parameters on an artificially generated model. The models for the function evaluation were either the standard *RSA* or the *RSA+move* used for generating the artificial sample. The results are displayed in Table 4.2. After convergence at (μ^*, σ^*) we restarted the algorithm, one vertex of the starting simplex was chosen as (μ^*, σ^*), the others randomly as $(\mu^* \pm 1, \sigma^*)$ and $(\mu^*, \sigma^* \pm 1)$ as above. We stopped the algorithm if the element-wise difference between consecutive optimal solutions was less than 0.05. We set $\epsilon = 0.001$ as convergence criterion.

Parameters of *AMC17xfine*

Based on labelled images of a resolution 25×25 $(nm)^2$ with size 1024×883 pixels, taken perpendicular to the extrusion direction, we estimated the values of the parameters for *AMC17xfine*, using the convex hulls of the labels. The results based on the three methods presented above are displayed in Table 4.3. The estimated volume fraction is $\widehat{V_V} = 0.252387$. Note that the displayed values refer to voxels of volume $25 \times 25 \times 25$ (nm^3). The average values from the estimation are $\hat{\mu} = 1.9514$ and $\hat{\sigma} = 2.8222$. At a voxel width of 25 nm, the equivalent spherical diameter of a SiC particle within the sample is therefore 0.22 μm.

Method	$\hat{\mu}$	$\hat{\sigma}$	$c(\hat{\mu}, \hat{\sigma})$
NM *RSA+move*	1.16303	3.31087	0.0891154
NM *RSA*	3.40018	2.38111	1.00008
Grid *RSA*	1.29099	2.7746	0.15305

Table 4.3: Estimation of the parameters μ and σ for *AMC17xfine*. "NM" refers to estimations obtained by a Nelder-Mead algorithm based on 1000 simulated particles per iteration.

Method	$\hat{\mu}$	$\hat{\sigma}$	$c(\hat{\mu}, \hat{\sigma})$
NM *RSA+move**	7.89849	1.49918	0.000082
NM *RSA*	8.04534	1.43506	1.00006
Grid *RSA*	8.14647	1.35387	0.0158811

Table 4.4: Estimation of the parameters μ and σ for *AMC17xe*. "NM" refers to estimations obtained by a Nelder-Mead algorithm based on 1000 simulated particles per iteration.

Parameters of *AMC17xe*

The parameters for *AMC17xe*, using the three methods presented above, are displayed in Table 4.4. The estimated volume fraction is $\widehat{V_V} = 0.300435$. Again, the displayed values refer to voxels of volume $25 \times 25 \times 25$ (nm^3) estimated on an image of size 1024×883 pixels of the convex hulls of the particles. The average values from the estimation are $\hat{\mu} = 8.0301$ and $\hat{\sigma} = 1.42937$. At a voxel width of 25 nm, the equivalent spherical diameter of a SiC particle within the sample is therefore 0.63 μm.

4.5 Discussion

4.5.1 Quality of the estimation

The stereological estimation based on the moments of the area sections of the SiC particles gives good results when the number of particles within the simulation is chosen large enough. This is especially evident when comparing the results obtained by a simple random grid search designed so that the expected number of particles in the simulation is at least 5000 to the Nelder-Mead optimisation where the expected number of particles per simulation was at least 1000. However, even in the latter case the equivalent spherical diameter is estimated quite well with an error of 6.5%.

4.5.2 Comparison with estimates on images

The estimated equivalent spherical diameters deviate largely from the nominal diameters (of 0.7 μm and 3.0 μm for *AMC17xfine* and *AMC17xe*) given by the manufacturer. For *AMC17xfine* it is possible to roughly estimate the equivalent spherical

diameters based on labelled FIB-SEM images. For the labelled images, we obtain an equivalent spherical diameter of 0.3355 μm, on the convexified labels the equivalent spherical diameter is 0.4144 μm. Our values deviate by 34.4% from the latter.

Chapter 5

Results and Conclusions

5.1 Results

This thesis was preoccupied with the micro-structure of various SiC-particle reinforced Al-alloys. In the beginning, we presented various methods how to binarise and label complicated data stemming from FIB-SEM and SEM imaging.

In the following chapter, we analysed the materials' distributions based on three-dimensional image data: For the sample taken from *AMC17xxfine*, the sample containing the smallest SiC particles, we find that the particles' elongation along the extrusion direction depends on the particle volume. After rescaling to isotropy, we find that the volume of the SiC particles is log-normally distributed. Based on the shape distribution of the rescaled SiC particles, we can fit a Laguerre tessellation to the data set. By isotropic and anisotropic rescaling we can obtain an adequate model for the particles. We additionally showed how to arrange them spatially to obtain a good model for the SiC phase of the material. Additionally, we propose a Laguerre tessellation to model the Al grains and a Cox-process base to model the centres of the Al_2Cu precipitations. The Al_2Cu precipitations were then modelled by placing overlapping spheres on the Cox process. For each method we showed how to estimate their parameters.

Additionally, we presented a method to estimate the parameters of the SiC particles' volume distribution from two-dimensional sections. Under the assumption that the particle shapes used to model *AMC17xxfine* pose a sufficient approximation of the particle shapes of other samples, we estimated the parameters of the volume distribution.

5.2 Comparison with manufacturer's claims

The manufacturer denotes the nominal equivalent SiC particle diameters as 0.3 μm (for *AMC17xxfine*), 0.7 μm (for *AMC17xfine*) and 3.0 μm (for *AMC17xe*). Our estimates for the equivalent spherical diameter deviate largely from these values:

Based on the analysis of FIB-SEM images of resolution $15.145 \times 15.145 \times 50$ $(nm)^3$, we obtain the equivalent spherical diameters 0.20 μm (for *AMC17xxfine*) and 0.4144 μm (for *AMC17xfine*, at resolution $16.5 \times 16.5 \times 50$ $(nm)^3$) for the convex hulls of the SiC particles. Based on stereological estimation, we obtain 0.22 μm (for *AMC17xfine*) and 0.63 μm (for *AMC17xe*) for the convex hulls of the SiC particles.

The manufacturer does not give any information on how the equivalent spherical diameters were estimated. The observed difference could however stem from different resolutions: The largest particles within the *AMC17xe* data set can be observed at quite low resolutions, resulting in a large estimate of the equivalent spherical diameter. When higher resolutions are used, there are many small particles within the observation window, thereby reducing the mean of the volume distribution.

Another reason for the discrepancy could be caused if the manufacturer used a sieving method. In this case the "nominal diameter" could refer to the mesh size of the sieve.

5.3 Future Work

5.3.1 Improvements of the model

For this work, all analysis of the particles' distributions were only conducted on sub-volumes of homogeneous particle intensity. However, on a larger scale, there are long, almost particle-free zones that are elongated along the extrusion direction of the sample, see Figure 5.1. They can be observed on SEM images at a lower resolution as bands or circular areas. As they are structures on a larger scale, these longitudinal gaps are only well-observable in 3D for samples with large particles, namely *AMC17xe*. Future works should integrate these changes of particle intensity into the model.

Similarly, our modeling of the Al grains uses polygonal grains as a crude approximation. However, when observed in SEM images, often these grains are not convex and have rough surfaces. This means that maybe a Johnson-Mehl tessellation would be more appropriate to model the Al grains. Then, it would be advisable to first model the SiC phase and then let the tessellation "grow" around it. Of course, this process would highly increase computation time. As of now, we assume that the Al_2Cu inclusions are generated by a Cox process on the boundaries of the Al and the SiC phases. However, conditional on lying on these surfaces, this Cox process is a Poisson point process. It could be worthwhile to investigate correlations with particle orientations.

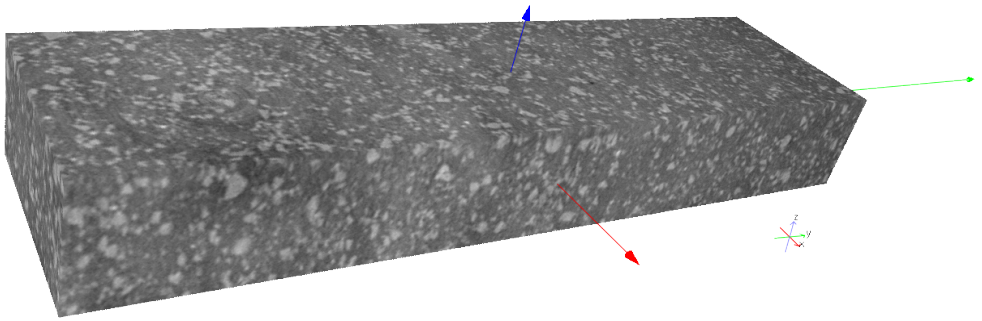
Future investigation could also analyse correlations between grain orientation and SiC particles. Additionally, the parameters of the Al grains are not part of the stereological estimation yet. This estimation could be incorporated into future work.

We found that the shape factors we used to fit the AMCxxfine's SiC particles' shapes are generally viable for the parameter estimation based on sections. However, it would be good to incorporate an estimation of the particle's shape distribution and

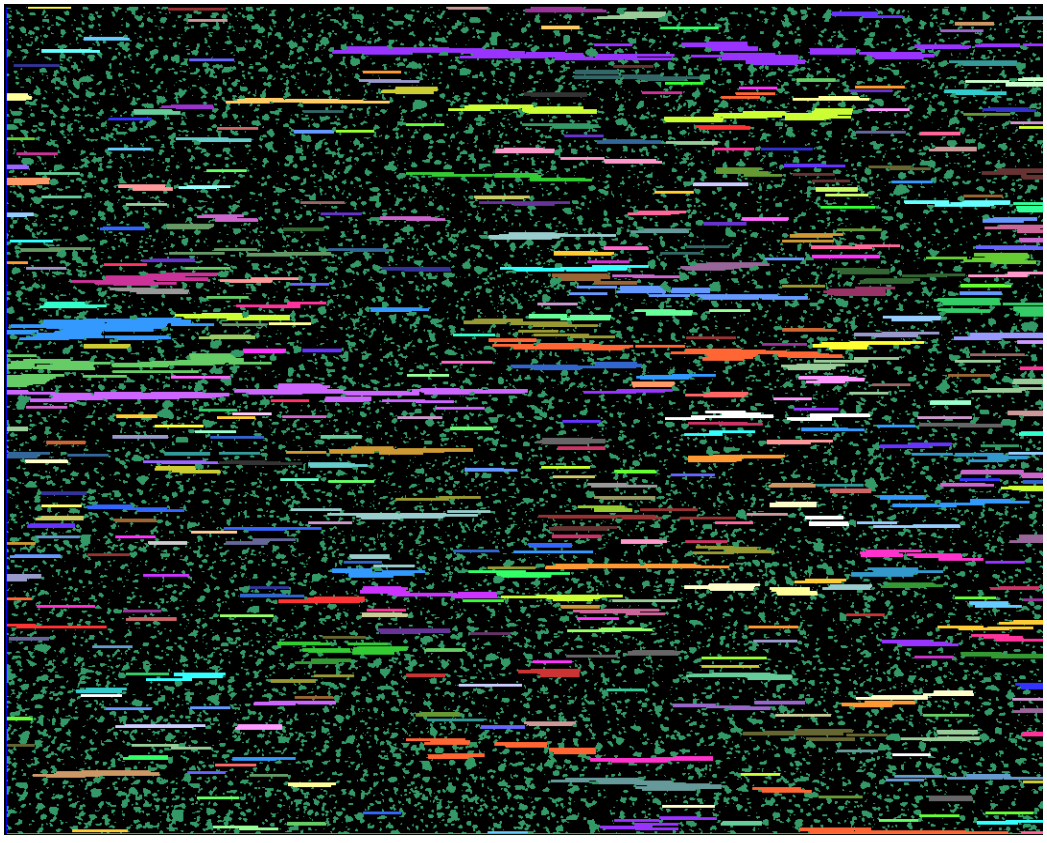
elongation classes from 2D sections, as well. This might be obtained using the stereological unfolding approach presented in Chapter 4.

5.3.2 Predicting material failure

The models presented in this work can be used as input for FEM simulations, similar to the work presented in [CGW04]. The advantage of our model compared to using binarised FIB-SEM images is its adaptability to other material parameters and the fact that an arbitrary number of virtual samples can be generated instead of costly FIB-SEM imaging.



(a)



(b)

Figure 5.1: (a) Volume rendering of an synchrotron image of *AMC17xe* obtained at ESRF, Grenoble. (b) Longitudinal gaps parallel to extrusion direction, labelled on an synchrotron image of *AMC17xe* obtained at ESRF, Grenoble. The images were generated with MAVI, [MAV05].

Appendix A

Tables

A.1 Parameters of the curves used to fit the rescaled SiC particles' shapes

In the following tables, we present the parameters for

$$f^j_{x,y}(cv_{Balls}) = \frac{p^j_{1,x,y} \cdot cv_{Balls} + p^j_{2,x,y}}{cv_{Balls} + q^j_{1,x,y}}$$

used to fit the relationships between statistics of the shape factors sf_1, sf_2 and sf_3 and the parameters (cv_{Balls}, V_V) used to generate force-biased ball packings for generating random Laguerre tessellations.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.3}$	0.69409	0.52256	0.81866
$p_{2,\text{mean},0.3}$	2.6205	2.1959	3.1681
$q_{1,\text{mean},0.3}$	3.0028	2.7784	3.385
$adjR^2_{\text{mean},0.3}$	0.99963	0.99971	0.99961
$p_{1,\text{median},0.3}$	0.69592	0.52424	0.83048
$p_{2,\text{median},0.3}$	2.1649	1.785	2.3341
$q_{1,\text{median},0.3}$	2.4437	2.1996	2.4736
$adjR^2_{\text{median},0.3}$	0.99966	0.9998	1
$p_{1,\text{stddev},0.3}$	0.089576	0.12291	0.066302
$p_{2,\text{stddev},0.3}$	0.00022901	-0.0039653	0.033104
$q_{1,\text{stddev},0.3}$	1.0434	0.8294	3.1288
$adjR^2_{\text{stddev},0.3}$	0.99234	0.99543	0.99472
$p_{1,v_Q,0.3}$	0.13989	0.24771	0.077552
$p_{2,v_Q,0.3}$	-0.066475	-0.10463	-0.0097686
$q_{1,v_Q,0.3}$	0.02393	0.16627	0.93184
$adjR^2_{v_Q,0.3}$	0.99836	0.99999	0.99

Table A.1: Parameters of the fitted curves for $V_V = 0.3$ with their adjusted R^2 goodness-of-fit statistic. Based on $cv_{Balls} = \{1, 2, 5, 10\}$.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.4}$	0.71144	0.54804	0.83425
$p_{2,\text{mean},0.4}$	2.091	1.8171	2.4119
$q_{1,\text{mean},0.4}$	2.3834	2.286	2.5705
$adjR^2_{\text{mean},0.4}$	0.9998	0.99986	0.99992
$p_{1,\text{median},0.4}$	0.69828	0.52731	0.83333
$p_{2,\text{median},0.4}$	2.3332	1.9357	2.4599
$q_{1,\text{median},0.4}$	2.6472	2.4075	2.6131
$adjR^2_{\text{median},0.4}$	0.99754	0.99813	0.99884
$p_{1,\text{stddev},0.4}$	0.086306	0.11859	0.05253
$p_{2,\text{stddev},0.4}$	-0.033748	-0.050706	-0.010893
$q_{1,\text{stddev},0.4}$	0.1591	0.0067146	0.66642
$adjR^2_{\text{stddev},0.4}$	0.99919	0.99746	0.99998
$p_{1,v_Q,0.4}$	0.15095	0.25958	0.073919
$p_{2,v_Q,0.4}$	-0.087024	-0.14369	-0.035588
$q_{1,v_Q,0.4}$	-0.1307	-0.087387	0.062008
$adjR^2_{v_Q,0.4}$	0.97798	0.97991	0.99446

Table A.2: Parameters of the fitted curves for $V_V = 0.4$ with their adjusted R^2 goodness-of-fit statistic. Based on $cv_{Balls} = \{1, 2, 5, 10\}$.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.5}$	0.69968	0.53258	0.82628
$p_{2,\text{mean},0.5}$	2.4818	2.1686	2.9198
$q_{1,\text{mean},0.5}$	2.847	2.7609	3.1227
$adjR^2_{\text{mean},0.5}$	1	1	0.99999
$p_{1,\text{median},0.5}$	0.68036	0.50329	0.82087
$p_{2,\text{median},0.5}$	3.0058	2.5004	3.459
$q_{1,\text{median},0.5}$	3.4389	3.161	3.6971
$adjR^2_{\text{median},0.5}$	0.99933	0.99926	0.99916
$p_{1,\text{stddev},0.5}$	0.094543	0.13006	0.057691
$p_{2,\text{stddev},0.5}$	-0.037081	-0.053965	-0.014809
$q_{1,\text{stddev},0.5}$	0.21722	0.082579	0.63746
$adjR^2_{\text{stddev},0.5}$	0.998	0.99632	0.99912
$p_{1,v_Q,0.5}$	0.1764	0.31239	0.088685
$p_{2,v_Q,0.5}$	-0.081912	-0.1227	-0.030251
$q_{1,v_Q,0.5}$	0.2362	0.42817	0.54634
$adjR^2_{v_Q,0.5}$	0.98488	0.99697	0.99989

Table A.3: Parameters of the fitted curves for $V_V = 0.5$ with their adjusted R^2 goodness-of-fit statistic. Based on $cv_{Balls} = \{1, 2, 5, 10\}$.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.6}$	0.6911	0.5194	0.81774
$p_{2,\text{mean},0.6}$	2.6659	2.392	3.3245
$q_{1,\text{mean},0.6}$	3.0656	3.0616	3.5605
$adjR^2_{\text{mean},0.6}$	0.99992	0.9999	0.99991
$p_{1,\text{median},0.6}$	0.66983	0.48676	0.81398
$p_{2,\text{median},0.6}$	3.3592	2.8169	3.8414
$q_{1,\text{median},0.6}$	3.8565	3.5804	4.1087
$adjR^2_{\text{median},0.6}$	0.99933	0.9992	0.99915
$p_{1,\text{stddev},0.6}$	0.10211	0.14025	0.064515
$p_{2,\text{stddev},0.6}$	-0.039587	-0.059232	-0.016738
$q_{1,\text{stddev},0.6}$	0.26738	0.10266	0.73387
$adjR^2_{\text{stddev},0.6}$	0.99768	0.99561	0.99991
$p_{1,v_Q,0.6}$	0.19744	0.34071	0.096813
$p_{2,v_Q,0.6}$	-0.084046	-0.14535	-0.037179
$q_{1,v_Q,0.6}$	0.41673	0.40797	0.50076
$adjR^2_{v_Q,0.6}$	0.98934	0.98884	0.99348

Table A.4: Parameters of the fitted curves for $V_V = 0.6$ with their adjusted R^2 goodness-of-fit statistic. Based on $cv_{Balls} = \{1, 2, 5, 10\}$.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.3}$	0.7042	0.5356	0.826
$p_{2,\text{mean},0.3}$	2.245	1.937	2.761
$q_{1,\text{mean},0.3}$	2.562	2.437	2.946
$adjR^2_{\text{mean},0.3}$	0.9989	0.9986	0.9980
$p_{1,\text{median},0.3}$	0.69158	0.51969	0.83186
$p_{2,\text{median},0.3}$	2.3699	1.9069	2.2876
$q_{1,\text{median},0.3}$	2.6858	2.3619	2.4238
$adjR^2_{\text{median},0.3}$	0.99939	0.99925	0.99891
$p_{1,\text{stddev},0.3}$	0.08296	0.1158	0.05732
$p_{2,\text{stddev},0.3}$	-0.01851	-0.02366	0.01544
$q_{1,\text{stddev},0.3}$	0.4733	0.4176	1.988
$adjR^2_{\text{stddev},0.3}$	0.9983	0.9974	0.9897
$p_{1,v_Q,0.3}$	0.1441	0.2485	0.07154
$p_{2,v_Q,0.3}$	-0.06394	-0.1061	-0.023
$q_{1,v_Q,0.3}$	0.1381	0.1758	0.3978
$adjR^2_{v_Q,0.3}$	0.9970	0.9983	0.9979

Table A.5: Parameters of the fitted curves for $V_V = 0.3$ with their adjusted R^2 goodness-of-fit statistic. Based on
$cv_{Balls} = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 4.0, 5.0\}$.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.4}$	0.7085	0.5446	0.8332
$p_{2,\text{mean},0.4}$	2.171	1.867	2.44
$q_{1,\text{mean},0.4}$	2.475	2.348	2.6
$adjR^2_{\text{mean},0.4}$	0.9984	0.9983	0.9984
$p_{1,\text{median},0.4}$	0.68131	0.50734	0.82719
$p_{2,\text{median},0.4}$	2.9232	2.3164	2.8131
$q_{1,\text{median},0.4}$	3.3345	2.9032	2.993
$adjR^2_{\text{median},0.4}$	0.99762	0.99762	0.99816
$p_{1,\text{stddev},0.4}$	0.08814	0.122	0.05281
$p_{2,\text{stddev},0.4}$	-0.0312	-0.04419	-0.01078
$q_{1,\text{stddev},0.4}$	0.2642	0.1603	0.6862
$adjR^2_{\text{stddev},0.4}$	0.9991	0.9991	0.9993
$p_{1,v_Q,0.4}$	0.1652	0.2826	0.07808
$p_{2,v_Q,0.4}$	-0.07338	-0.1216	-0.03125
$q_{1,v_Q,0.4}$	0.269	0.2841	0.3071
$adjR^2_{v_Q,0.4}$	0.9971	0.9972	0.9979

Table A.6: Parameters of the fitted curves for $V_V = 0.4$ with their adjusted R^2 goodness-of-fit statistic. Based on
$cv_{Balls} = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 4.0, 5.0\}$.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.5}$	0.7052	0.5407	0.8303
$p_{2,\text{mean},0.5}$	2.221	1.938	2.579
$q_{1,\text{mean},0.5}$	2.536	2.448	2.752
$adjR^2_{\text{mean},0.5}$	0.9989	0.9989	0.9989
$p_{1,\text{median},0.5}$	0.67496	0.49905	0.82054
$p_{2,\text{median},0.5}$	3.2059	2.5493	3.3732
$q_{1,\text{median},0.5}$	3.6723	3.2202	3.6016
$adjR^2_{\text{median},0.5}$	0.99866	0.99866	0.99868
$p_{1,\text{stddev},0.5}$	0.09737	0.1346	0.05843
$p_{2,\text{stddev},0.5}$	-0.03388	-0.04777	-0.01483
$q_{1,\text{stddev},0.5}$	0.356	0.2451	0.6743
$adjR^2_{\text{stddev},0.5}$	0.9989	0.9986	0.9992
$p_{1,v_Q,0.5}$	0.1904	0.3258	0.08974
$p_{2,v_Q,0.5}$	-0.07354	-0.12135	-0.03297
$q_{1,v_Q,0.5}$	0.5503	0.5604	0.5168
$adjR^2_{v_Q,0.5}$	0.9969	0.9968	0.9968

Table A.7: Parameters of the fitted curves for $V_V = 0.5$ with their adjusted R^2 goodness-of-fit statistic. Based on
$cv_{Balls} = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 4.0, 5.0\}$.

	sf_1	sf_2	sf_3
$p_{1,\text{mean},0.6}$	0.6983	0.5313	0.8242
$p_{2,\text{mean},0.6}$	2.305	2.043	2.778
$q_{1,\text{mean},0.6}$	2.634	2.586	2.964
$adjR^2_{\text{mean},0.6}$	0.9993	0.9993	0.9992
$p_{1,\text{median},0.6}$	0.66079	0.47558	0.80829
$p_{2,\text{median},0.6}$	3.6361	3.0122	4.2058
$q_{1,\text{median},0.6}$	4.1758	3.8303	4.501
$adjR^2_{\text{median},0.6}$	0.99906	0.99911	0.999
$p_{1,\text{stddev},0.6}$	0.1062	0.1467	0.06515
$p_{2,\text{stddev},0.6}$	-0.03574	-0.05112	-0.01823
$q_{1,\text{stddev},0.6}$	0.4445	0.31584	0.7238
$adjR^2_{\text{stddev},0.6}$	0.9983	0.9979	0.9988
$p_{1,v_Q,0.6}$	0.2151	0.3716	0.104
$p_{2,v_Q,0.6}$	-0.073025	-0.1189	-0.03168
$q_{1,v_Q,0.6}$	0.8084	0.8483	0.8471
$adjR^2_{v_Q,0.6}$	0.9974	0.9981	0.9980

Table A.8: Parameters of the fitted curves for $V_V = 0.6$ with their adjusted R^2 goodness-of-fit statistic. Based on
$cv_{Balls} = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 4.0, 5.0\}$.

A.2 Parameters of the curves used to fit the Al grains' statistics

In the following tables, we present the parameters for

$$f_x^j(cv_{Balls}) = p_{1,x,j} \cdot cv_{Balls}^3 + p_{2,x,j} \cdot cv_{Balls}^2 + p_{3,x,j} \cdot cv_{Balls} + p_{4,x,j}$$

used to fit the relationships between statistics F_C, stdF_C, stdE_F, $\bar{\bar{b}}$, std$\bar{b}$ and std$V$ of the Al grains and the parameters (cv_{Balls}, V_V) used to generate force-biased ball packings for generating random Laguerre tessellations. For the relationship between E_F and (cv_{Balls}, V_V) we fitted rational curves, namely

$$f_x^j(cv_{Balls}) = \frac{p_{1,x,j} \cdot cv_{Balls} + p_{2,x,j}}{cv_{Balls} + q_{1,x,j}}.$$

statistic	$p_{1,\cdot,0.3}$	$p_{2,\cdot,0.3}$	$p_{3,\cdot,0.3}$	$p_{4,\cdot,0.3}$	$adjR^2_{f_\cdot^{0.3}}$
stdEF	-0.0092943	-0.026754	0.18249	2.2265	0.037935
meanFC	-0.53563	3.0508	-5.4675	18.686	-0.66603
stdFC	0.01196	-0.25485	0.7079	6.4506	-0.39023
mean$\bar{\bar{b}}$	-0.096301	0.5567	-1.2587	22.832	0.8863
std$\bar{\bar{b}}$	-0.34098	1.3188	-0.68333	10.572	0.93344
std$V$	-2108.4	10401	-13577	9108.6	0.24282

statistic	$p_{1,\cdot,0.3}$	$p_{2,\cdot,0.3}$	$q_{1,\cdot,0.3}$	$adjR^2_{f_\cdot^{0.3}}$
meanEF	5.5271	19.321	3.6178	-0.12366

Table A.9: Parameters of the fitted curves for $V_V = 0.3$ with their adjusted R^2 goodness-of-fit statistic.

statistic	$p_{1,\cdot,0.4}$	$p_{2,\cdot,0.4}$	$p_{3,\cdot,0.4}$	$p_{4,\cdot,0.4}$	$adjR^2_{f_\cdot^{0.4}}$
stdEF	-0.033547	0.10721	-0.08493	2.3744	0.61393
meanFC	-0.40799	1.9603	-2.9647	17.021	0.22213
stdFC	-0.12755	0.36888	-0.21701	6.8674	0.56104
mean$\bar{\bar{b}}$	-0.13575	0.70316	-1.5422	22.843	0.91934
std$\bar{\bar{b}}$	-0.016655	-0.42289	2.131	9.2853	0.92002
std$V$	-1194.7	5718.5	-6268.4	5886.4	0.95692

statistic	$p_{1,\cdot,0.4}$	$p_{2,\cdot,0.4}$	$q_{1,\cdot,0.4}$	$adjR^2_{f_\cdot^{0.4}}$
meanEF	3.8661	3621.5	673.51	-0.21315

Table A.10: Parameters of the fitted curves for $V_V = 0.4$ with their adjusted R^2 goodness-of-fit statistic.

statistic	$p_{1,\cdot,0.5}$	$p_{2,\cdot,0.5}$	$p_{3,\cdot,0.5}$	$p_{4,\cdot,0.5}$	$adjR^2_{f^{0.5}}$
stdEF	0.0090559	-0.053084	0.080924	2.3144	-0.11133
meanFC	0.13044	-0.61147	0.6586	15.45	0.27167
stdFC	0.25305	-1.4191	2.3889	5.5669	0.041527
mean$\bar{\bar{b}}$	0.003746	0.079527	-0.89137	22.555	0.98699
std$\bar{\bar{b}}$	-0.065567	0.060254	1.1813	9.8628	0.91091
std$V$	1302	-6336.6	12879	-3666.9	0.93035

statistic	$p_{1,\cdot,0.5}$	$p_{2,\cdot,0.5}$	$q_{1,\cdot,0.5}$	$adjR^2_{f^{0.5}}$
meanEF	-33.628	10258	1900.8	0.23656

Table A.11: Parameters of the fitted curves for $V_V = 0.5$ with their adjusted R^2 goodness-of-fit statistic.

statistic	$p_{1,\cdot,0.6}$	$p_{2,\cdot,0.6}$	$p_{3,\cdot,0.6}$	$p_{4,\cdot,0.6}$	$adjR^2_{f^{0.6}}$
stdEF	-0.32096	1.6663	-2.7579	3.8053	-1.0689
meanFC	-1.4708	7.5747	-12.533	22.162	0.12016
stdFC	-1.6887	8.7371	-14.403	14.386	0.20768
mean$\bar{\bar{b}}$	-0.85187	4.6311	-8.7444	26.687	0.94711
std$\bar{\bar{b}}$	-0.57902	2.3584	-1.8273	11.181	0.9779
std$V$	-1310.3	5710	-4243.2	4437.4	0.97803

statistic	$p_{1,\cdot,0.6}$	$p_{2,\cdot,0.6}$	$q_{1,\cdot,0.6}$	$adjR^2_{f^{0.6}}$
meanEF	5.271	0.36214	0.047311	0.42055

Table A.12: Parameters of the fitted curves for $V_V = 0.6$ with their adjusted R^2 goodness-of-fit statistic.

Appendix B

Curriculum Vitae

Academic Curriculum Vitae

04/2014 - 01/2017	Scholarship within GrK 1932 of Deutsche Forschungsgemeinschaft
06/2013 - 01/2017	PhD student in Mathematics at Technische Universität Kaiserslautern, Germany
06/2013 - 01/2017	PhD student at Fraunhofer ITWM, Department of Image Processing, Germany
10/2007 - 03/2013	Study of Mathematics at Technische Universität Kaiserslautern, Germany

Akademischer Lebenslauf

04/2014 - 01/2017	Promotionsstipendium innerhalb des GrK 1932 der Deutschen Forschungsgemeinschaft
06/2013 - 01/2017	Doktorandin am Fachbereich Mathematik an der Technischen Universität Kaiserslautern, Deutschland
06/2013 - 01/2017	Doktorandin am Fraunhofer ITWM, Abteilung Bildverarbeitung, Deutschland
10/2007 - 03/2013	Studium der Mathematik an der Technischen Universität Kaiserslautern, Deutschland

List of Publications

H. Berek, U. Ballaschk, C. G. Aneziris, K. Losch, and K. Schladitz. The correlation between local deformation and local phase transformation in MMC foams as investigated by in situ XCT experiments and EBSD. *Materials Characterization*, 107:139–148, 2015.

J. H. Fitschen, K. Losch, and G. Steidl. Unsupervised multi class segmentation of 3D images with intensity inhomogeneities. *Journal of Visual Communication and Image Representation*, 46:312–323, 2017.

K. Losch, K. Schladitz, U. Ballaschk, H. Berek, and C. G. Aneziris. Interrupted in-situ compressive deformation experiments on mmc foams in an xct: Experiments and estimation of displacement fields. *ImageAnalStereol*, 33(2):131–145, 2014.

Bibliography

[AJ09] H. Altendorf and D. Jeulin. 3D directional mathematical morphology for analysis of fiber orientations. *Image Analysis and Stereology*, 28(3):143–153, 2009.

[AJ10] H. Altendorf and D. Jeulin. Random walk based stochastic modeling of 3D fiber systems. *Physical Review E*, 83(4), 2010.

[BBS02] A. Bezrukov, M. Bargieł, and D. Stoyan. Statistical analysis of simulated random packings of spheres. *Part. Part. Syst. Charact.*, (19):111–118, 2002.

[BGS04] J. Boselli, P. J. Gregson, and I. Sinclair. Quantification of particle distribution effects on fatigue in an $Al - SiC_p$ composite. *Materials Science and Engineering: A*, 379(1-2):72–82, 2004.

[BR04] V. Beneˇz and J. Rataj. *Stochastic Geometry: Selected Topics*. Kluwer Academic Publishers, Dordrecht, 2004.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[BS06] A. Bezrukov and D. Stoyan. Simulation and statistical analysis of random packings of ellipsoids. *Part. Part. Syst. Char.*, 23(5):388–398, 2006.

[BVJ05] A. Baddely and E.B. Vedel Jensen. *Stereology for Statisticians*. Chapman & Hall/CRC, 2005.

[CC06] N. Chawla and K. K. Chawla. Microstructure-based modeling of the deformation behavior of particle reinforced metal matrix composites. *Journal of Materials Science*, 41(3):913–925, 2006.

[CGW04] N. Chawla, V. V. Ganesh, and B. Wunsch. Three-dimensional (3D) microstructure visualization and finite element modeling of the mechanical behavior of SiC particle reinforced aluminum composites. *Scripta Materialia*, 51(2):161–165, 2004.

[COH86] L. M. Cruz-Orive and E. B. Hunziker. Stereology for anisotropic cells: applicatiopn to growth cartilage. *Journal of Microscopy*, 143(1):47–80, 1986.

[CSG06] N. Chawla, R. Sidhu, and V. Ganesh. Three-dimensional visualization and microstructure-based modeling of deformation in particle-reinforced composites. *Acta Materialia*, 54(6):1541–1548, 2006.

[CSKM13] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. Wiley, 2013.

[Dem06] W. Demtröder. *Experimentalphysik 1*. Springer, 4 edition, 2006.

[Dob10] D. Dobrovolskij. Optimierungsmethoden für geometrische Mikrostrukturmodelle. Bachelor's thesis, Beuth Hochschule für Technik Berlin, 2010.

[Ebe08] D. Eberly. Intersection of convex objects: The method of separating axes. `http://www.geometrictools.com/`, 2008. Accessed: 2015.

[Esc12] J. Escoda. *Modélisation morphologique et micromécanique 3D de matériaux cimentaires*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2012.

[FMS] J. H. Fitschen, J. Ma, and S. Schuff. Removal of curtaining effects by a variational model with directional forward differences. `https://arxiv.org/abs/1507.00112`. Accessed: 2017.

[Fra11] Fraunhofer ITWM, Department of Image Processing. MAVIlib – modular algorithms for volume images. `http://www.mavi-3d.de`, 2011.

[FRZ16] J. Franke, C. Redenbach, and N. Zhang. On a mixture model for directional data on the sphere. *Scandinavian Journal of Statistics*, 43(1):139–155, 2016.

[FVF88] M. Fátima Vaz and M. A. Fortes. Grain size distribution: The lognormal and the gamma distribution functions. *Scripta Metallurgica*, 22(1):35–40, 1988.

[GC05] V. V. Ganesh and N. Chawla. Effect of particle orientation anisotropy on the tensile behavior of metal matrix composites: Experiments and microstructure-based simulation. *Materials Science and Engineering: A*, 391(1-2):342–353, 2005.

[GJ85] H. J. G. Gundersen and E. B. Jensen. Stereological estimation of the volume-weighted mean volume of arbitrary particles observed on random sections. *Journal of Microscopy*, 138(2):127–142, 1985.

[HK04] N. Hort and K. U. Kainer. Pulvermetallurgisch hergestellte Metall-Matrix-Verbundwerkstoffe. In Karl Ulrich Kainer, editor, *Metallische Verbundwerkstoffe*, pages 260–295. Wiley-VCH, Weinheim, 2004.

[Hum04] F. J. Humphreys. Characterisation of fine-scale microstructures by electron backscatter diffraction (ebsd). *Scripta materialia*, 51(8):771–776, 2004.

[HWS94] J. N. Hall, J. Wayne Jones, and A. K. Sachdev. Particle size, volume fraction and matrix strength effects on fatigue behavior and particle fracture in 2124 aluminum-SiC_p composites. *Materials Science and Engineering: A*, 183(1-2):69–80, 1994.

[ITW14] Fraunhofer ITWM. Toolip. `http://www.itwm.fraunhofer.de/abteilungen/bildverarbeitung/oberflaecheninspektion/toolip.html`, 2014.

[Kai03] K.U. Kainer. *Metallische Verbundwerkstoffe*. Wiley-VCH, Weinheim, 2003.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[Lau07] C. Lautensack. *Random Laguerre Tessellations*. PhD thesis, Universität Karlsruhe (TH), 2007.

[Lie14] A. Liebscher. *Stochastic Modelling of Foams*. PhD thesis, Technische Universität Kaiserslautern, 2014.

[LR12] A. Liebscher and C. Redenbach. 3D image analysis and stochastic modelling of open foams. *International Journal of Materials Research*, 103(2):155–161, 2012.

[LSB+ed] K. Losch, S. Schuff, F. Balle, T. Beck, and C. Redenbach. A stochastic microstructure model for particle reinforced aluminium matrix composites. *Journal of Microscopy*, submitted.

[LY09] D. Liu and J. Yu. Otsu method and K-means. In *Ninth International Conference on Hybrid Intelligent Systems, 2009. HIS'09.*, volume 1, pages 344–349. IEEE, 2009.

[Lyo04] R.G. Lyons. *Understanding Digital Signal Processing*. Prentice Hall, 2 edition, 2004.

[Mat] Materion Aerospace Metal Composites Limited. AMC225XE: Particle reinforced aluminum alloy. `https://materion.com/-/media/files/pdfs/aerospace-metal-composites/amc_amc225xe-particle-reinforced-aluminum-alloy-11-8-2012.pdf`. Accessed: 2017.

[MAT16] MATLAB. *version (R2016b)*. The MathWorks Inc., Natick, Massachusetts, 2016.

[mat17] 2-d Gaussian filtering of images - MATLAB imgaussfilt - MathWorks Deutschland. `https://de.mathworks.com/help/images/ref/imgaussfilt.html`, 2017. Accessed: 2017.

[MAV05] MAVI. Modular algorithms for volume images. `http://www.mavi-3d.de`, 2005.

[MW04] J. Møller and R. P. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. Chapman& Hall/CRC, Boca Raton, FL, 2004.

[NM65] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[OJG90] A. Odgaard, E. B. Jensen, and H. J. G. Gundersen. Estimation of structural anisotropy based on volume orientation. a new concept. *Journal of Microscopy*, 157(2):149–162, 1990.

[OM00] J. Ohser and F. Mücklich. *Statistical Analysis of Microstructures in Materials Science*. Wiley, Chichester, New York, 2000.

[OS09] J. Ohser and K. Schladitz. *3D Images of Materials Structures: Processing and Analysis*. Wiley-VCH, Weinheim, 2009.

[Ots79] N. Otsu. A threshold selection method from gray level histograms. *IEEETSystManCyb*, 9(1):62–66, March 1979.

[R D08] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[RAB+12] B.W. Reed, B.L. Adams, J.V. Bernier, C.M. Hefferan, A. Henrie, S.F. Li, J. Lind, R.M. Suter, and M. Kumar. Experimental tests of stereological estimates of grain boundary populations. *Acta Materialia*, 60(6-7):2999–3010, 2012.

[Red09] C. Redenbach. Microstructure models for cellular materials. *Computational Materials Science*, 44(4):1397–1407, 2009.

[RSVW14] C. Redenbach, K. Schladitz, I. Vecchio, and O. Wirjadi. Image analysis for microstructures based on stochastic models. *GAMM Mitteilungen*, 37(2):281–305, 2014.

[Rud87] W. Rudin. *Reelle und komplexe Analysis*. Oldenbourg Verlag, 1987.

[SACF+12] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.

[SJW90] B. W. Silverman, M. C. Jones, and J. D. Wilson. A smoothed EM approach to indirect estimation problems, with particular reference to stereology and emission tomography. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(2):271–324, 1990.

[Soi99] P. Soille. *Morphological Image Analysis*. Springer, Berlin, 1999.

[SP00] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *PatternRecogn*, 33(2):225–236, 2000.

[SPRB+06] K. Schladitz, S. Peters, D. Reinel-Bitzer, A. Wiegmann, and J. Ohser. Design of acoustic trim based on geometric modeling and flow simulation for non-woven. *Computational Materials Science*, 38(1):56–66, 2006.

[SS92] Y. Sugimura and S. Suresh. Effects of sic content on fatigue crack growth in. *Metallurgical Transactions A*, 23(8):2231–2242, 1992.

[SS12] B. Shafei and G. Steidl. Segmentation of images with separating layers by fuzzy c-means and convex optimization. *Journal of visual communication and image representation*, 23(4):611–621, 2012.

[SSKH11] C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 8th IEEE International Symposium on Biomedical Imaging (ISBI 2011)*, pages 230–233, 2011.

[SW08] R. Schneider and W. Weil. *Stochastic and Integral Geometry*. Probability and Its Applications. Springer, Heidelberg, 2008.

[tea13] The GIMP team. Gimp 2.8.10. `www.gimp.org`, 2013.

[TGC02] L.M Tham, M. Gupta, and L. Cheng. Effect of reinforcement volume fraction on the evolution of reinforcement size during the extrusion of Al-SiC composites. *Materials Science and Engineering: A*, 326(2):355–363, 2002.

[TY09] J. Toriwaki and H. Yoshida. *Fundamentals of Three-Dimensional Digital Image Processing*. Springer, 2009.

[Vec14] I. Vecchio. *Image based characterization and geometric modeling of 3d materials microstructures*. PhD thesis, Technische Universität Kaiserslautern, 2014.

[Ved98] E. B. Vedel Jensen. *Local Stereology*. World Scientific Publishing, Singapore, 1998.

[Wal] C. Walck. Hand-book on statistical distributions for experimentalists. Technical report. Accessed: 2017.

[wik] Variationskoeffizient. `https://de.wikipedia.org/wiki/Variationskoeffizient`. Accessed: 2017.

[YLS98] H. Yu, G. Liu, and X. Song. On the relation between the number-weighted and volume-weighted grain volume distribution parameters. *Metallurgical and Materials Transactions A*, 29(12):3081–3086, 1998.

[Zha13] N. Zhang. *Fibre Processes and their Applications*. PhD thesis, Technische Universität Kaiserslautern, 2013.