

Algorithmic structural segmentation of defective particle systems: A lithium-ion battery study

Daniel Westhoff¹, Donal P. Finegan², Paul R. Shearing², and
Volker Schmidt¹

¹Institute of Stochastics, Ulm University, D-89069 Ulm, Germany

²Electrochemical Innovation Laboratory, Department of Chemical Engineering, University College London, WC1E 6BT, London, UK

August 22, 2017

Keywords: Segmentation, particles, cracks, breakages, thermal runaway

Abstract

We describe a segmentation algorithm that is able to identify defects (cracks, holes and breakages) in particle systems. This information is used to segment image data into individual particles, where each particle and its defects are identified accordingly. We apply the method to particle systems that appear in Li-ion battery electrodes. First, the algorithm is validated using simulated data from a stochastic 3D microstructure model, where we have full information about defects. This allows us to quantify the accuracy of the segmentation result. Then we show that the algorithm can successfully be applied to tomographic image data from real battery anodes and cathodes, which are composed of particle systems with very different morphological properties. Finally, we show how the results of the segmentation algorithm can be used for structural analysis.

Corresponding author: Daniel Westhoff, Institute of Stochastics, Ulm University, D-89069 Ulm, Germany, Tel: +49-731-50-23617; Fax +49-731-50-23649; e-mail daniel.westhoff@uni-ulm.de

1 Introduction

In many applications investigated in materials science, the microstructure of the considered material is of great interest, as it significantly influences its properties, see, e.g., Torquato (2013) and Gaiselmann et al. (2014). Moreover, in many cases, defects of the materials under consideration can occur under various conditions, see, e.g., Antonyuk et al. (2006). A typical example where this phenomena occurs are electrodes in lithium-ion batteries that undergo thermal runaway, see, e.g., Finegan et al. (2015). During this process, which can occur as a result of extreme conditions such as high temperature or overcharging, the temperature in the cell increases dramatically, leading to deformation of the particles in the electrodes, which can be observed in tomographic images, see Finegan et al. (2016). Moreover, cracks in particles do not only occur due to failure, but, e.g., due to the heterogenous nature of materials. A typical example are electrodes made from natural graphite. Regardless of their origin, cracks, holes and breakages in the particles can strongly influence the underlying electrochemical and kinetic processes, as they lead to a higher surface area of the active material. The nature of these particle morphologies will affect the operation of the cell, as well as the nucleation and propagation of failure. Moreover the post-mortem particle morphology can provide valuable forensic information regarding the thermal failure process.

Therefore, it is of high interest to analyse the occurrence of such defects by investigating which kind of particles are prone to be damaged or what kind of shapes the defects have. This makes it necessary to perform a structural segmentation of tomographic image data. A widely used algorithm for the segmentation of a system into individual particles is the watershed approach, see Roerdink and Meijster (2001) for an overview. Particles are identified based on watershed seeds, which are chosen as the local maxima of the distance map of the binary image. It is assumed that for each particle there is exactly one local maximum of the distance map. However, this is not necessarily true for

defective particle systems. In Gillibert and Jeulin (2013), an approach for the 3D reconstruction of fragmented grains is proposed, which is based on a morphological closing and so-called h -maxima. It reduces the number of watershed seeds. Furthermore, a second variant using a cluster algorithm is considered in Gillibert and Jeulin (2013). Both methods reconstruct the original particle from a system of multiple fragments. In Petrich et al. (2017), an approach for detection of breakages using machine learning is proposed, which is trained using hand-labelled data and is then able to automatically detect broken particles in systems similar to the training data.

In the present paper, a different approach for segmenting defective particle systems is considered, which not only detects broken particles, but also detects cracks (which do not separate the particle into multiple pieces) and holes. First, possible voxels in the image that could be part of a defect (either a crack, hole or complete breakage of a particle) are found. Then, those voxels are temporarily added to the particle phase and a segmentation into individual particles is performed using watershed techniques. Finally, the list of ‘defect voxels’ is ‘thinned’ again according to some characteristic criteria of holes, breakages and cracks. The output is a system of individual particles with their corresponding defects marked accordingly. Note that the whole algorithm is performed in 3D, while some figures in the present paper show 2D slices of the 3D data sets to enhance visibility of details.

To show the broad applicability of the algorithm, we use it for the segmentation of various different data sets. First, for validation purposes, we generate simulated data describing the microstructure of a battery anode using the stochastic microstructure model described by Westhoff et al. (2017). By artificially inserting cracks, holes and breakages, we can validate the accuracy of the segmentation algorithm. Finally, we apply the algorithm to two different tomographic data sets of anode and cathode material from a lithium-ion battery, which exhibit very different particle shapes. The first data set was extracted from the cathode of a commercial Li-ion pouch cell that underwent overcharge-

induced thermal runaway. The second data set was extracted from a commercial Li-ion anode in its fresh state. We show that the algorithm described in the present paper is able to detect both, defects induced by overcharging as well as cracks that stem from the heterogenous nature of graphite.

The outline of the paper is as follows. In Section 2, we provide a detailed description of the segmentation algorithm. Then, in Section 3, we validate the algorithm on simulated data. In Section 4, we show an application to real electrode data from tomographic images. Finally, we conclude the paper in Section 5.

2 Description of the algorithm

The idea of the algorithm is as follows. We assume that the tomographic image data is already preprocessed in a way that we have a binary image $I : W \rightarrow \{0, 1\}$, where $W \subset \mathbb{Z}^3$ is the observation window with \mathbb{Z} being the set of integers. For each voxel $v = (x, y, z) \in W$ belonging to a particle, we have $I(v) = 1$, whereas for the remaining voxels we have $I(v) = 0$. Note that the remaining voxels include the pore space as well as further additives. We will call this part the background phase in the following. An example is shown in Figure 1, where particles are shown in grey and the background phase is shown in white. Note that some cracks and holes in the particles are visible in the image, which appear in white, i.e., they are assigned to the background phase. The aim is to identify those background voxels where a defect of a particle is present. An overview of the corresponding algorithm is shown in Figure 2, and an example of application of each step is visualized in Figure 3. The basic idea is as follows. After closing holes using a cluster detection algorithm and marking them as defect voxels (red voxels in Figure 3(a)), further candidates for those defect voxels are identified based on the distance map (red voxels in Figure 3(b)). All candidates are initially added to the particle phase, and a further cluster detection is performed to identify possible new resulting holes

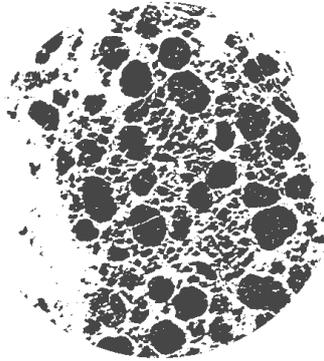


Figure 1: 2D slice of a binarized 3D image of a lithium-ion battery cathode, where many defects (cracks, holes and breakages) of particles are visible.

(red voxels in Figure 3(c)). Given all candidates for defect voxels, the system of particle voxels and defect voxels is considered as one phase (blue and white voxels in Figure 3(d)) and segmented into individual particles using watershed techniques (Figure 3(e)). Then, all candidates for defect voxels that belong to a watershed separator are removed (red voxels in Figure 3(f)). This however still leaves some wrongly chosen candidates for defect voxels, which are also deleted using a criterion that considers their local neighbourhood (red voxels in Figure 3(g)). Finally, small clusters of defect voxels, which might result from small inaccuracies, are deleted (red voxels in Figure 3(h)). Due to the deletion of candidates, some particles together with their defects identified by the watershed algorithm might not be connected any more, so the final set of particles is identified using a cluster detection algorithm, where small particles are considered as noise in the greyscale image and removed. An example of the output of the algorithm is shown in Figure 3(i), where particles are shown in different colours and defects are marked in blue.

The algorithm is implemented in the software framework GeoStoch, see Mayer et al. (2004). In the next subsections, each step is explained in detail.

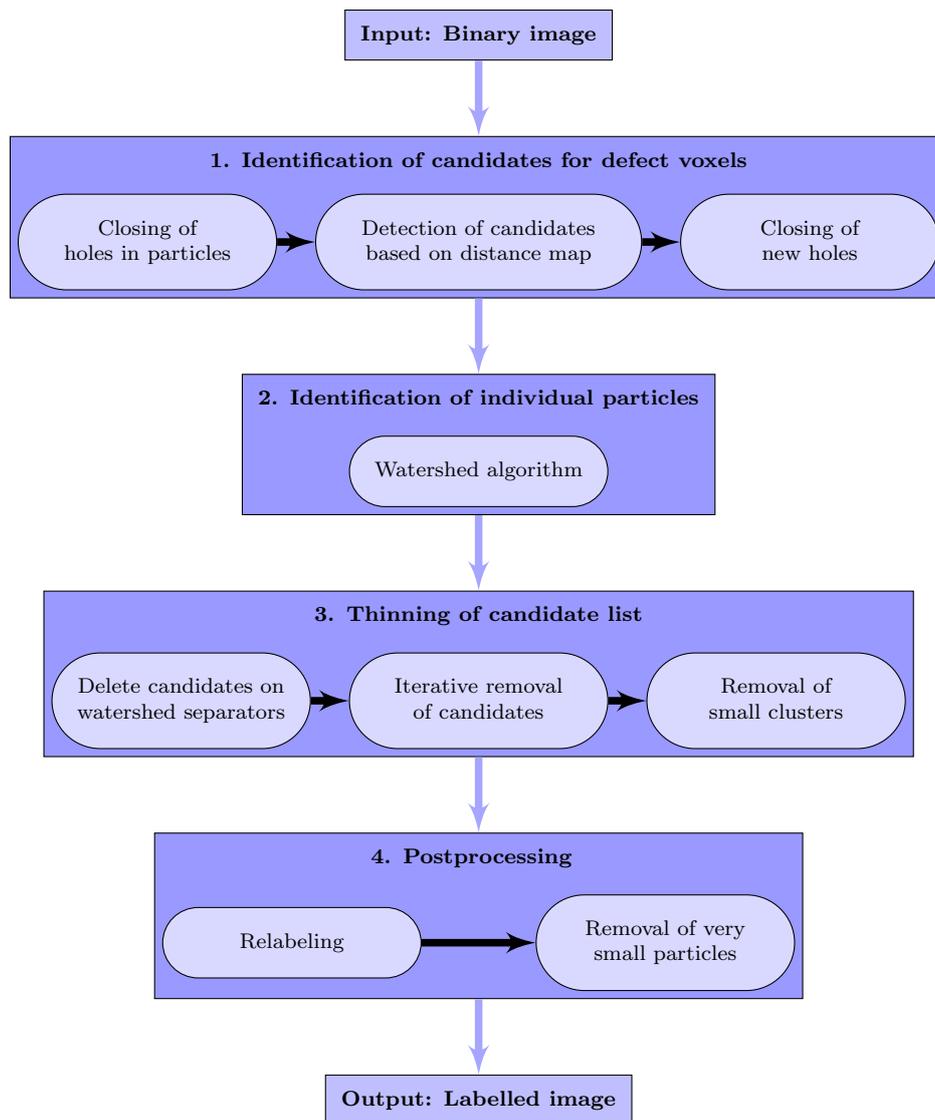


Figure 2: Overview of the four stages involved in the segmentation algorithm.

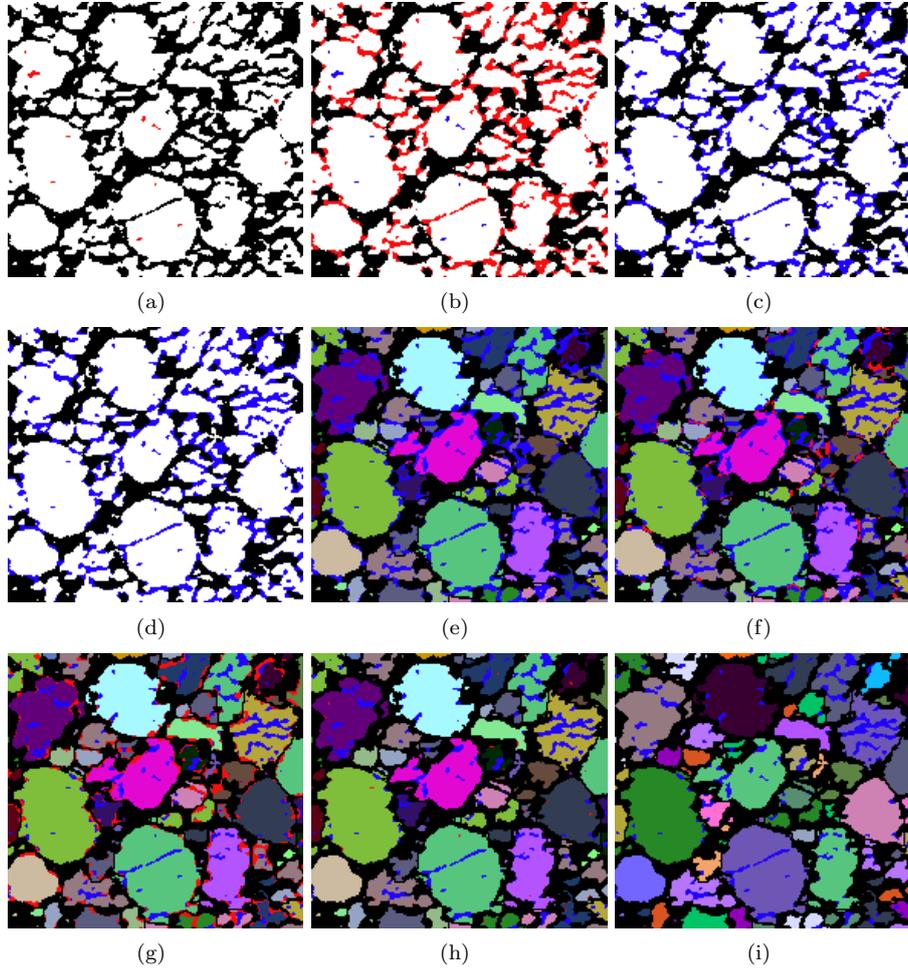


Figure 3: Application of the algorithm to an example data set. Candidates for defect voxels are always shown in blue, while red voxels show those candidates that are added or removed in the corresponding step. Note that the algorithm is performed in 3D, but only 2D slices are shown here. (a) Holes that are completely contained in the particles are closed. (b) Candidates are added based on the distance map. (c) If all candidates found so far are added to the particle phase, new holes occur, which are closed in this step. (d) All candidates are temporarily added to the particle phase and the image is used for segmentation into individual particles using the watershed algorithm. (e) The result of the watershed algorithm. (f) Candidates for defect voxels that are located on watershed separators are removed. (g) Candidates are removed based on their local neighbourhood. (h) Small clusters of candidate voxels are identified and added to the corresponding particle. (i) Final output of the segmentation algorithm, with particles labelled in different colours and defects shown in blue.

2.1 Identification of candidates for defect voxels

In the following, the identification of candidates for defect voxels is explained (Stage 1 in Figure 2). We add all candidates to a list L . Whenever a candidate v is added to L , we put $I'(v) = 1$ in a copy I' of I , i.e., we assign the defect voxel candidates to the particle phase. This copy will be needed for technical reasons, e.g., when identifying the individual particles in Section 2.2.

2.1.1 Closing of holes in particles

To begin with, small holes in the particles that have no contact to the remaining background phase are closed. This is done using a cluster detection algorithm on the background phase, see Hoshen and Kopelman (1976) for details. Then, every cluster with a size less than some parameter h_{\max} is added to L , see the red voxels in Figure 3(a).

2.1.2 Detection of candidates based on distance map

In the next step, further candidates are identified based on the distance map $D : W \rightarrow [0, \infty)$, which is constructed as follows. For each voxel $v \in W$, we put

$$D(v) = \min_{v': I'(v')=1} |v - v'|,$$

i.e., $D(v)$ is the shortest distance from v to a voxel that belongs to the particle phase. An example is shown in Figure 4 (top), where the blue parts indicate particle voxels, and the numbers in the voxels correspond to the values of the distance map. The particle on the top has no defects, whereas the two particles on the bottom show a breakage (left) and a crack (right). We can observe that the values of the distance map at and around voxels that belong to a defect are very small compared to other background voxels. This is exemplarily indicated for three voxels by the circles containing the numbers coloured in orange in the top of Figure 4. In the case of the two voxels that belong to a defect (center of circle on the bottom left and right), almost all values within the circle are 0

(particle voxels) or 1, whereas in the case of the voxel on the top, which does not belong to a defect, also much larger numbers occur within the corresponding circle. Therefore, for each background voxel $v \in W$ with $I'(v) = 0$, we sum up all the values of $D(v')$ with $|v - v'| \leq r$ for some parameter r and save the result in a new image $S : W \rightarrow [0, \infty)$. The radius r is determined by visual inspection for the sketch. Throughout this paper, parameters are chosen by visual inspection of the resulting segmentation, where it was always unproblematic to determine parameters leading to good results. As the runtime of the algorithm depends on the size of the image under consideration, a good strategy for exploring the parameter space is to apply the algorithm to a small, representative cutout of the data set with different parameters and to analyse the quality of the segmentation. Further methods for parameter choice are discussed in Section 5. The result is shown at the bottom of Figure 4 (with $r = 3$ in this example). Now we add all voxels v with $S(v) < d_{\max}$ for some parameter d_{\max} to the list of defect voxel candidates L . At the bottom of Figure 4, all voxels v with $S(v) < 27$ are marked in orange colour, where again, the number 27 has been chosen by visual inspection. Note that this includes all voxels that belong to defects, however, also some voxels between the particle on the top and the one on the bottom left. Those will be removed from L in a later step, see Section 2.3.2 for details.

For voxels with a distance of less than r to the boundary, we only sum up all the available values, and scale the threshold d_{\max} by the ratio of voxel that are available and the number of voxel that are in a sphere with radius r .

In Figure 3(b) (red voxels), an example of defect candidate voxels that are found in this step is shown.

2.1.3 Further closing of holes

When assigning defect voxel candidates to the particle phase in I' , this might lead to new holes, i.e., clusters of background voxels that are completely contained within the particle phase. We proceed like in Section 2.1.1 to identify

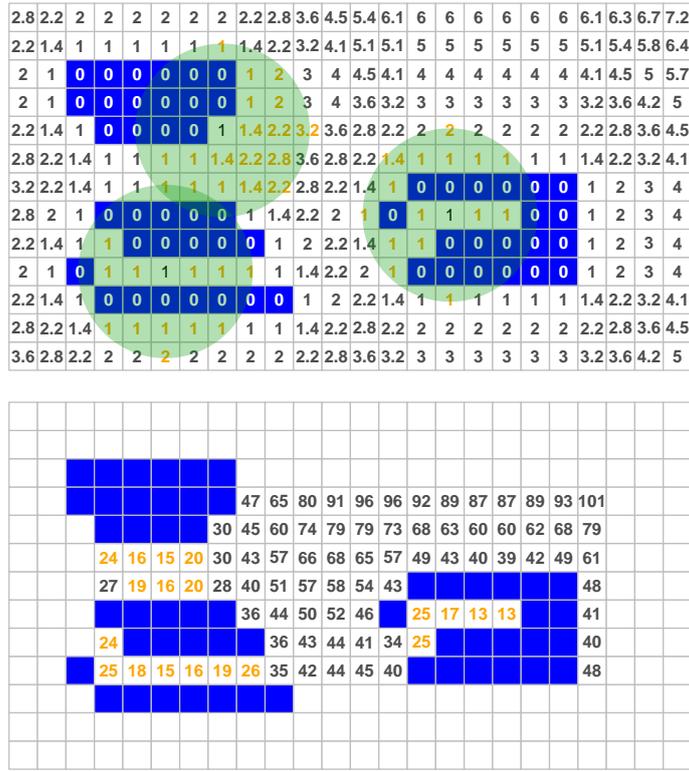


Figure 4: Top: 2D sketch showing the distance map (numbers in the voxels) for a system of 3 particles (blue voxels). For each voxel, the values of the distance map of all voxels within a certain region (green circles shown exemplarily for three voxels) are considered for identifying defect voxel candidates. Bottom: Sum of all distance map values in a circle with radius $r = 3$. Smaller numbers (orange) indicate defect voxel candidates.

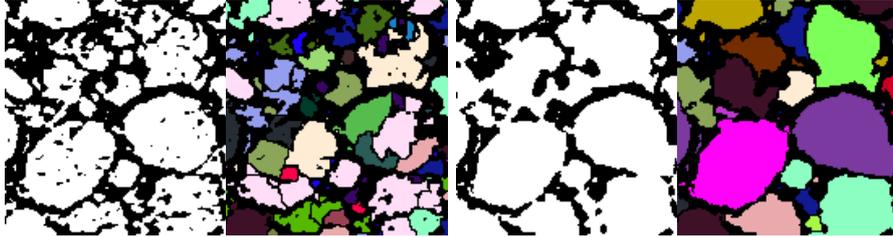
these holes, add them to L and assign them to the particle phase in I' . An example is shown in Figure 3(c), red voxels.

2.2 Segmentation into individual particles

As already mentioned, the problem of segmenting defective particle systems using watershed techniques is that this procedure would result in an oversegmentation, as the defects have an influence on the local maxima of the distance map. However, by identifying candidates for defect voxels and assigning them to the particle phase in I' , it is possible to successfully apply watershed segmentation to I' . This is illustrated in Figure 5, where in Figure 5(a) the defects of the particles induce many local maxima of the distance map, which is why the segmentation results in a system of highly fragmented particles. By first adding defect voxel candidates to the binary image, the distance map is smoothed and therefore the number of local maxima is reduced, so the segmentation results in a more reasonable set of particles, see Figure 5(b).

Depending on the data set, different variations of the standard watershed algorithm are useful in order to get a reasonable segmentation into individual particles. In many cases, the standard approach leads to oversegmentation, so many extensions have been proposed, which, e.g., join several local maxima of the distance map to a single watershed seed, see, e.g., Spettl et al. (2015). For a detailed analysis of a data set, a suitable methodology for watershed separation has to be found. In order to achieve comparable results, throughout this paper, we use the watershed algorithm implemented in Avizo, see VSG - Visualization Sciences Group - Avizo Standard (2017). Avizo's watershed algorithm provides reasonable results for all the data sets under consideration, however, there are some particles that are not well detected. We note that for a more detailed analysis of defective particle systems, special attention has to be paid to this step.

The watershed segmentation (Stage 2 in Figure 2) results in an image $A : W \rightarrow \mathbb{N}$, where \mathbb{N} is the set of all non-negative integers including zero. Each



(a) Watershed algorithm applied to binary data with defects. (b) Watershed algorithm applied to the same binary data, where defect voxel candidates are added to the particle phase.

Figure 5: 2D slices showing the effect of defects on the watershed algorithm. While particles are highly fragmented when using the original binary image as input (a), particles are much better identified when first assigning defect voxel candidates to the particle phase (b).

number has the role of a label, i.e., if two voxels have the same number, they both belong to the same particle. The background phase is labelled with zero. We add a further label $d \in \mathbb{N}$ (that is not given to any particle yet) and set all voxel $v \in L$ to d in A . An example can be seen in Figure 3(e), where the colours represent the labels of the particles, and blue voxels correspond to candidates for defect voxels.

2.3 Thinning of candidate list

As can be observed in Figure 3(e), so far, too many candidates for defect voxels are detected, especially at the boundary of particles. Thus, candidates have to be removed from the list L (Stage 3 in Figure 2). The procedure is described in the following subsections in detail.

2.3.1 Removing of candidates on watershed separators

Wherever the watershed algorithm separates two particles, we assume a boundary between two neighbouring particles, which does not belong to a defect. This is why in the first place, all candidates from L that are located on a watershed separator are removed, see Figure 3(f), red voxels. The corresponding voxels are assigned to the background phase in A .

Note that, as already seen in Section 2.1.2 and Figure 4, there might be some wrong candidates between two neighbouring particles. This leads to the particles growing together in I' , however, they are split again by the watershed algorithm. A part of the corresponding wrong candidates is therefore deleted in this step. The remaining wrong defect voxels are located at the boundary of particles and will be removed in the next step.

2.3.2 Further iterative removal

At this point, there are still some wrong defect voxel candidates, which mainly occur at the boundary of particles (as here the corresponding values of the distance map considered in Section 2.1.2 are small). These wrong candidates will be identified in this step based on their local neighbourhood. Each voxel has 26 neighbours. Typically, voxels on positions with defects will be surrounded by further defect or particle voxels, whereas wrong candidates on the boundary of particles are surrounded by many void (background) voxels. This is also illustrated in Figure 6, where on the left-hand side, an example of a wrong candidate (grey) at the surface of a particle (blue voxels) is shown. It is surrounded by 14 void neighbours. On the right-hand side, the two grey voxels belong to a crack in the particle. The considered grey voxel is only surrounded by 8 void neighbours. Thus, we iteratively remove voxels from L with more than or equal to n_{\max} void neighbours in the labelled image A . To be more precise, for each voxel in L , we compute the number of its void neighbours in A . If this number is larger than or equal to n_{\max} , we save the voxel in a list L' . When all voxels from L have been considered and L' is not empty, we delete every voxel $v \in L'$ from L and set $A(v) = 0$. Then, a new list L' is built as described above, until there are no more candidates with more than or equal to n_{\max} void neighbours. An example can be found in Figure 3(g), where the red voxels show those candidates found in this step.

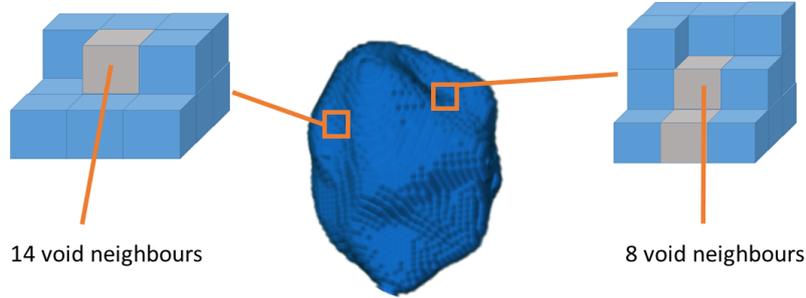


Figure 6: Sketch showing different neighbourhoods of defect voxel candidates. Blue: Particle voxels; Grey: Defect voxel candidates.

2.3.3 Removal of too small clusters

Finally, we consider all the connected components of defect voxel candidates. We only keep those connected components the size of which is greater than some parameter c_{\min} . This removes some small artefacts, see the red voxels in Figure 3(h). Those voxels are assigned to the corresponding particle.

2.4 Relabelling and removal of very small particles

Due to the removal of defect voxel candidates, it might be that a particle identified by the watershed algorithm in the step described in Section 2.2 is separated again. This is why a cluster detection algorithm on the particle- and defect phase is performed, and particles are relabelled such that each particle and its corresponding defects is composed of only one connected component. Moreover, very small particles are considered as a result of noise in the greyscale image and, therefore, are removed. These two steps correspond to Stage 4 in Figure 2. Throughout this paper, we remove every particle that consists of less than 27 voxels, which is equivalent to a cube of 3 voxels side length.

The final result of the algorithm can be found in Figure 3(i). The particles are labelled with different colours, and the final set of defect voxels is shown in blue.

3 Validation

In this section, we validate the segmentation algorithm using a simulated data set, where we have full information about defects. This allows us to quantify the accuracy of the algorithm, i.e., we can compare which of the (known) defect voxels are actually found by the algorithm. We generate the data sets using a similar procedure as it has also been successfully done for validation purposes in Petrich et al. (2017).

3.1 Generation of simulated data

We use the model described in Westhoff et al. (2017) to generate a completely connected system of particles. The algorithm is based on a decomposition of space into convex polytopes using a so-called Laguerre tessellation, and a subsequent simulation of particles using so-called spherical harmonics. For further details, we refer to Westhoff et al. (2017) and Kuchler et al. (2017) regarding stochastic modeling of active particle systems in anodes and cathodes, respectively. Note that the model parameters have been slightly adjusted to achieve a structure with larger particles and higher volume fraction.

The procedure described above results in a system of pristine particles that can be discretized in a voxel image. In a next step, we add defects to the particles, namely cracks, breakages and holes. The algorithm is as follows. First, we choose 70% of the particles to which a defect will be added, where very small particles (less than 100 voxels in the discretized image) are ignored. Then, for each particle, we decide if a crack in the particle will be simulated (80%), the particle will be broken (10%) or holes will be added to the particle (10%). Those three scenarios are described in the next subsections.

3.1.1 Simulation of cracks

For the simulation of a crack in a particle, we randomly choose two uniformly distributed points q_1 and q_2 at the particle surface, and one point p uniformly

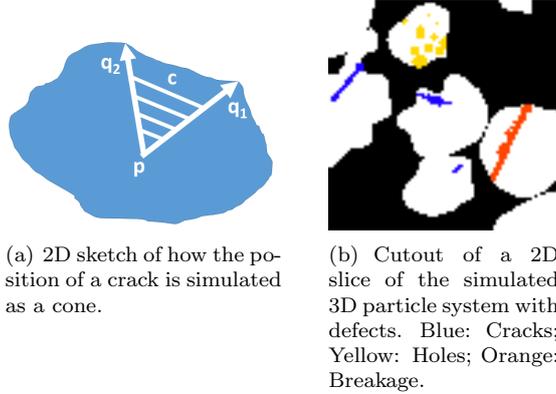


Figure 7: Simulation of a particle system with defects for validation purposes.

distributed in the inner part of the particle. Then the two vectors $v_1 = \overrightarrow{pq_1}$ and $v_2 = \overrightarrow{pq_2}$ describe a convex cone $c = \lambda_1 v_1 + \lambda_2 v_2$, $\lambda_1, \lambda_2 > 0$, see Figure 7(a) for a 2D sketch. All the particle voxels that intersect with this cone are added to the list of crack voxels. However, this only gives cracks that are exactly one voxel thick, which is not realistic considering e.g. Figure 1. This is why we consider a Poisson point process (for details we refer to Chiu et al. (2013)) on the crack with some intensity $\eta_1 > 0$, where $\eta_1 = 0.02$ led to reasonable results. For each point of the Poisson point process, we consider a sphere with radius $\rho \sim U(1, 2)$, i.e., the radius is uniformly distributed between one and two. Each voxel the center of which is contained in such a sphere is added to the list of crack voxels.

3.1.2 Simulation of breakage

Broken particles are generated exactly like cracks, with the only difference that we do not consider a convex cone, but the linear combination $c = \lambda_1 v_1 + \lambda_2 v_2$, $\lambda_1, \lambda_2 \in \mathbb{R}$, i.e., c describes not a cone but a plane in \mathbb{R}^3 . Thus, the particle is split into two parts. Exactly as in the case of cracks, additionally spheres are generated on the plane such that the simulated breakage has a more realistic structure.

3.1.3 Simulation of holes

As it can be observed in Figure 1, holes in particles seem to occur as clusters, i.e., there is usually not only one hole in a particle, but several holes are scattered over the whole particle. This is why we simulate holes in particles using a so-called Boolean model. We consider a Poisson point process with some intensity $\eta_2 > 0$ within the particle, where $\eta_2 = 0.02$ led to reasonable results. Then, a radius, which is uniformly distributed in the interval $(1, 2.5)$ is attached to each point. This results in a system of spheres. Every voxel that is covered by such a sphere is added to the list of hole voxels.

Note that the model for generating defective particle systems is not supposed to represent physical results regarding defects in particle systems, but is only used to produce reasonable input for the segmentation algorithm. A formal validation of the algorithm for simulation of cracks, breakages and holes is not possible as we do not have information about their real shapes a priori (as gaining this information is the goal of the algorithm proposed in the present paper). However, a qualitative validation for tomographic image data is possible, see Figure 7(b) compared to Figure 1. Moreover, the segmentation algorithm is voxel-based and does not depend on the actual shape of cracks / breakages, i.e., it does not assume a specific structure of the defects. Due to the randomness of the simulation algorithm, we make sure that a variety of differently shaped cracks and breakages are generated.

3.2 Results

We apply the segmentation algorithm to a structure generated as described in the previous section. The sampling window consists of $440 \times 440 \times 240$ voxels containing 1338 particles which altogether exhibit 918 different defects, where we only evaluate the results in a window eroded by 20 voxels on each side, because the algorithm might be less precise at the boundary of the sampling window due to missing information. The cubic voxels have a side length of 440 nm. The

Segmented \ Real	Defect voxel	Background voxel
Defect voxel	482 258 (81.84%)	68 545 (00.41%)
Background voxel	107 040 (18.16%)	16 560 502 (99.59%)
Total	589 298 (100%)	16 629 047 (100%)

Table 1: Overview of number of correctly and wrongly identified defect voxels for simulated data set.

parameters are chosen as $h_{\max} = 10000$, $r = 4$, $d_{\max} = 200$, $n_{\max} = 10$ and $c_{\min} = 9$. The marker extent parameter for segmentation in Avizo is chosen to be 1.

In Figure 8, the result of the defect detection algorithm is visualized for a 2D slice of the image stack. Dark-blue voxels show correctly identified defects, light-blue voxels indicate background voxels which are wrongly identified as defects, and orange voxels show defect voxels that are not found by the algorithm. In general, it can be observed that most of the defect voxels are correctly identified, whereas only very few background voxels are wrongly identified as defect voxels. As can be seen in Table 1, the overall amount of correctly identified defect voxels is more than 80%. However, we note that this does not mean that only about 80% of the defects are found. As can be seen in Figure 8, orange voxels (defect voxels that are not found) occur mostly close to dark-blue voxels (correctly identified voxels). This means that most of the defects are detected, but not all the voxels that belong to this defect are found. Thus, we compute the number of defects from which at least one voxel is found. The results are shown in Table 2, where we see that roughly 97% of defects are detected, whereas on average 75% of the voxels per defect are found for cracks and holes, and 68% of voxels per breakage. Finally, we can conclude that the algorithm performs very well on the simulated data set, detecting almost all defects and more than 80% of the overall amount of defect voxels, where only a small amount of background voxels is wrongly identified as defect voxels.

In the next section, the algorithm is applied to tomographic image data of real particle systems from battery electrodes to show that also for real data a reasonable segmentation result regarding visual inspection can be achieved.

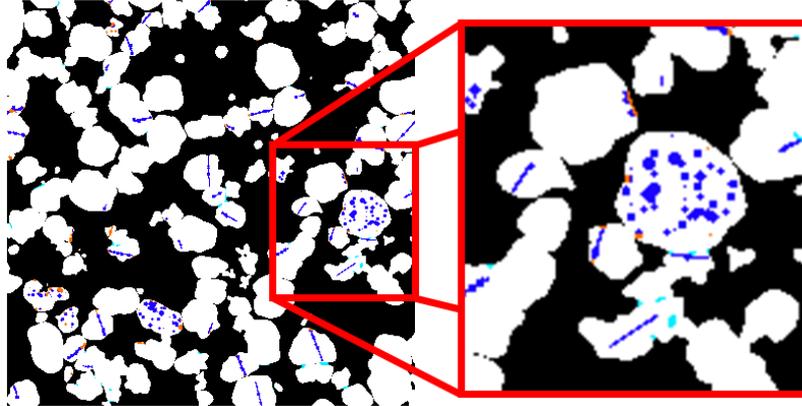


Figure 8: 2D slice of the validation data set. Dark-blue: Correctly identified defect voxels; Orange: Defect voxels that are not found; Light-blue: Background voxels that are wrongly identified as defect voxel.

Defect	Detected at all	Detected voxels on average
Crack	98.15%	76.77%
Breakage	96.25%	68.42%
Holes	97.56%	74.46%

Table 2: Overview of number of detected cracks, breakages and holes as well as on average correctly identified voxels per defect.

4 Application to tomographic image data

In this section, we show how the algorithm works for tomographic image data of real electrodes, where one of the data sets comes from a battery cathode made from LiCoO_2 and the other one from a graphite anode. Note that for these data sets no formal validation is possible, as the defects are unknown, however, we present a comparison of greyscale images and corresponding segmentation to show that defects visible in the greyscale images are recognised in the segmentation. Subsequently, we perform a statistical analysis of the segmented data sets. Note that the surface area calculated in this section is calculated from the segmented tomographic reconstructions which is inherently limited by the resolution of the utilised X-ray CT systems. For this reason, surface area comparisons provide a qualitative insight into the role of microstructural defects and, due to the limited surface texture details that can be captured, should not

be considered to be actual specific surface areas of the material.

4.1 Results for LiCoO_2

The LiCoO_2 sample was extracted from a commercial Li-ion pouch cell that underwent overcharge-induced thermal runaway, as described in Finegan et al. (2016). The particle size distribution of the LiCoO_2 electrode showed a significant reduction in the mean particle diameter following thermal runaway which is evidently caused by the decrepitation of particles (cracking and breakage) when exposed to the rapid increase in temperature (up to $> 1000^\circ\text{C}$). The sample was removed from the cell packaging and imaged in a lab-based CT system (Zeiss Xradia Ultra 810, Carl Zeiss XRM, Pleasanton, CA, USA) with a quasi-monochromatic beam energy of 5.4keV, a pixel size of 63.1 nm, and a field of view of $65\mu\text{m} \times 65\mu\text{m}$. The radiographs were reconstructed using a commercial software package (Zeiss XMReconstructor), which uses an algorithm based on the standard filtered back-projection.

The 3D greyscale image is binarized using the automatic thresholding method from ImageJ, see Schindelin et al. (2012), without any filtering, because this might remove some of the very fine defects. Then the algorithm described in Section 2 is applied, where the parameters are chosen as $h_{\text{max}} = 10,000$, $r = 4$, $d_{\text{max}} = 200$, $n_{\text{max}} = 10$ and $c_{\text{min}} = 9$. The marker extent parameter for segmentation in Avizo is chosen to be 1.

In Figure 9 the result of the algorithm is shown. First, a 2D slice of the 3D greyscale image (Figure 9(a)) is compared to the corresponding 2D slice of the segmentation (Figure 9(b)). The different colours in Figure 9(b) indicate different particles, grey voxels indicate defects. Comparing the result to the greyscale image, we find that defects visible in the greyscale slice are recognised in the segmented data set and the segmentation into individual particles mostly matches how one would label particles manually by visual inspection. In Figure 9(c) a 3D view of the segmented structure is shown together with a view of an individual particle and its defects.

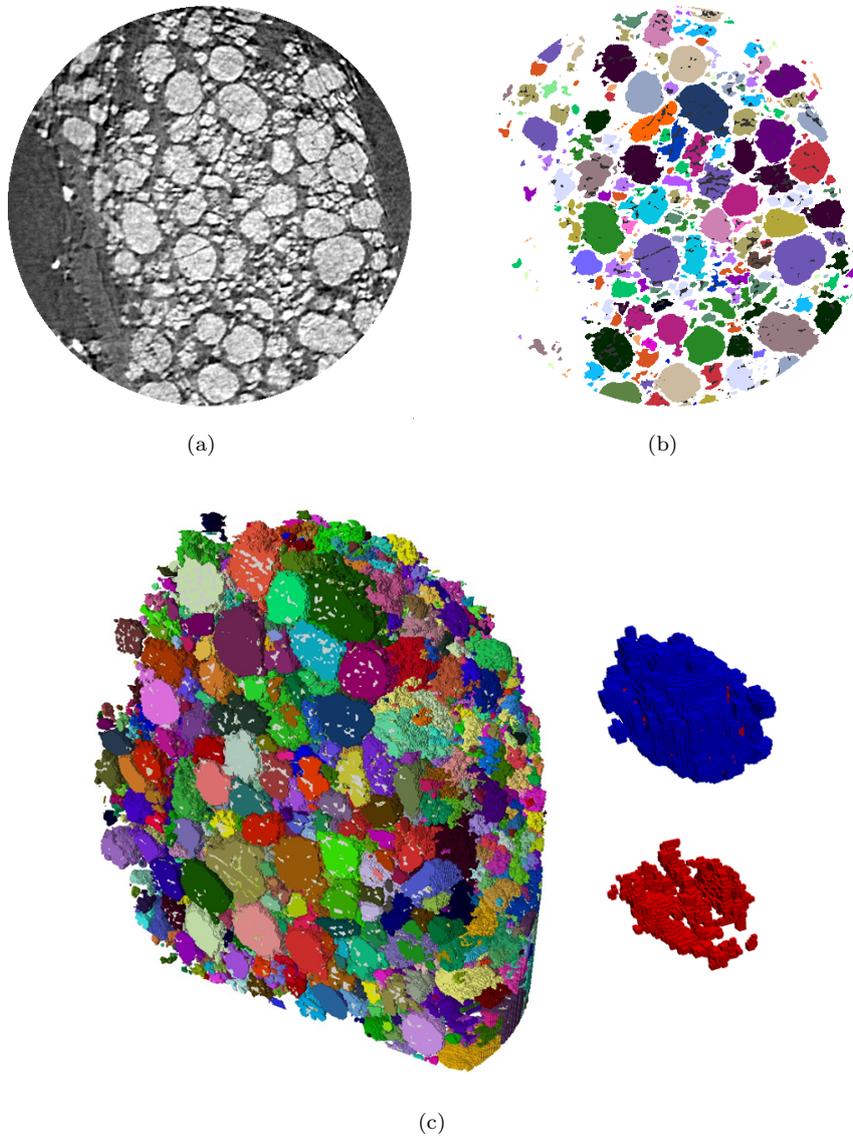


Figure 9: Results for LiCoO_2 data set. (a) Slice of the greyscale image. (b) Corresponding segmentation. Particles are shown in different colours and grey voxels indicate defects. (c) Left: 3D view of the segmented LiCoO_2 data set. Particles are shown in different colours, defects are marked in light grey. Note that several slices at the front have been removed to allow a view “inside” the structure. Right: Solid particle phase (blue) and its defects (red).

Characteristic	Result
Percentage of particles with defect	49.91%
Specific surface area of particle system with defects	$1.31\mu\text{m}^{-1}$
Specific surface area of particle system without defects	$1.09\mu\text{m}^{-1}$
Additional specific surface area induced by cracks	20.77%

Table 3: Analysis of segmented cathode data.

To give a first outlook to possible applications of the algorithm and the information that can be gained, we perform a structural analysis of the segmented image. The results are given in Table 3. First, we find that roughly 50% of the particles exhibit a defect, which is in good accordance with the visual impression. In electrodes, the active surface area of the particles is of high interest, because it significantly influences the ongoing processes. Therefore, we compute the surface area (a) for the 3D image where all defects are added to the particles, i.e., no defects are present, and (b) for the defective particle system, where we only close holes, as they are not reachable from the pore space. The results are also given in Table 3. We found that the defects increase the specific surface area by more than 20%, which would significantly affect the rate capability of the lithium ion cells as well as the reaction rate of the LiCoO_2 during thermal runaway.

4.2 Results for graphite

The graphite sample was extracted from a commercial Li-ion cell in its fresh state. A full ‘separator-to-current collector’ pillar was prepared using a micromachining laser (A-Series Compact Laser Micromachining System, Oxford Lasers Ltd., UK) as described in Bailey et al. (2017). For this analysis a section midway between the surface of the electrode and the current collector was imaged and analysed. The graphite electrode was imaged using both Zernike phase and absorption contrast imaging modes in a lab-based CT system (Zeiss Xradia Ultra 810, Carl Zeiss XRM, Pleasanton, CA, USA) with a quasi-monochromatic beam energy of 5.4keV, a pixel size of 126.2nm, and a field of view of $65\mu\text{m} \times 65\mu\text{m}$. The phase and absorption contrast images were reconstructed using

a commercial software package (Zeiss XMReconstructor) and were thereafter blended to give enhanced image quality, as described in Taiwo et al. (2016). A non-local means filter (Buades et al. (2005)) was applied to the resulting image to reduce noise whilst preserving edges.

The filtered 3D greyscale image is then binarized in the same way as stated in the previous section. Then the algorithm described in Section 2 is applied, where the parameters are chosen as $h_{\max} = 10,000$, $r = 6$, $d_{\max} = 1000$, $n_{\max} = 10$ and $c_{\min} = 9$. The marker extent parameter for segmentation in Avizo is chosen to be 4.

In Figure 10, the result of the segmentation algorithm is shown. First, 2D greyscale slices are compared to the corresponding segmentations. As the graphite particles are not spherical, but have rather the form of flakes, a top view (Figure 10(a) and 10(b)) and a side view (Figure 10(c) and 10(d)) is shown. Again, we find that the defects visible in the greyscale slice are recognised by the algorithm. In Figure 10(e) a 3D view of the segmented structure is shown together with a view of an individual particle and its defects.

Also for the graphite particle system, we perform a statistical analysis of the segmented data set, investigating the same characteristics as in Section 4.1. The results are given in Table 4. The percentage of defective particles is about 18% here, which at first view looks different from the visual impression of Figure 10. However, there are many very small particles which do not exhibit cracks, and rather the larger graphite flakes predominantly have cracks. The influence of cracks on the active surface area of the graphite is much greater than in the case of the LiCoO_2 due to its flake-like microstructure, contributing to more than a 70% increase in surface area when compared to the equivalent microstructure without cracks.

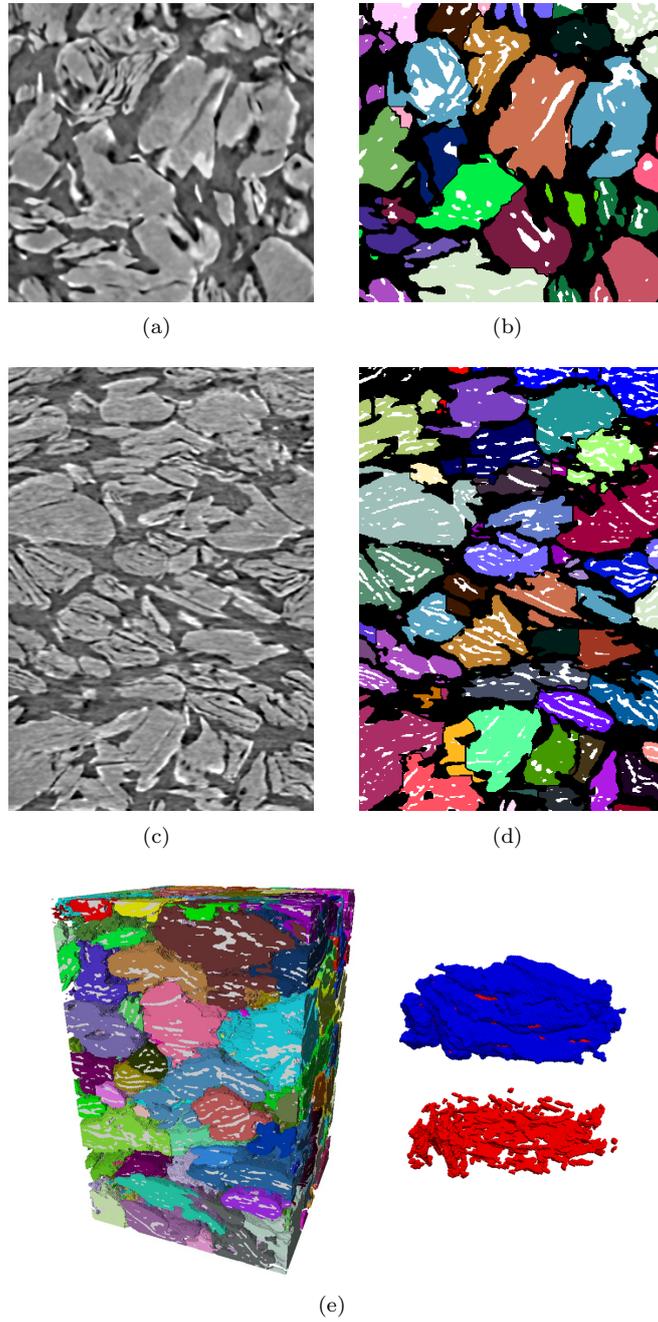


Figure 10: Results for graphite data set. (a) Top view on a slice of the greyscale image. (b) Segmentation corresponding to (a). Particles are shown in different colours and white voxels indicate defects. (c) Side view on a slice of the greyscale image. (d) Segmentation corresponding to (c). (e) Left: 3D view of the segmented graphite data set. Particles are shown in different colours, defects are marked in light grey. Note that several slices on each side have been removed to exclude boundary effects. Right: Solid particle phase (blue) and its defects (red).

Characteristic	Result
Percentage of particles with defect	18.31%
Specific surface area of particle system with defects	$1.14\mu\text{m}^{-1}$
Specific surface area of particle system without defects	$0.66\mu\text{m}^{-1}$
Additional specific surface area induced by cracks	71.69%

Table 4: Analysis of segmented graphite data.

5 Conclusions

We have developed a structural segmentation algorithm for image data that detects cracks, breakages and holes in particulate systems and performs a segmentation into individual particles, where fragments of one and the same particle are recognised accordingly. The algorithm has been validated using simulated data from a stochastic 3D microstructure model, where positions of defects are known. This allows a quantification of the accuracy of the algorithm. Finally, the method has been applied to tomographic image data from a battery anode and cathode, which exhibits its broad applicability for data sets with different defects with a reasonable range of sizes. If very fine and much bigger defects occur in one and the same structure, the algorithm might have problems and it could be helpful to consider both scales individually.

A crucial point of the algorithm presented, as with most advanced segmentation algorithms, is the choice of parameters. Usually, there is no obvious decision rule and good parameters are found using rather a trial-and-error method and visual inspection. However, a possible improvement could be using stochastic 3D microstructure models like described in Section 3.1 and generating virtual particle systems with similar properties like the data under consideration. Then, it is possible to choose appropriate parameters of the segmentation algorithm by optimizing with respect to the (evaluable) accuracy on simulated data. This is one possibility to further develop the methodology described in the present paper. Moreover, the structural segmentation algorithm allows for a comprehensive analysis of defects in particle systems. In particular, the spatial frequency of defects is of high interest. Furthermore, defects in lithium-ion battery elec-

trodes do not only occur during operation, but already during the production process, especially the calendering process. A corresponding analysis is subject of further research.

Acknowledgement

D.W. gratefully acknowledges travel funding by the Science and Technology Facilities Council (STFC) in the STFC Futures Early Career Award programme ST/N002385/1.

References

- Antonyuk, S., Khanal, M., Tomas, J., Heinrich, S., and Mörl, L. (2006). Impact breakage of spherical granules: Experimental study and DEM simulation. *Chemical Engineering and Processing: Process Intensification*, 45(10):838–856.
- Bailey, J. J., Heenan, T. M. M., Finegan, D. P., Lu, X., Daemi, S. D., Iacoviello, F., Backeberg, N. R., Taiwo, O. O., Brett, D. J. L., Atkinson, A., and Shearing, P. R. (2017). Laser-preparation of geometrically optimised samples for X-ray nano-CT. *Journal of Microscopy*, 267(3):384–396.
- Buades, A., Coll, B., and Morel, J.-M. (2005). A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 4(2):490–530.
- Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). *Stochastic Geometry and its Applications*. J. Wiley & Sons, Chichester, 3rd edition.
- Finegan, D. P., Scheel, M., Robinson, J. B., Tjaden, B., Hunt, I., Mason, T. J., Millichamp, J., Michiel, M. D., Offer, G. J., Hinds, G., Brett, D. J., and Shearing, P. R. (2015). In-operando high-speed tomography of lithium-ion batteries during thermal runaway. *Nature Communications*, 6:6924.

- Finegan, D. P., Scheel, M., Robinson, J. B., Tjaden, B., Michiel, M. D., Hinds, G., Brett, D. J. L., and Shearing, P. R. (2016). Investigating lithium-ion battery materials during overcharge-induced thermal runaway: an operando and multi-scale X-ray CT study. *Physical Chemistry Chemical Physics*, 18:30912–30919.
- Gaiselmann, G., Neumann, M., Schmidt, V., Pecho, O., Hocker, T., and Holzer, L. (2014). Quantitative relationships between microstructure and effective transport properties based on virtual materials testing. *AIChE Journal*, 60(6):1983–1999.
- Gillibert, L. and Jeulin, D. (2013). 3D reconstruction and analysis of the fragmented grains in a composite material. *Image Analysis & Stereology*, 32(2):107–115.
- Hoshen, J. and Kopelman, R. (1976). Percolation and cluster distribution. I. cluster multiple labeling technique and critical concentration algorithm. *Physical Review B*, 14(8):3438–3445.
- Kuchler, K., Westhoff, D., Feinauer, J., Mitsch, T., Manke, I., and Schmidt, V. (2017). Stochastic model for the 3D microstructure of pristine and cyclically aged cathodes in Li-ion batteries. *submitted*.
- Mayer, J., Schmidt, V., and Schweiggert, F. (2004). A unified simulation framework for spatial stochastic models. *Simulation Modelling Practice and Theory*, 12:307–326.
- Petrich, L., Westhoff, D., Feinauer, J., Finegan, D. P., Daemi, S. R., Shearing, P. R., and Schmidt, V. (2017). Crack detection in lithium-ion cells using machine learning. *Computational Materials Science*, 136:297 – 305.
- Roerdink, J. B. T. M. and Meijster, A. (2001). The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228.

- Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, B. S., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., and Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9:676–682.
- Spettl, A., Wimmer, R., Werz, T., Heinze, H., Odenbach, S., Krill, C., and Schmidt, V. (2015). Stochastic 3D modeling of ostwald ripening at ultra-high volume fractions of the coarsening phase. *Modelling and Simulation in Materials Science and Engineering*, 23(6):065001.
- Taiwo, O. O., Finegan, D. P., Gelb, J., Holzner, C., Brett, D. J., and Shearing, P. R. (2016). The use of contrast enhancement techniques in X-ray imaging of lithium-ion battery electrodes. *Chemical Engineering Science*, 154:27 – 33.
- Torquato, S. (2013). *Random Heterogeneous Materials*. Springer, New York.
- VSG - Visualization Sciences Group - Avizo Standard (2017). <http://www.vsg3d.com/>.
- Westhoff, D., Feinauer, J., Kuchler, K., Mitsch, T., Manke, I., Latz, A., and Schmidt, V. (2017). Parametric stochastic 3D model for the microstructure of anodes in lithium-ion power cells. *Computational Materials Science*, 126:435–467.