Article

Virtual reassembling of 3D fragments for the data-driven analysis of fracture mechanisms in multi-component materials

Thomas Wilhelm[1],* , Trang Thu Võ[2], Orkun Furat[1], Urs A. Peuker[2] and Volker Schmidt[1]

[1] Institute of Stochastics, Ulm University, 89069 Ulm, Germany
[2] Institute of Process Engineering and Mineral Processing, TU Bergakademie Freiberg, 09599 Freiberg, Germany
* Correspondence: thomas.wilhelm@uni-ulm.de

Abstract: This paper introduces a novel method for characterizing fracture mechanisms in multi-component materials using 3D image data gained by X-ray computed tomography (CT) measurements. In mineral liberation, the understanding of these mechanisms is crucial, particularly whether fractures occur along the boundaries of mineral phases (intergranular fracture) and/or within mineral phases (transgranular fracture). Conventional techniques for analyzing fracture mechanisms are focused on globally comparing the surface exposure of mineral phases extracted from image measurements before and after fracture. Instead, we present a virtual reassembling algorithm based on image registration techniques, which is applied to 3D data of multi-component materials before and after fracture in order to determine and characterize the individual fracture surfaces. This enables us to conduct a local quantitative analysis of fracture mechanisms by voxelwise comparing adjacent regions at fracture surfaces. A quantitative analysis of fracture mechanisms is especially important in the context of recycling processes. As primary deposits are decreasing worldwide and the focus is shifting on reducing wast, the interest in secondary raw materials has increased. However, lower-concentrated, but valuable materials are often overlooked in recycling routes designed for higher-concentration materials and dissipate. Therefore, efforts are being made to enrich valuable elements, such as lithium, as engineered artificial minerals through pyrometallurgical processes. The subsequent liberation through comminution processes, such as crushing, is essential for the recovery of valuable minerals. A better understanding of crushing processes, especially fracture mechanisms in slags, is crucial for the success of recycling. The reassembling algorithm presented in this paper is evaluated through a simulation study, followed by an application to a naturally occurring ore and a slag resulting from a recycling process.

Keywords: Fracture mechanism; 3D X-ray computed tomography; Image preprocessing; Statistical micostructure analysis; Reassembling algorithm; Mineral liberation

---

1. Introduction

Fracture mechanisms causing mineral liberation were initially investigated by Gaudin [1], who differentiated between fracture along the boundaries of mineral phases (intergranular fracture) and random fracture (transgranular fracture). The latter is often considered to be the predominant mechanism for liberation as it is not influenced by mineral properties, like hardness differences of various minerals phases [2], nor by mineral morphology [3]. However, in reality, pure transgranular fracture does not occur, but fracture occurs as a combination of transgranular and intergranular fracture [4]. Recently, Mariano et al. [5] provided a summary of the definitions of random fracture used by various authors. In addition to investigating the complex interplay of various factors that influence fracture, several methods are proposed in the literature for quantifying fracture mechanisms in comminution processes. For example, Little et al. [6] demonstrated the presence of intergranular fracture in ores by analyzing the preservation of the shape of mineral phases and comparing the degree of liberation. More recently, Leißner et al. [7], Mirzaei and Khalesi [8] have introduced quantitative approaches to determine the fraction of intergranular and transgranular fracture based on 2D analysis of mineral surface exposure. However, 3D data are essential for adequate quantification of fracture mechanisms. A first step in this direction was taken in [9], where fracture mechanisms were investigated using 3D X-ray computed tomography measurements, focusing on the analysis of interfacial areas between copper mineral particles and host rock, computing the specific interfacial fraction before and after fracture. Nevertheless, these studies primarily focus on the global characterization of fracture mechanisms by comparing phase-based descriptors such as surface exposure or interfacial area before and after fracture.

In the present paper, we propose a virtual reassembling algorithm based on image registration techniques utilizing 3D CT measurements of multi-component materials before and after fracture. More precisely, before fracture a single non-broken particle is visualized in a CT image and, after fracture, another CT image shows how the particle has been broken into smaller fragments. By reassembling the non-broken particle from fragments, we can voxelwise

extract the fracture surfaces. This detailed information about the fracture surfaces is essential for the quantitative characterization of fracture mechanisms. In particular, this method can be used to locally determine whether fractures occur predominantly intergranular, transgranular or as a superposition of these fracture mechanisms by comparing adjacent regions at fracture surfaces. In more detail, in the reassembling algorithm we employ an image registration technique that involves translation and rotation of a given pattern (so-called moving image) within a larger image (so-called fixed image) [10]. In our case, the moving image corresponds to a 3D image of a fragment, whereas the fixed image represents the non-broken particle. Using the reassembling algorithm, we iteratively reassemble the individual fragments, where we consider a specific order on how to reassemble the fragments. After reassembling it is possible to extract the fracture surfaces as regions, where two reassembled fragments are adjacent and to transfer these fracture surfaces to the non-broken particle. By characterizing the 3D microstructure of the non-broken particle in this way, i.e., knowing the positions of fracture surfaces within the non-broken particle, we are able to voxelwise specify if there is an intergranular or transgranular fracture. The workflow of this kind of a local quantitative analysis of breaking mechanisms is sketched in Figure 1.
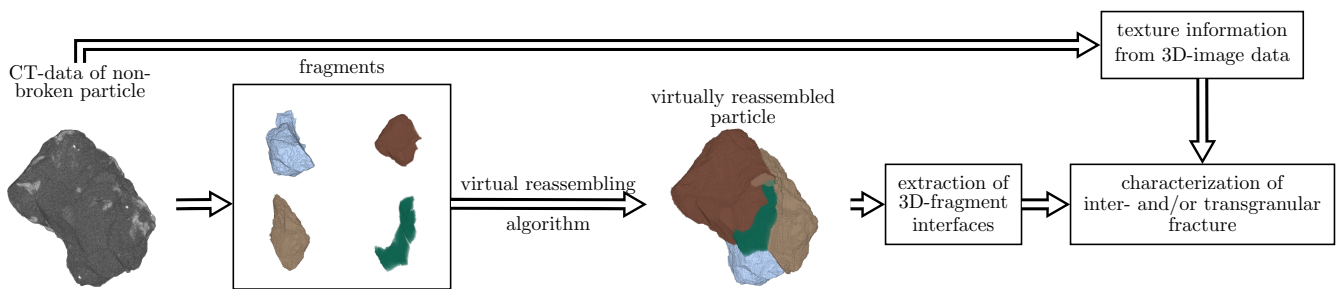

Figure 1. Workflow of the virtual reassembling of 3D fragments for the data-driven analysis of fracture mechanisms in multi-component materials. Initially, various fragments are reassembled to determine the interfaces between the fragments (fracture surfaces). These surfaces, along with the texture information (spatial mineralogical composition) from 3D measurements of the non-broken particle, are then utilized to locally analyze the presence of intergranular or transgranular fracture.

In the literature, there exist various approaches for image registration, with some involving the placement of markers on both (moving and fixed) images, followed by an alignment of the markers [11]. Other methods include matching extracted features or surfaces [12]. Recently, much attention has been directed towards intensity-based approaches, where intensity values are utilized to compute similarity measures between two images [13]. Intensity-based registration typically does not require extensive preprocessing, such as segmentation or feature extraction. In the present paper, we employ an iterative 3D intensity-based approach for the registration of a fragment's image, considering six degrees of freedom. This involves finding the translations in three directions and rotations around the three axes for each fragment, where the translation of a single fragment is determined by means of a mask-based image registration approach, which is implemented via fast Fourier transformation (FFT) [14]. Thus, this approach involves so-called masking, which is essential in the registration of fragments. It ensures that regions in the image of the non-broken particle and regions in the image of the fragments, which can wrongly influence the registration result, are ignored when computing the similarity measure to evaluate the registration result. Masked images are used to consider only those regions in the images where the non-broken particle body and the fragments are present, i.e., the background is ignored in both images. Optimal rotations are determined by means of global particle swarm optimization (PSO)[15,16]. This technique has proven successful in biomedical image registration, as demonstrated by Wachowiak et al. [17]. Furthermore, for evaluating how closely intensity values of the moving image match the intensity values of the fixed image, we consider a normalized cross-correlation coefficient as similarity measure. Registration in the Fourier domain using normalized cross-correlation is well-suited due to its robustness and short computation time, which has been extensively studied in the literature [13,14,18–21].

In the present paper, we first evaluate the goodness of fit of the reassembling algorithm by means of a simulation study. Here, we utilize CT data of a non-broken particle and virtually generate fragments using distance-based tessellation models, which are commonly employed for creating random grain architectures of polycrystalline materials [22–24]. We adapt these models to generate a grain architecture within the non-broken particle by employing the Euclidean distance as the tessellation distance, which results in a simplified tessellation also referred to as a Voronoi diagram or Voronoi tessellation. The cells of the Voronoi tessellation define a grain architecture, where a subset of the resulting cells is used as generated fragments. This fragment generation allows us to know the original positions of the fragments within the non-broken particle, serving as ground truth for evaluating the reassembling

algorithm. After applying this algorithm to the fragments obtained from the Voronoi tessellation, we compare the algorithm's output with the ground truth. This comparative analysis enables us to assess the accuracy and reliability of the algorithm in reconstructing the fragments and, thus, obtaining results consistent with the original tessellation.
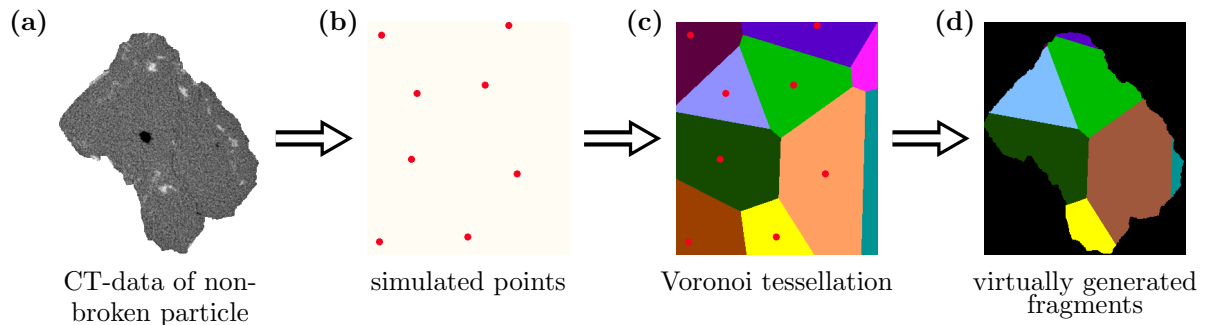

Figure 2. Workflow of virtual fragment generation. Initially, a point pattern (b) is generated within the sampling window of the non-broken particle (a). Subsequently, a Voronoi tessellation is constructed based on this point pattern, resulting in a decomposition of the sampling window into cells (c). A subset of these cells serve as virtually generated fragments (d).

A quantitative analysis of fracture mechanisms is especially important in the context of recycling processes. In view of the progressive depletion of primary deposits on the world market and the requirement to reduce wast, there is growing interest in the development of resources from secondary raw materials. However, they often contain low concentrations of valuable elements, with lithium being a prime example [25]. Nevertheless, the increased attention on secondary raw materials stems from the need to explore alternative sources and implement sustainable practices to meet the increasing demand for valuable elements in various industries. This shift is a strategic response to the challenges posed by the decreasing availability of primary deposits and the need to secure a more sustainable resource base. A well-known approach to extract valuable elements from secondary raw materials involves enriching them as engineered artificial minerals in the slag phase through pyrometallurgical processes. A comprehensive understanding of generation and comminution processes of slags for the optimization of subsequent separation processes, aimed at enriching valuable minerals, requires insight into the fracture mechanisms within the slag. Moreover, the understanding of fracture mechanisms in naturally occurring multi-component materials is also of great interest, particularly in the mining industry, since also here the aim is to extract valuable materials, e.g. from ores. Therefore, we apply the reassembly algorithm, proposed in the present paper, to two different kinds of multi-component materials, where we characterize the fracture mechanisms of a greisen ore as well as of a slag produced in a real pyrometallurgical process.

The rest of this paper is organized as follows. In Section 2, we provide a description of the particulate multi-component materials considered in this paper and how these materials are fractured by in-situ experiments. Section 3 outlines the segmentation methods employed, including the separation of foreground (particle phase) from the background by means of binarization. Subsequently, a phase-based segmentation is performed to extract and analyze different mineral phases observed in CT image data. Section 4 presents the reassembling algorithm, which is evaluated in a simulation study by means of goodness-of-fit measures, see Section 5. Then, in Section 6, the reassembling algorithm is applied to two different multi-component materials, where their fracture mechanisms are analyzed through a local quantitative fracture characterization. Section 7 concludes.

2. Materials and sample preparation

In this paper, a greisen ore from the Zinnwald deposit in the Erzgebirge region is considered for characterizing the fracture mechanisms in natural occurring multi-component materials. The main component in this ore is quartz, followed by zinnwaldite, topaz, muscovite, and traces of further minerals. The greisen ore was crushed using a jaw crusher and a cone crusher, and subsequently, it was further reduced to particles with sizes of approximately 4 mm using a flat cone crusher, followed by sieving and dividing. A particle with a size from the $4 - 5$ mm size fraction was selected for in-situ compression, described below. Notably, the greisen ore does not exhibit dendritic structures.

Additionally, a lithium-aluminum slag was chosen for in-situ compression experiments, especially for its unique manufacturing and cooling process conducted at the IME Process Metallurgy and Metal Recycling - RWTH Aachen. At first, a selection of salts has been heated to their melting temperature, allowing for the homogenization of the melt. Then, the melt underwent controlled cooling at a rate of $50 \, \mathrm{K} \, \mathrm{h}^{-1}$. The resulting slag exhibits a dendritic microstructure which indicates that the cooling rate was too large for a controlled crystallization and enrichment

of specific minerals. Thus, the microstructure of the slag differs significantly from the naturally occurring greisen ore, which results from slower cooling rates. Analogously to the greisen-ore particle, a slag particle with a size of $4 - 5$ mm was selected for in-situ compression measurements.

In order to acquire 3D image data of the non-broken greisen/slag particles and the fragments after fracture, a load cell was coupled with a CT measuring device. More precisely, the CT5000 5 kN in-situ load cell from Deben UK, customized for the Zeiss Xradia Versa 510 CT system, was utilized. The non-broken particle is placed in the load cell between ceramic pistons that are held in position by a guiding tube and measured at 5 N, a force selected to ensure particle stability and prevent movement during the CT measurement. The particle is gradually loaded until a drop in force in the force-displacement curve and a change in X-ray transmission can be seen in the live projection image, indicating that the particle is broken. Following the force drop, the stress on the particle will be stopped only when the force increases again to ensure that the crack remains opened. Subsequently, another CT measurement is conducted. The measurement and reconstruction parameters for an in-situ measurement series, corresponding to either the ore or slag particle, have been chosen consistently, see Table 1.

Table 1. CT measurement and reconstruction settings for in-situ compression of ore and slag particle

	ore		slag	
measurement settings	non-broken	broken	non-broken	broken
source distance in mm	70	70	70	70
detector distance in mm	200	200	200	200
optical magnification	0.4 x	0.4 x	0.4 x	0.4 x
acceleration voltage in kV	80	80	80	80
electrical power in W	7	7	7	7
source filter (Zeiss standard)	LE4	LE4	LE4	LE4
voxel size in µm	17.8	17.8	17.8	17.8
camera binning	2	2	2	2
number of projections	401	401	1601	1601
exposure time in s	5	5	5	5
angle range in °	360	360	360	360
reconstruction settings				
reconstruction algorithm	FBP	FBP	FBP	FBP
center shift	-0.56	-0.48	-0.866	-0.804
defect correction	none	none	none	none
byte scaling	$(-0.02, 0.057)$	$(-0.015, 0.06)$	$(-0.003670, 0.0411)$	$(-0.003670, 0.0411)$
beam hardening constant	0.05	0.05	0.02	0.02

3. Image processing

A quantitative characterization of the 3D microstructure of a particle before fracture and, thus, the characterization of the fracture mechanisms cannot be performed directly on image data, which is obtained from CT measurements as described in Section 2. Some image preprocessing is required for further characterization. First, we apply a binarization step to the CT image data of the particle before and after fracture to separate the foreground (particle) from the background in the CT images (Section 3.1). This is followed by a phase-based segmentation step to extract mineral phases from the non-broken particle (Section 3.2). Then we apply a watershed-based segmentation step to the image data of the particles after fracture to obtain an image of each fragment (Section 3.3), as the fragments are generally connected with each other in the raw image data of the broken particle, i.e., they are present as one contiguous region.

For each multi-component particle we obtained grayscale images by CT measurements before and after fracture, which are formally defined as mappings $I_{intact} : W_{intact} \rightarrow \{0, \ldots, 65535\}$ and $I_{broken} : W_{broken} \rightarrow \{0, \ldots, 65535\}$, where the sets W_{intact} and W_{broken} of voxels are finite subsets of $\mathbb{Z}^3$. The mapping I_{intact} assigns each voxel $x \in W_{intact}$ to its respective grayscale value $I_{intact}(x)$. Analogously, I_{broken} assigns each voxel $x \in W_{broken}$ to its respective grayscale value $I_{broken}(x)$.

3.1. Particle-based segmentation

To separate the foreground (particle) from the background in the CT images before and after fracture, we employ a multi-step segmentation process. Initially, we reduce noise in the CT images by applying a Gaussian filter with a standard deviation of 0.2. Subsequently, an unsharp mask filter [26] is applied to enhance edges in the image. Then, each image is binarized using the ISODATA thresholding method [27,28] to distinguish foreground from background. After binarization, we obtain an image $M_{\text{intact}} : W_{\text{intact}} \to \{0,1\}$ for the non-broken particle, which assigns each voxel either to the foreground or background of the particle, i.e., M_{intact} is given by

$$M_{\text{intact}}(x) = \begin{cases} 1, & \text{if } x \text{ belongs to the foreground,} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

for each $x \in W_{\text{intact}}$. Analogously, we obtain the image $M_{\text{broken}} : W_{\text{broken}} \to \{0,1\}$, which assigns each voxel either to the foreground or background of the broken particle. Furthermore, denote the sets of voxels, which are assigned to the foreground of the non-broken and broken particle by $W'_{\text{intact}} = \{x \in W_{\text{intact}} : M_{\text{intact}}(x) = 1\}$ and $W'_{\text{broken}} = \{x \in W_{\text{broken}} : M_{\text{broken}}(x) = 1\}$, respectively. Furthermore after binarization, we set $I_{\text{intact}}(x) = 0$ for all voxels $x \in W_{\text{intact}}$ for which $M_{\text{intact}}(x) = 0$. Analogously, we set $I_{\text{broken}}(x) = 0$ for all voxels $x \in W_{\text{broken}}$ for which $M_{\text{broken}}(x) = 0$.

This allows us to check if the extracted non-broken particle and the fragments of the broken particle have the same volume. Note that a large discrepancy between these volumes would indicate an insufficient preprocessing result or missing fragments. Therefore, we computed the volume of the extracted non-broken particle as well as the volume of the fragments of the broken particle, and checked if $|W'_{\text{broken}}|/|W'_{\text{intact}}| \approx 1$, where $|\cdot|$ denotes cardinality. It turned out that $|W'_{\text{broken}}|/|W'_{\text{intact}}| = 0.993$ for the ore particle, and $|W'_{\text{broken}}|/|W'_{\text{intact}}| = 0.995$ for the slag particle, which means that the image measurement and processing steps discussed so far seem to be correct. Minor disparities in particle volumes before and after fracture may arise from partial volume effects and the potential absence of small fragments that are not captured in the CT images after fracture. Figures 3a and 3c show 2D slices of the CT images before fracture of both multi-component particles after binarization.

3.2. Phase-based segmentation

This preprocessing step enables the extraction of different mineral phases, where we assume that they can be identified by means of the grayscale values observed in the CT images, facilitating a quantitative analysis of the phase-based fracture mechanisms. To extract different mineral phases from the image measurements of the particles before fracture, we employ a k-means clustering algorithm [29]. Here, for each $x \in W_{\text{intact}}$ the grayscale values of the image I_{intact} in a $3 \times 3 \times 3$ neighborhood are considered as features for identifying $k > 1$ clusters using the k-means algorithm. This clustering technique assigns each foreground voxel its corresponding label from the set $\{1, \ldots, k\}$, based on its feature vector, i.e., the grayscale values in the voxel's $3 \times 3 \times 3$ neighborhood [30]. Each label corresponds to a distinct mineral phase present in the non-broken particle. The number k of clusters for each dataset is manually chosen based on visual inspection.

In addition to the mineral phases within the slag body, there are also air bubbles present. Initially identified as part of the background, these air bubbles are subsequently labeled as an additionally phase in the phase-based segmentation. Note that due to partial volume effects, the grayscale values of voxels located close to the background are influenced by neighboring particles or the background. Thus, this makes it challenging to correctly label mineral phases especially at the boundary of particles. Nevertheless, in the present paper we provide a phase-based segmentation for identifying mineral phases, which can be further improved incorporating other measurement techniques like diffraction contrast tomography or correlative tomography [31].

After extracting $n_{\text{phases}} \geq 1$ different mineral phases, we obtain a phase-based segmentation $I_{\text{phases}} : W_{\text{intact}} \to \{0, 1, \ldots, n_{\text{phases}}\}$ of I_{intact}, which assigns each voxel a mineral phase or the background. More precisely, I_{phases} is given by

$$I_{\text{phases}}(x) = \begin{cases} i, & \text{if } x \text{ belongs to phase } i \in \{1, 2, \ldots, n_{\text{phases}}\}, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

for each $x \in W_{\text{intact}}$. In Figures 3b and 3d both particles are visualized, where voxels assigned to different phases are visualized by different colors. For both multi-component particles considered in this paper, we observed three

different clusters in the CT image data, where we assume that these clusters correspond to three different mineral phases, i.e., $n_{\text{phases}} = 3$. For each $i \in \{1, 2, 3\}$, the i-th phase P_i is then given by $P_i = \{x \in W_{\text{intact}} : I_{\text{phases}}(x) = i\}$.

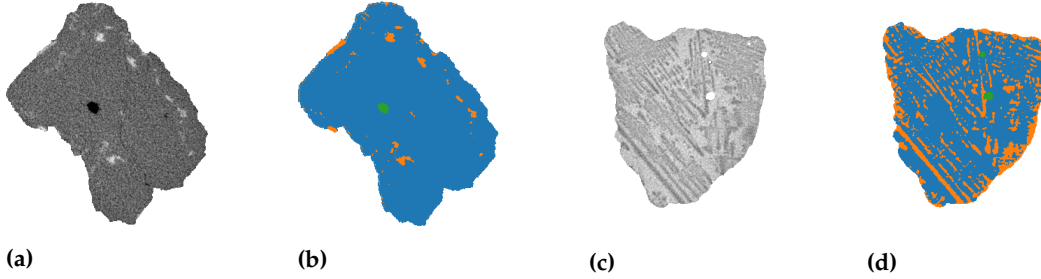| **(a)** | **(b)** | **(c)** | **(d)** |

Figure 3. 2D slices of CT images depicting multi-component particles before fracture (a: ore, c: slag) and after binarization with various mineral phases indicated by colors for the ore (b) and slag particle (d). Three different mineral phases have been identified for both particles, where phase P_1 is colored in blue, phase P_2 in orange and phase P_3 in green.

Using the phase-based segmentation stated above, a particle can be characterized by aggregated descriptors such as the volume fractions of its mineral phases, which are crucial for quantifying the transgranular and intergranular fracture mechanisms. For each $i \in \{1, 2, 3\}$, the volume fraction $\rho_i \in [0, 1]$ of the i-th phase is given by

$$\rho_i = \frac{|P_i|}{|\{x \in W_{\text{intact}} : I_{\text{phases}}(x) > 0\}|}. \tag{3}$$

In our case, both particles consist mainly of a single phase, called P_1 in the following, which has the largest volume fraction, see Table 2. The slag particle additionally consists of a second phase P_2 with around 20% of the particle's volume, whereas the phases P_2 and P_3 of the ore particle only share very small volume fractions. In the slag particle, phase P_3 corresponds to air bubbles, whereas in the ore particle no air bubbles have been detected. Although we denote the three phases of the ore and slag particles by the same symbol (i.e., P_1, P_2 and P_3), this does not mean that these phases correspond in both particles to the same mineral.

Table 2. Volume fractions of mineral phases in the ore and slag particles, observed in image measurements before fracture.

	phase P_1 (blue)	phase P_2 (orange)	phase P_3 (green)
ρ_i (ore)	0.96	0.04	0.001
ρ_i (slag)	0.78	0.22	0.006

3.3. Fragment-based segmentation

Grayscale images of broken particles do not directly provide a segmentation into individual fragments, because the fragments are partially interconnected, making it impossible to identify single fragments as connected components. Therefore, we employ a combination of the watershed algorithm [32–35] and a 3D morphological reconstruction approach [36,37] to segment grayscale images of broken particles into individual fragments while simultaneously preventing oversegmentation. In comparison to techniques which are commonly used in the literature to prevent oversegmentation, like extended regional minima [38] or post-processing steps involving dilation [39], the advantage of our approach is that it only requires manual tuning of a single parameter.

The output of our segmentation procedure is an image $S: W_{\text{broken}} \to \{0, \ldots, n_F\}$ where $n_F \geq 1$ denotes the number of fragments and S assigns each voxel to the i-th fragment or the background. More precisely, it holds that

$$S(x) = \begin{cases} i, & \text{if } x \text{ belongs to the } i\text{-th fragment for some } i \in \{1, 2, \ldots, n_F\}, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

for each $x \in W_{\text{broken}}$. Then, we get n_F fragments $F_1, \ldots, F_{n_F} \subset W_{\text{broken}}$, where $F_i = \{x \in W_{\text{broken}} : S(x) = i\}$ for each $i = 1, \ldots, n_F$. For the purpose of reassembling a fragment $F \in \mathcal{F} = \{F_1, \ldots, F_{n_F}\}$ in Section 4, we need the

grayscale value $I_{\text{broken}}(x)$ for each voxel $x \in F$. Therefore, we define the fragment image $I_F \colon W_F \to \{0, 1, \ldots, 65565\}$ by cropping, i.e.,

$$
I_F(x) = \begin{cases} I_{\text{broken}}(x), & \text{if } x \in F, \\ 0, & \text{otherwise,} \end{cases}
\tag{5}
$$

for each $x \in W_F$, where $W_F \subset \mathbb{Z}^3$ denotes the smallest cuboidal sampling window which contains the fragment F. Thus, the mapping I_F, assigning each voxel of $x \in F$ the corresponding grayscale value of I_{broken}, can be considered to be a cutout of I_{broken} representing the fragment F. Furthermore, we consider the mask $M_F \colon W_F \to \{0, 1\}$ of the fragment F, where $M_F(x) = 1$ for each $x \in F$, and $M_F(x) = 0$ for $x \in W_F \setminus F$.

4. Virtual reassembling algorithm

This section outlines the virtual reassembling algorithm which is used to virtually arrange a family $\mathcal{F} = \{F_1, \ldots, F_{n_F}\}$ of fragment images to a new image, reassembling the reference image I_{intact} of a non-broken particle. In the reassembling process, the fragment image I_F is iteratively registered by aligning I_F with the reference image I_{intact} for each $F \in \mathcal{F}$. This alignment involves finding a transformation that minimizes the discrepancy between the transformed fragment image and the reference image of the non-broken particle. More precisely, we determine the translation and rotation of fragment images $I_{\mathcal{F}} = \{I_F\}_{F \in \mathcal{F}}$ with masks $M_{\mathcal{F}} = \{M_F\}_{F \in \mathcal{F}}$, taking into account that during fracture, a fragment $F \in \mathcal{F}$ may shift and rotate away from its original position in the non-broken particle, where the transformation of a fragment image I_F, which aligns it within I_{intact}, is determined by applying a so-called mask-based image registration approach. In the present paper, we use an intensity-based image registration approach [13] extended with PSO [15,16], which has been successfully utilized in biomedical image registration [17] for the registration of single pairs of 3D image data. We extend this approach by not only registering two images, but by proposing a reassembly algorithm, which iteratively registers the images of a family of (possibly multiple) fragment images with a reference image that is updated during reassembly. Furthermore, we consider masking in the image registration approach by applying FFT as described by Padfield [14].

The rest of this section is divided into two parts. In Section 4.1. to make our paper more self-contained, the mask-based image registration approach is explained for a general setup. Then, in Section 4.2 we describe how the reassembling algorithm iteratively utilizes the mask-based image registration approach in order to reassemble a set of fragments.

4.1. Mask-based image registration

In order to determine the translation and rotation of an image with respect to a reference image, image registration algorithms can be utilized [17], where a moving image $I^{\text{m}} \colon W^{\text{m}} \to \{0, \ldots, 65535\}$ (corresponding to a fragment image I_F) and a fixed image $I^{\text{f}} \colon W^{\text{f}} \to \{0, \ldots, 65535\}$ (corresponding to the image I_{intact} of the non-broken particle) are registered. Here, the fixed image I^{f} is observed within a certain sampling window $W^{\text{f}} \subset \mathbb{Z}^3$, and the moving image I^{m} within $W^{\text{m}} \subset \mathbb{Z}^3$.

An important step in implementing image registration algorithms is to determine a suitable function $T \colon \mathbb{R}^3 \to \mathbb{R}^3$, which maps a point $x^{\text{m}} \in \mathbb{R}^3$ to a point $x^{\text{f}} \in \mathbb{R}^3$, i.e., $T(x^{\text{m}}) = x^{\text{f}}$. Such functions can be used to define transformed images, where, in the context of image registration, the transformed moving image should resemble the fixed image. As we consider translation and rotation operations in the registration process, the function T is a superposition of a function $T_t \colon \mathbb{R}^3 \to \mathbb{R}^3$ which shifts a point $x \in \mathbb{R}^3$ by a translation vector $t \in \mathbb{R}^3$ to the new position $x + t \in \mathbb{R}^3$, and a function $R_\theta \colon \mathbb{R}^3 \to \mathbb{R}^3$ which rotates a point $x \in \mathbb{R}^3$ by a vector of Euler angles $\theta = (\theta_1, \theta_2, \theta_3) \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$, see [40] for details. Note that singularities which might occur by using Euler angles are not addressed here, instead we refer e.g. to [41]. The function $T = T_{t,\theta} \colon \mathbb{R}^3 \to \mathbb{R}^3$ is then given by

$$
T_{t,\theta}(x) = R_\theta(T_t(x)),
\tag{6}
$$

for each $x \in \mathbb{R}^3$. Thus, the transformed moving image $I^{\text{m}}_{t,\theta} \colon W^{\text{f}} \to \{0, \ldots, 65535\}$ is given by

$$
I^{\text{m}}_{t,\theta}(x) = I^{\text{m}}(T^{-1}_{t,\theta}(x)),
\tag{7}
$$

for each $x \in W^{\text{f}}$. If, for some $x \in W^{\text{f}}$, the value of $T^{-1}_{t,\theta}(x)$ on the right-hand side of Eq. (7) does not belong to the convex hull of W^{m}, we put $I^{\text{m}}_{t,\theta}(x) = 0$. Otherwise, since the mapping I^{m} is defined on the discrete set $W^{\text{m}} \subset \mathbb{Z}^3$,

whereas $T_{t,\theta}^{-1}$ can take arbitrary values in the (continuous) Euclidean space $\mathbb{R}^3$, we employ spline interpolation of order zero such that $T_{t,\theta}^{-1}(x)$ is replaced by the closest voxel of W^m [40].

The objective of image registration is to maximize the similarity between the transformed moving image $I_{t,\theta}^m$ and the fixed image I^f, where similarity is quantified by a so-called similarity measure which we will further specify in Section 4.1.1 below. Therefore, the image registration problem can be reformulated as an optimization problem with some objective function $f : \mathbb{R}^3 \times [0, 2\pi) \times [0, \pi) \times [0, 2\pi) \to [0, 1]$, which maps parameter pairs of shift and rotation, $t \in \mathbb{R}^3$ and $\theta \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$, onto the similarity $f(t, \theta)$ between $I_{t,\theta}^m$ and I^f. The goal is to maximize $f(t, \theta)$. In other words, we seek an optimal shift parameter $t_{\mathrm{opt}} \in \mathbb{R}^3$ and an optimal rotation parameter $\theta_{\mathrm{opt}} \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$ such that

$$(t_{\mathrm{opt}}, \theta_{\mathrm{opt}}) = \underset{t \in \mathbb{R}^3, \theta \in [0,2\pi) \times [0,\pi) \times [0,2\pi)}{\arg \max} f(t, \theta). \tag{8}$$

It is important to note that maximizing $f(t, \theta)$ with respect to both parameters t and θ simultaneously can be challenging in practice due to the complexity of the optimization problem stated in Eq. (8), involving six degrees of freedom (three shift coordinates and three rotations around the three axes). Therefore, we adopt an iterative approach in an inner-outer circle step fashion. In the inner circle step, for a given θ, we determine an optimal value t_θ for the shift parameter t—the one that maximizes $f(\cdot, \theta)$. In the outer circle step, we then determine an optimal value for the rotation parameter θ which maximizes $f(t_\theta, \theta)$.

To determine an optimal shift $t_\theta \in \mathbb{R}^3$ for a given rotation parameter θ, we employ a mask-based image registration procedure which is implemented via FFT, as outlined in [14]. Note that in the context of reassembling fragments, it is essential to apply masking to both (moving and fixed) images. More precisely, masking is crucial because a fragment $F \in \mathcal{F}$ within the 3D image I_{broken} of the broken particle is often surrounded by background or even adjacent partially observable fragments. Thus, image registration without using the mask of a fragment could wrongly be influenced by such regions outside of F. Additionally, registering the fragment within the image I_{intact} of the non-broken particle requires masking of the latter. This ensures that the moving image is registered within the non-broken particle in the fixed image. In particular, by $M^m : W^m \to \{0, 1\}$ and $M^f : W^f \to \{0, 1\}$ we denote the masks of the moving and fixed image, respectively. Then, the mask $M_{t,\theta}^m : W^f \to \{0, 1\}$ of the transformed moving image $I_{t,\theta}^m$ introduced in Eq. (7) is given by

$$M_{t,\theta}^m(x) = M^m(T_{t,\theta}^{-1}(x)), \tag{9}$$

for each $x \in W^f$. Furthermore, we consider the set

$$D_{t,\theta} = \{x \in W^f : M^f(x) = M_{t,\theta}^m(x) = 1\}, \tag{10}$$

for any shift $t \in \mathbb{R}^3$ and orientation angle vector $\theta \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$, where $D_{t,\theta}$ contains those foreground voxels of the fixed image I^f which are foreground voxels of the transformed moving image $I_{t,\theta}^m$.

4.1.1. Similarity measure

To assess the similarity between a transformed moving image $I_{t,\theta}^m$ and the fixed image I^f, restricted to the set $D_{t,\theta}$ given in Eq. (10), a similarity measure is required to indicate how closely the grayscale values of $I_{t,\theta}^m$ match those of I^f. In this context, the masked normalized cross-correlation coefficient (MNCC) is employed. Note that the normalized cross-correlation coefficient (NCC) is a widely used similarity measure in image registration for assessing registration outcomes [40]. However, in the present paper, we consider a modified version of the NCC, which is adapted to masked images [14]. More precisely, we measure the similarity of the moving image I^m, transformed by $T_{t,\theta}$, and the fixed image I^f by means of the quantity $\mathrm{MNCC}_{t,\theta}(I^f, I^m) \in [-1, 1]$, which is given by

$$\mathrm{MNCC}_{t,\theta}(I^f, I^m) = \frac{\sum_{x \in D_{t,\theta}} (I^f(x) - \overline{I_{t,\theta}^f})(I_{t,\theta}^m(x) - \overline{I_{t,\theta}^m})}{\sqrt{\sum_{x \in D_{t,\theta}} (I^f(x) - \overline{I_{t,\theta}^f})^2} \sqrt{\sum_{x \in D_{t,\theta}} (I_{t,\theta}^m(x) - \overline{I_{t,\theta}^m})^2}}, \tag{11}$$

for all $t \in \mathbb{R}^3$ and $\theta \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$, where values of $\mathrm{MNCC}_{t,\theta}(I^\mathrm{f}, I^\mathrm{m})$ close to 1 indicate a large similarity of I^m and I^f. The mean values $\overline{I^\mathrm{f}_{t,\theta}}$ and $\overline{I^\mathrm{m}_{t,\theta}}$ are given by

$$\overline{I^\mathrm{f}_{t,\theta}} = \frac{1}{|D_{t,\theta}|} \sum_{x \in D_{t,\theta}} I^\mathrm{f}(x) \qquad \text{and} \qquad \overline{I^\mathrm{m}_{t,\theta}} = \frac{1}{|D_{t,\theta}|} \sum_{x \in D_{t,\theta}} I^\mathrm{m}_{t,\theta}(x). \tag{12}$$

Note that in the case of $D_{t,\theta} = \varnothing$ we put $\mathrm{MNCC}_{t,\theta}(I^\mathrm{f}, I^\mathrm{m}) = -1$, i.e., to the smallest similarity. The evaluation of the sums in Eq. (11) can be computationally expensive, especially in the optimization process, since they must be repeatedly computed for all considered candidates for t and θ. To address this challenge, we represent all these sums directly in the Fourier domain, following the method outlined by Padfield [14]. This approach involves treating the sums in Eq. (11) as convolutions which corresponds to pointwise multiplication in the Fourier domain, where the latter can be efficiently computed using FFT. This enables an efficient computation of $\mathrm{MNCC}_{t,\theta}(I^\mathrm{f}, I^\mathrm{m})$ for any potential shift $t \in \mathbb{R}^3$ and for any given rotation $\theta \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$.

By using the quantity $\mathrm{MNCC}_{t,\theta}(I^\mathrm{f}, I^\mathrm{m})$ considered in Eq. (11) as similarity measure, the objective function f to be optimized in Eq. (8) is given by

$$f(t, \theta) = \mathrm{MNCC}_{t,\theta}(I^\mathrm{f}, I^\mathrm{m}) \tag{13}$$

for any $t \in \mathbb{R}^3$ and $\theta \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$.

4.1.2. Optimization of shift and rotation

As mentioned above, for any fixed rotation $\theta \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$, the optimal shift $t_{\mathrm{opt},\theta} \in \mathbb{R}^3$ such that

$$t_{\mathrm{opt},\theta} = \arg\max_{t \in \mathbb{R}^3} f(t, \theta) \tag{14}$$

can be computed efficiently using the FFT-based method outlined in [14]. Therefore, the optimization problem considered in Eq. (8) can be written as

$$\theta_{\mathrm{opt}} = \arg\max_{\theta \in [0,2\pi) \times [0,\pi) \times [0,2\pi)} f(t_{\mathrm{opt},\theta}, \theta), \quad \text{and} \quad t_{\mathrm{opt}} = t_{\mathrm{opt},\theta_{\mathrm{opt}}}. \tag{15}$$

Thus, in Eq. (15) we only have to consider an optimization problem with three variables instead of one with six, as it is the case in Eq. (8).

In order to numerically solve the optimization problem formulated in Eq. (15), we deploy PSO [15,16]. This method, which turned out to be successful in biomedical image registration [17], will be adapted for the present application of reassembling fragments. This means in particular that the optimization algorithm operates under minimal assumptions on the objective function being optimized and can search large spaces of candidates. Unlike classical optimization methods, such as gradient descent and quasi-Newton methods, PSO does not rely on the gradient of the objective function. Intuitively speaking, the PSO algorithm considers numerous candidates (called particles in this context) for solving the optimization problem which it updates iteratively. In our application, i.e., for solving the optimization problem formulated in Eq. (15), we put the total number of particles equal to 3.

As output of this procedure we obtain an optimal pair of rotation θ_{opt} and shift t_{opt}, which can then be used to obtain the optimally transformed moving image $\widetilde{I}^\mathrm{m} : W^\mathrm{f} \to \{0, 1, \ldots, 65565\}$ given by

$$\widetilde{I}^\mathrm{m}(x) = I^\mathrm{m}(T^{-1}_{t_{\mathrm{opt}},\theta_{\mathrm{opt}}}(x)), \tag{16}$$

for each $x \in W^\mathrm{f}$. Furthermore, we obtain the optimally transformed moving mask $\widetilde{M}^\mathrm{m} : W^\mathrm{f} \to \{0, 1\}$ given by

$$\widetilde{M}^\mathrm{m}(x) = M^\mathrm{m}(T^{-1}_{t_{\mathrm{opt}},\theta_{\mathrm{opt}}}(x)), \tag{17}$$

for each $x \in W^\mathrm{f}$.

4.2. Virtual reassembling of fragments

This section describes the virtual reassembling algorithm that is applied to a family of fragments $\mathcal{F} = \{F_1, \ldots, F_{n_F}\}$ represented by the images $I_\mathcal{F} = \{I_{F_i}\}_{i=1}^{n_F}$ and masks $M_\mathcal{F} = \{M_{F_i}\}_{i=1}^{n_F}$. Simply speaking, the algorithm works iteratively by reassembling one fragment in each step before continuing with the next fragment. In this context, the reassembling of a fragment $F \in \mathcal{F}$ can be achieved by considering I_F as the moving image with mask M_F and I_{intact} as the fixed image with mask M_{intact}, followed by applying the image registration method described in Section 4.1. In each step of the iterative reassembling algorithm, the reassembling results of the previous steps are taken into account. Therefore, in each step the fixed image I^{f} used for image registration is updated by removing the fragment registered in the current step from the fixed image. In this manner, further fragments tend to be moved/rotated to regions of the fixed image in which no fragments have been placed in previous steps. By incorporating this modification into the algorithm, the outcome of the reassembling algorithm depends on the quality of the reassembled fragment at each step. Consequently, we introduce a similarity measure in Section 4.2.1, which is then used in each step of the algorithm to assess the quality of a reassembled fragment. Furthermore, it has turned out that the order of fragments being reassembled influences the reassembling result. Thus, we provide a strategy on how to reassemble fragments. Overall, the reassembling algorithm is designed as sketched below.

Algorithm 1 Virtual reassembling algorithm

---

function VIRTUALREASSEMBLEFRAGMENTS(Set of fragments $\mathcal{F}$, images of fragments $I_\mathcal{F}$, masks of fragments $M_\mathcal{F}$, image of non-broken particle I_{intact}, mask of non-broken particle M_{intact})

  $\{\mathcal{F}_i\}_{i=1}^n \leftarrow$ FragmentSetPartition($\mathcal{F}$) as described in Section 4.2.3

  $I^{\text{f}} \leftarrow I_{\text{intact}}$ and $M^{\text{f}} \leftarrow M_{\text{intact}}$

  Set $I_{\text{R}} : W_{\text{intact}} \rightarrow \{0\}$

  for all $i = 1, 2, \ldots, n$ **do**

    $\text{MSSM}_{\text{best}} \leftarrow -1$

    while $\mathcal{F}_i \neq \varnothing$ **do**

      for all fragments $F \in \mathcal{F}_i$ **do**

        $I^{\text{m}} \leftarrow I_F$ and $M^{\text{m}} \leftarrow M_F$

        $\widetilde{I}^{\text{m}}$ and $\widetilde{M}^{\text{m}} \leftarrow$ ImageRegistration($I^{\text{f}}, M^{\text{f}}, I^{\text{m}}, M^{\text{m}}$) as described in Section 4.1

        $\widetilde{I}^{\text{f}} \leftarrow$ ModifyFixedImage($I^{\text{f}}, M^{\text{f}}, \widetilde{I}^{\text{m}}, \widetilde{M}^{\text{m}}$) by means of Eq. (18)

        $\text{MSSM}_F \leftarrow$ ComputeMSSM($I^{\text{f}}, \widetilde{I}^{\text{f}}, \widetilde{M}^{\text{m}}$) by means of Eq. (19)

        if $\text{MSSM}_F \geq \text{MSSM}_{\text{best}}$ **then**

          $\widetilde{F} \leftarrow F$, $M_{\widetilde{F}} \leftarrow \widetilde{M}^{\text{m}}$ and $\text{MSSM}_{\text{best}} \leftarrow \text{MSSM}_F$

      $\mathcal{F}_i \leftarrow \mathcal{F}_i \setminus \{\widetilde{F}\}$

      I^{f} and $M^{\text{f}} \leftarrow$ UpdateFixedImage($I^{\text{f}}, M^{\text{f}}, M_{\widetilde{F}}$) by means of Eq. (20) and (21)

      $I_{\text{R}} \leftarrow$ UpdateReassemblingResult($I_{\text{R}}, M_{\widetilde{F}}$) by means of Eq. (22)

  return I_{R}

We now describe the similarity measure for evaluating how well we reassembled a fragment, see Section 4.2.1. Afterwards, we explain the process of modifying the fixed image, which is used for registration after each step of the reassembling algorithm, see Section 4.2.2. Then, Section 4.2.3 outlines the virtual reassembling order, specifying the particular set of fragments from which the fragment to be reassembled is chosen. Finally, in Section 4.2.4, we describe the kind of results obtained by the reassembling algorithm.

4.2.1. Evaluation of reassembling quality

Typically, there are multiple fragments which have to be reassembled. Thus, we need a strategy for choosing the fragment which, after reassembly, leads to the best result. It turned out that the MNCC given in Eq. (11) is an appropriate similarity measure in mask-based image registration of single fragments. However, for evaluating the reassembling results of multiple fragments we use the mean structural similarity measure (MSSM) as described in [42]. By utilizing MSSM, we can measure the structural similarity of two images by comparing the local patterns of their voxel values which have been normalized for luminance and contrast. This similarity measure is advantageous for evaluating the reassemble results of different fragments and it is expected to have a larger value if a fragment is better reassembled than another fragment. In the following, we state the definition of MSSM and show how this measure is applied to evaluate the goodness of fit of a reassembled fragment.

Recall that for each fragment $F \in \mathcal{F}$ with image I_F and mask M_F, we registered the moving image $I^{\mathrm{m}} = I_F$ within the fixed image $I^{\mathrm{f}} = I_{\mathrm{intact}}$, using the mask-based registration procedure described in Section 4.1, where the registration leads an (optimally) transformed moving image $\widetilde{I}^{\mathrm{m}}$ with mask $\widetilde{M}^{\mathrm{m}}$, see Eqs. (16) and (17). Now, in order to evaluate the goodness of fit of the masked-based registration result for F, we utilize the image $\widetilde{I}^{\mathrm{f}} : W^{\mathrm{f}} \to \{0, 1, \ldots, 65565\}$, which is given by

$$\widetilde{I}^{\mathrm{f}}(x) = \begin{cases} \widetilde{I}^{\mathrm{m}}(x), & \text{if } \widetilde{M}^{\mathrm{m}}(x) = 1, \\ I^{\mathrm{f}}(x), & \text{otherwise,} \end{cases} \tag{18}$$

for each $x \in W^{\mathrm{f}}$. Note that $\widetilde{I}^{\mathrm{f}}$ coincides with the mask-based registration result $\widetilde{I}^{\mathrm{m}}$ within its mask $\widetilde{M}^{\mathrm{m}}$, where $\widetilde{I}^{\mathrm{f}}$ is compared to I^{f} using MSSM. More precisely, we evaluate the structural similarity of the images I^{f} and $\widetilde{I}^{\mathrm{f}}$, restricted to the set $\widetilde{D} = \{x \in W^{\mathrm{f}} : \widetilde{M}^{\mathrm{m}}(x) = 1\}$, by means of

$$\mathrm{MSSM}(I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}, \widetilde{M}^{\mathrm{m}}) = \frac{1}{|\widetilde{D}|} \sum_{x \in \widetilde{D}} \mathrm{SSM}_{I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}}(x), \tag{19}$$

where the formal definition of SSM is given in the Appendix. Here, we just remark that $\mathrm{SSM}_{I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}}(x)$ is a structural similarity measure of I^{f} and $\widetilde{I}^{\mathrm{f}}$ evaluated at $x \in \widetilde{D}$. Moreover, the definition of SSM immediately implies that $\mathrm{MSSM}(I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}, \widetilde{M}^{\mathrm{m}}) \in [0, 1]$, where values of $\mathrm{MSSM}(I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}, \widetilde{M}^{\mathrm{m}})$ close to 1 indicate a high degree of structural similarity. In particular, in the context of reassembling, a value of $\mathrm{MSSM}(I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}, \widetilde{M}^{\mathrm{m}})$ close to 1 indicates that the fragment F is well reassembled.

4.2.2. Updating the fixed image

In this section, we explain how the fixed image I^{f}, used in the mask-based registration procedure, is updated after identifying the best reassembled fragment $F \in \mathcal{F}$, i.e., that fragment $F \in \mathcal{F}$ which leads to the largest value of MSSM in comparison to all other fragments. Note that without updating I^{f}, we would not take the reassembling result of F into account in subsequent steps of the reassembling algorithm, which could lead to overlapping of reassembled fragments.

More precisely, we update I^{f} as follows: After identifying the best reassembled fragment $F \in \mathcal{F}$, with (optimally) transformed image $\widetilde{I}_F$ and mask $\widetilde{M}_F$ determined by means of Eqs. (16) and (17), respectively, we replace I^{f} by the image $\widetilde{I}^{\mathrm{f}} : W^{\mathrm{f}} \to \{0, 1, \ldots, 65565\}$, which is given by

$$\widetilde{I}^{\mathrm{f}}(x) = \begin{cases} 0, & \text{if } \widetilde{M}_F(x) = 1, \\ I^{\mathrm{f}}(x), & \text{otherwise,} \end{cases} \tag{20}$$

for each $x \in W^{\mathrm{f}}$. Analogously, we replace M^{f} by the image $\widetilde{M}^{\mathrm{f}} : W^{\mathrm{f}} \to \{0, 1\}$, where

$$\widetilde{M}^{\mathrm{f}}(x) = \begin{cases} 0, & \text{if } \widetilde{M}_F(x) = 1, \\ M^{\mathrm{f}}(x), & \text{otherwise,} \end{cases} \tag{21}$$

for each $x \in W^{\mathrm{f}}$. By incorporating $\widetilde{I}^{\mathrm{f}}$ as the new fixed image during registration of a fragment image in the next step of the algorithm, we take into consideration the outcomes of previous steps. This approach effectively constrains the range of potential translations and rotations for reassembling fragments in the following steps of the reassembling algorithm.

4.2.3. Reassembling order

We observed that choosing a random order in which we register fragment images does not lead to satisfactory reassembling results. A reason for this is that fragments can vary in terms of their size (fragment volume), shape and original position within the non-broken particle. The reassembling process for fragments entirely contained within the non-broken particle relies solely on the texture exhibited in the interior of the non-broken particle. In contrast, the reassembling process for fragments located at the boundary of the non-broken particle is influenced by both texture and the shape of the boundary, i.e., further information is available which in turn makes registration easier. Moreover, an additional challenge arises when reassembling small fragments. These smaller fragments often exhibit a more

homogeneous texture, leading to a wide range of appropriate translations and rotations within the registration step of the algorithm which results in large values of MSSM.

Therefore, we introduce a heuristics with which we prioritize the registration of "promising" fragments. Then, due to the subsequent updates of the fixed image, the set of potential translations and rotations decreases in size, see Section 4.2.2. In other words, this leads to less ambiguity when registering more difficult, low priority cases like images of small fragments. More precisely, we partition the set $\mathcal{F}$ of all fragments into $n \geq 1$ pairwise disjoint subsets $\mathcal{F}_1, \ldots, \mathcal{F}_n \subset \mathcal{F}$ with $\mathcal{F} = \bigcup_{i=1}^{n} \mathcal{F}_i$, where each subset $\mathcal{F}_i$ contains $n^{(i)} \geq 1$ fragments for $i \in \{1, 2, \ldots, n\}$. The specification of this partitioning influences the quality of reassembling results, where partitioning of the set $\mathcal{F}$ of fragments based on their sizes turned out to be a reasonable choice, as larger fragments are more likely to share their boundary with the boundary of the non-broken particle. Therefore, it is reasonable to reassemble larger fragments first, i.e., the subsets $\mathcal{F}_1, \ldots, \mathcal{F}_n$ of $\mathcal{F}$ are defined such that $\mathcal{F}_1$ contains the largest $n^{(1)}$ fragments in $\mathcal{F}$, the subset $\mathcal{F}_2$ contains the largest $n^{(2)}$ fragments in $\mathcal{F} \setminus \mathcal{F}_1$, and so on. This procedure is repeated until all fragments of $\mathcal{F}$ are assigned to some set $\mathcal{F}_1, \ldots, \mathcal{F}_n$. The number n of subsets and their sizes $n^{(1)}, \ldots, n^{(n)}$ depend on the number and sizes of fragments considered in the given virtual reassembling task. Note that in the case of the experimental data from both the ore and slag particles described in Section 2, where the number of fragments is small, no partitioning of $\mathcal{F}$ is required. However, in the simulation study considered in Section 5 we have a larger number of fragments and, thus, we assign three fragments to each set $\mathcal{F}_1, \ldots, \mathcal{F}_{n-1}$, and the remaining fragments to $\mathcal{F}_n$.

In other words, the reassembling algorithm works as follows: First, all fragments of $\mathcal{F}_1$ are reassembled and the MSSM is computed for each $F \in \mathcal{F}_1$ as described in Section 4.2.1. Second, the fragment $\widetilde{F} \in \mathcal{F}_1$ with the largest MSSM value is identified. The fixed image is then updated by the reassembling result of $\widetilde{F}$, as described in Section 4.2.2, Then, this process is repeated on the set $\mathcal{F}_1 \setminus \{\widetilde{F}\}$ with the updated fixed image. Once all fragments in $\mathcal{F}_1$ are reassembled, the algorithm proceeds to the next set $\mathcal{F}_2$ and continues this process until all fragments are reassembled.

4.2.4. Final output of the reassembling algorithm

The reassembling algorithm creates an image $I_R : W_{\text{intact}} \to \{0, 1, \ldots, n_F\}$, which assigns each voxel of W_{intact} to a reassembled fragment $F_i \in \mathcal{F}$ for some $i \in \{1, 2, \ldots, n_F\}$, or to the background, where I_R is constructed iteratively and initialized as $I_R \equiv 0$. After each reassembling step, when we identified the best reassembled fragment $F_i \in \mathcal{F}$ for some $i \in \{1, 2, \ldots, n_F\}$ with (optimally) transformed image I_{F_i} and mask M_{F_i} determined by means of Eqs. (16) and (17), respectively, we replace I_R by the image $\widetilde{I}_R : W_{\text{intact}} \to \{0, 1, \ldots, n_F\}$, which is given by

$$\widetilde{I}_R(x) = \begin{cases} i, & \text{if } M_{F_i}(x) = 1, \\ I_R(x), & \text{otherwise,} \end{cases} \tag{22}$$

for each $x \in W_{\text{intact}}$. In Section 6, the finally obtained image I_R is used to determine the fracture surfaces.

5. Simulation study

In order to evaluate the reassembling algorithm described in Section 4, we applied the algorithm to various sets of virtually generated fragments. Below we explain the simulation study we conducted for this purpose. In Section 5.1, we describe the virtual generation of fragments and, subsequently, in Section 5.2 we introduce several evaluation measures to assess the goodness of fit of the reassembling algorithm, considering both the overall reassembling outcome and, in particular, the fit of the fracture surfaces.

5.1. Virtual generation of fragments

In the following, we provide a brief overview of the procedure involved in generating virtual fragments by employing (distance-based) Voronoi tessellations [22,43]. Such tessellations can be used to partition the sampling window W_{intact} of the non-broken particle, with image I_{intact} and mask M_{intact}, into a collection of $m \geq 1$ non-overlapping sets $C_1, C_2, \ldots, C_m \subset W_{\text{intact}}$, so-called cells. This model is fully specified by a point pattern $\mathcal{P} = \{s_i\}_{i=1}^{m} \subset \mathbb{R}^3$, where $s_i \in \mathbb{R}^3$ is the seed point of cell C_i for each $i \in \{1, 2, \ldots, m\}$. More precisely, the i-th cell C_i of the Voronoi tessellation $\mathcal{T} = \{C_i\}_{i=1}^{m}$ is given by

$$C_i = \{x \in W_{\text{intact}} : \|x - s_i\|^2 \leq \|x - s_j\|^2 \text{ for each } j = 1, \ldots, m, j \neq i\}, \tag{23}$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^3$.

For the purpose of generating virtual fragments, suitable point patterns $\mathcal{P} = \{s_i\}_{i=1}^m$ have to be chosen. In this simulation study, we uniformly sample points in the convex hull of W_{intact} for generating point patterns $\mathcal{P}$, with an adjustable number $m \geq 1$ of points. By varying m, we can generate different tessellations with varying numbers of cells, thus resulting in different numbers of generated fragments, where a larger number of points leads to a larger number of fragments, while a smaller m produces fewer fragments. For details on simulating random point patterns (also referred to as point processes), see e.g. [43].

Recall that we denoted the set of voxels associated with the non-broken particle by $W'_{\text{intact}} \subset W_{\text{intact}}$, see Section 3.1. In order to generate virtual fragments within the non-broken particle we need to find all cells of $\mathcal{T}$, which overlap with W'_{intact}, i.e., to identify those $C \in \mathcal{T}$ for which $C \cap W'_{\text{intact}} \neq \varnothing$ holds. In this way, we obtain a tessellation $\mathcal{T}' = \{C'_i\}_{i=1}^{m'}$ of W'_{intact}, consisting of $m' \geq 1$ non overlapping and non empty subsets $C'_1, \ldots, C'_{m'}$ of W'_{intact} such that $\bigcup_{i=1}^{m'} C'_i = W'_{\text{intact}}$. Then, for each $i \in \{1, 2, \ldots, m'\}$, the i-th virtually generated fragment corresponds to C'_i. Furthermore, for each fragment $C' \in \mathcal{T}'$ we get the fragment image $I_{C'} : W_{C'} \to \{0, 1, \ldots, 65565\}$ by cropping, i.e.,

$$I_{C'}(x) = \begin{cases} I_{\text{intact}}(x), & \text{if } x \in C', \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

for each $x \in W_{C'}$, where $W_{C'} \subset \mathbb{Z}^3$ denotes the smallest cuboidal sampling window which contains the fragment C'. Thus, the mapping $I_{C'}$, assigning each voxel of $x \in C'$ the corresponding grayscale value of I_{intact}, can be considered to be a cutout of I_{intact} representing the fragment C'. Besides this, we consider the mask $M_{C'} : W_{C'} \to \{0, 1\}$ of C', where $M_{C'}(x) = 1$ for each $x \in C'$, and $M_{C'}(x) = 0$ otherwise.

Additionally, we consider the image $S' : W_{\text{intact}} \to \{0, 1, \ldots, m'\}$, which assigns each voxel to a generated fragment or background, i.e.,

$$S'(x) = \begin{cases} i, & \text{if } x \in C'_i \text{ for some } i \in \{1, 2, \ldots, m'\}, \\ 0, & \text{otherwise,} \end{cases} \tag{25}$$

for each $x \in W_{\text{intact}}$. This image serves as the ground truth for evaluating the reassembling algorithm. The workflow of generating fragments by means of tessellations is sketched in Figure 2.

To account for the possible rotations of fragments during fracture, we select a rotation angle $\theta \in [-\frac{\pi}{6}, \frac{\pi}{6}]^3$ at random for each fragment $C' \in \mathcal{T}'$. Subsequently, the images $I_{C'}$ and $M_{C'}$ are rotated by means of the rotation function $R_\theta : \mathbb{R}^3 \to \mathbb{R}^3$, as described in Section 4.1. The choice of a random rotation within the range of $[-\frac{1}{6}\pi, \frac{1}{6}\pi]^3$ was intentional, as component-wise larger rotation angles led to greatly extended computation times for achieving satisfactory reassembling results. Furthermore, for the experimentally acquired datasets of ore and slag, described in Section 2, only minor rotations of fragments are typically encountered.

5.2. Evaluation of reassembling algorithm for virtually generated fragments

To evaluate the reassembling algorithm introduced in Section 4, we apply this algorithm to the virtually generated fragments as described in Section 5.1. Therefore, we consider two performance measures to evaluated the goodness of fit of the resulting reassembled particles, by comparing the image I_R given in Eq. (22) with the ground truth image S' given in Eq. (25). Namely, the fraction $\lambda \in [0, 1]$ of correctly assigned voxels, which is given by

$$\lambda = \frac{|\{x \in W_{\text{intact}} : S'(x) = I_R(x)\}|}{|\{x \in W_{\text{intact}} : S'(x) > 0\}|}, \tag{26}$$

and the fraction $\lambda^{\text{fracture}} \in [0, 1]$ of correctly assigned voxels located on the fracture surface, defined as

$$\lambda^{\text{fracture}} = \frac{|\{x \in W_{\text{intact}} : I^{\text{fracture}}(x) = I_R^{\text{fracture}}(x)\}|}{|\{x \in W_{\text{intact}} : I^{\text{fracture}}(x) = 1\}|}, \tag{27}$$

where the images $I^{\text{fracture}} : W_{\text{intact}} \to \{0, 1\}$ and $I_R^{\text{fracture}} : W_{\text{intact}} \to \{0, 1\}$ describe the fracture surfaces observed in S' and I_R, respectively. More details on the definition and computation of I^{fracture} and I_R^{fracture} are given in Section 6.1.

To evaluate the reassembling algorithm, we generated 20 different sets of fragments as described in Section 5.1, where the number of fragments in these sets varies between 2 and 13. The performance measures provided in Eqs. (26)

and (27) were computed for each of the 20 sets of fragments, where we obtain the averaged values $\lambda = 0.96$ and $\lambda^{\text{fracture}} = 0.69$. Note that the mean value of $\lambda^{\text{fracture}}$ is smaller than that of λ. This discrepancy results from the fact that even a slight misalignment or rotation of a reassembled fragment compared to its original position or orientation leads to a relatively large mismatch between the fracture surfaces observed in I^{fracture} and I_R^{fracture}, which leads to smaller values of $\lambda^{\text{fracture}}$. Nevertheless, the original positions of the fractures are well captured by the reassembling algorithm, as the mean value of λ is close to 1. These results indicate that the reassembly algorithm is capable of accurately reassembling the simulated fragments.

6. Quantitative analysis of fracture mechanisms

As already mentioned above, a deeper understanding of whether fractures during fragmentation of particles occur along grain boundaries (intergranular fractures), randomly (transgranular fractures) or as a superposition of both fracture mechanisms is important for the liberation of minerals. To characterize fracture mechanisms locally, we need to analyze the fracture surfaces between adjacent fragments. Therefore, in Section 6.1 we explain how fracture surfaces can be determined voxelwise from the output of the virtual reassembling algorithm presented in Section 4. Then, in Section 6.2, two different descriptors of fracture surfaces are considered, which characterize global and local features of fracture mechanisms. Finally, in Section 6.3, these tools are applied in order to analyze the fracture mechanisms of two (ore and slag) particles, which consist of three mineral phases as described in Section 3.2.

6.1. Voxelwise determination of fracture surfaces

Recall that the output of the reassembling algorithm considered in this paper is an image $I_R : W_{\text{intact}} \rightarrow \{0, 1, \ldots, n_F\}$ of the virtually reassembled particle, where each fragment is associated with a label $i \in \{1, \ldots, n_F\}$, as described in Section 4.2.4. In this context, it is important to note that gaps between labeled fragments may occur in I_R. The number and sizes of these gaps depend on the quality of the reassembling and the discretization effects that arise when rotating discrete images during reassembling. These gaps prevent the determination of fracture surfaces directly from the reassembled fragments and need to be filled first. Therefore, we define the image $I^{\text{gaps}} : W_{\text{intact}} \rightarrow \{0, 1\}$, which assigns each voxel of W_{intact} either to 1 or 0, provided that the voxel belongs to a gap or not. i.e.,

$$I^{\text{gaps}}(x) = \begin{cases} 1, & \text{if } I_{\text{intact}}(x) > 0 \text{ and } I_R(x) = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{28}$$

for each $x \in W_{\text{intact}}$. To extract the fracture surfaces in I_R, we first utilize a region-growing approach based on the Euclidean distance transform [26] in order to fill the gaps resulting from slight errors of the reassembling algorithm, see Figure 4. More precisely, for each pair $i, j \in \{1, 2, \ldots, n_F\}$ with $i \neq j$, we compute the Euclidean distances of voxels in the region with label i to voxels in the region with label j. Each voxel $x \in W_{\text{intact}}$ with $I^{\text{gaps}}(x) = 1$ is assigned to the labeled region to which it has the smallest Euclidean distance. If x has the same (smallest) Euclidean distance to more than one labeled regions, it is assigned to one of these labeled regions at random.

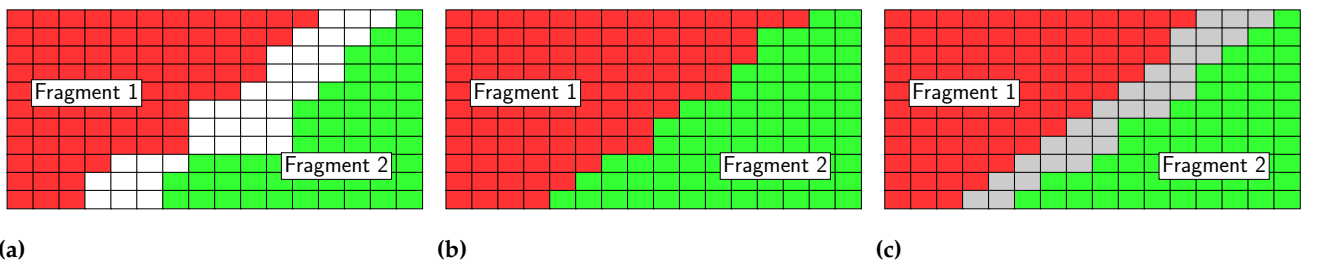**(a)**   **(b)**   **(c)**

Figure 4. Determination of fracture surfaces. (a) Gaps between two differently labeled regions corresponding to distinct fragments (colored in red and green) are identified by white voxels, and (b) filled using a region-growing approach. (c) The fracture surfaces are then determined by identifying all boundary voxels (colored in gray) of adjacent (differently labeled) fragments.

After filling the gaps resulting from reassembling, all boundary voxels between adjacent fragments are identified. Hereby, we consider any voxel a boundary voxel, which is not completely surrounded by voxels corresponding to the same fragment with respect to the 6-neighborhood, where the 6-neighborhood of a voxel $x \in W_{\text{intact}}$ is given by the set $M_x = \{y \in W_{\text{intact}} : \|x - y\| \leq \sqrt{2}\}$. The boundary voxels defined in this way correspond to the fracture surfaces of adjacent reassembled fragments. Thus, the fracture surfaces observed in I_R are given by the binary image

$I_R^{\text{fracture}} : W_{\text{intact}} \to \{0, 1\}$, which assigns each voxel of W_{intact} to the boundary between adjacent fragments, or the background, i.e.,

$$I_R^{\text{fracture}}(x) = \begin{cases} 1, & \text{if } x \text{ belongs to the boundary between adjacent reassembled fragments,} \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

for each $x \in W_{\text{intact}}$.

6.2. Descriptors of fracture surfaces

We explore two different descriptors that characterize fracture surfaces and thus enable us to analyze whether fractures occur typically intergranular, transgranular or as a superposition of both fracture types. First, we investigate the global behavior of fracture mechanisms, considering the set F^{fracture} of voxels which are located at fracture surfaces, i.e.,

$$F_R^{\text{fracture}} = \{x \in W_{\text{intact}} : I_R^{\text{fracture}}(x) = 1\}, \tag{30}$$

where I_R^{fracture} is given in Eq. (29). In particular, we investigate the global fracture behavior of each material phase $i \in \{1, \ldots, n_{\text{phases}}\}$ observed in images of non-broken particles as described in Section 3.2. More precisely, for each $i \in \{1, \ldots, n_{\text{phases}}\}$, the fraction $\rho_i^{\text{fracture}} \in [0, 1]$ of voxels at fracture surfaces of the i-th phase is considered, where

$$\rho_i^{\text{fracture}} = \frac{|\{x \in F_R^{\text{fracture}} : I_{\text{phases}}(x) = i\}|}{|F_R^{\text{fracture}}|}. \tag{31}$$

Furthermore, we investigate the local (voxelwise) behavior of fracture mechanisms, considering the local entropy $E(x) \in [0, \infty)$ for each $x \in F_R^{\text{fracture}}$, which is defined as follows. For each $x \in F_R^{\text{fracture}}$ and for each $i \in \{1, \ldots, n_{\text{phases}}\}$, we consider the relative frequency $p_x^{(i)} \in [0, 1]$ of neighbors of x belonging to the i-th material phase, where

$$p_x^{(i)} = \frac{1}{6} |\{y \in M_x : I_{\text{phases}}(y) = i\}|. \tag{32}$$

Then, the local entropy $E(x)$ of $x \in F_R^{\text{fracture}}$ is given by

$$E(x) = \sum_{i=1}^{n_{\text{phases}}} p_x^{(i)} \log_2(p_x^{(i)}), \tag{33}$$

where $\log_2$ denotes the logarithm to the basis 2 (i.e., $s = 2^{\log_2(s)}$ for all $s > 0$).

Note that $E(x)$ can be interpreted as degree of disorder of the mineral phases in the local neighborhood of $x \in F_R^{\text{fracture}}$. In other words, a fracture voxel has a large value $E(x)$ if many neighboring voxels are assigned to different mineral phases. Vice versa, if $E(x) = 0$, then all neighboring voxels of $x \in F_R^{\text{fracture}}$ are assigned to the same mineral phase. Therefore, a voxel $x \in F_R^{\text{fracture}}$ with $E(x) = 0$ indicates an intergranular fracture, whereas transgranular fractures are typically indicated by local entropy values larger than zero. In particular, for each $i \in \{1, \ldots, n_{\text{phases}}\}$, the transgranular fraction $\lambda_i^{\text{trans}} \in [0, 1]$ of the i-th mineral phase is given by

$$\lambda_i^{\text{trans}} = \frac{|\{x \in F_R^{\text{fracture}} : E(x) > 0, I_{\text{phases}}(x) = i\}|}{|\{x \in F_R^{\text{fracture}} : I_{\text{phases}}(x) = i\}|}, \tag{34}$$

whereas the intergranular fraction $\lambda_i^{\text{inter}} \in [0, 1]$ of the i-th phase is given by

$$\lambda_i^{\text{inter}} = 1 - \lambda_i^{\text{trans}}. \tag{35}$$

6.3. Numerical results

In Table 3, numerical values of the fraction ρ_i^{fracture} of voxels at fracture surfaces of the i-th phase, as well as of its transgranular and intergranular fractions λ_i^{trans} and λ_i^{trans}, are given which we obtained for the three phases of the ore and slag particles described in Section 3.2.

Comparing the values of ρ_i^{fracture} with those of the corresponding global volume fraction ρ_i, i.e., the fraction of all voxels associated with the i-th phase, we can observe slight discrepancies between these values, see Table 2. In particular, for the ore particle, the fraction ρ_1^{fracture} of voxels at fracture surfaces of phase P_1 is smaller than the volume fraction ρ_1 of the entire particle, whereas the opposite behavior is observed for the other two phases P_2 and P_3 of the ore particle. Similar observations are made for the slag particle, which indicates that for both particles, fractures occur more densely in phases P_2 and P_3, compared to phase P_1.

Given that in the ore particle, P_1 constitutes the phase with the largest volume fraction ($\rho_1 = 0.96$), fractures predominantly occur within this phase, i.e., fractures in phase P_1 of the ore particle occur predominantly transgranular. In contrast, regarding the other two phases, there is no clear indication of fractures occurring either intergranular or transgranular, i.e., it seems that both P_2 and P_3 exhibit a mixture of both fracture mechanisms.

With respect to latter aspect, the fracture behavior of the slag particle is different. Furthermore, note that there is a more pronounced balancing between the volume fractions of the three mineral phases of the slag particle, i.e., there is not only one predominant phase, see Table 2. Besides this, the slag particle contains air bubbles in phase P_3 and with respect to the definition of the transgranular fraction, fracture surfaces always occur transgranular in this phase. However, for phases P_1 and P_2 of the slag particle, there seem to be no dominant transgranular nor integranular fracture mechanisms, as the transgranular and integranular fractions λ_2^{trans}, λ_2^{inter} as well as λ_3^{trans}, λ_3^{inter} are significantly larger than zero, see Table 3.

Table 3. Numerical values obtained for the descriptors of fracture surfaces of ore and slag particles ($i = 1, 2, 3$).

ore	phase P_1	phase P_2	phase P_3
ρ_i^{fracture}	0.91	0.08	0.002
λ_i^{trans}	0.05	0.36	0.57
λ_i^{inter}	0.95	0.64	0.43
slag	phase P_1	phase P_2	phase P_3
ρ_i^{fracture}	0.70	0.24	0.06
λ_i^{trans}	0.37	0.68	1.00
λ_i^{inter}	0.63	0.32	0.00

7. Conclusion

We introduced a novel method for characterizing fracture mechanisms in multi-component materials by means of 3D CT image data. In mineral liberation, the understanding of these mechanisms is crucial, particularly whether fractures occur along the boundaries of mineral phases (intergranular fracture) and/or within mineral phases (transgranular fracture). Conventional techniques for analyzing fracture mechanisms are focused on globally comparing the surface exposure of mineral phases extracted from image measurements before and after fracture. Instead, we presented a virtual reassembling algorithm based on image registration techniques, which is applied to 3D data of two different particles before and after fracture in order to determine and characterize the individual fracture surfaces. This enabled us to conduct a local quantitative analysis of fracture mechanisms by voxelwise comparing adjacent regions at fracture surfaces. The algorithm has been validated through a simulation study and subsequently applied to image data of two particles: a natural occurring ore and a slag from a pyrometallurgical process.

Since the characterization of the fracture mechanisms cannot be performed directly on image data obtained from CT measurements, some image preprocessing was required, where we binarized the CT image data of the ore and slag particles before and after fracture, followed by a phase-based segmentation step to extract mineral phases from the non-broken particles. However, since different mineral phases may have similar X-ray attenuation coefficients leading to similar grayscale values [44], it can happen that the segmentation of mineral phases, merely based on CT image data, is difficult or even impossible. Therefore, in a forthcoming paper, we will apply our method for characterizing fracture mechanisms to an extended set of image data, combining 3D CT measurements of particles with 2D SEM-EDS data acquired by means of the mineral liberation analyzer (MLA) [45,46].

References

1. Gaudin, A. *Principles of Mineral Dressing*. McGraw-Hill, **1939**.
2. Hsih, C.S.; Wen, S.B. An extension of Gaudin's liberation model for quantitatively representing the effect of detachment in liberation. *International Journal of Mineral Processing* **1994**, *42*, 15-35.
3. Singh, V.; Venugopal, R.; Banerjee, P.K.; Saxena, V.K. Effect of morphology on breakage and liberation characteristics of minerals and coal. *Mining, Metallurgy & Exploration* **2014**, *31*, 186-192.
4. King, R.P.; Schneider, C.L. Mineral liberation and the batch communion equation. *Minerals Engineering* **1998**, *11*, 1143-1160.
5. Mariano, R.A.; Evans, C.L.; Manlapig, E. Definition of random and non-random breakage in mineral liberation-A review. *Minerals Engineering* **2016**, *94*, 51-60.
6. Little, L.; Mainza, A.N.; Becker, M.; Wiese, J.G. Using mineralogical and particle shape analysis to investigate enhanced mineral liberation through phase boundary fracture. *Powder Technology* **2016**, *301*, 794–804.
7. Leißner, T.; Hoang, D.H.; Rudolph, M.; Heining, T.; Bachmann, K.; Gutzmer, J.; Schubert, H.; Peuker, U.A. A mineral liberation study of grain boundary fracture based on measurements of the surface exposure after milling. *International Journal of Mineral Processing* **2016**, *156*, 3–13.
8. Mirzaei, Z.S.; Khalesi, M.R. Development of a simulator for random and non-random breakage of particles and liberation of grains based on Voronoi tessellation. *Minerals* **2019**, *9*, 341.
9. Xu, W.; Dhawan, N.; Lin, C.L.; Miller, J.D. Further study of grain boundary fracture in the breakage of single multiphase particles using x-ray microtomography procedures. *Minerals Engineering* **2013**, *46*, 89–94.
10. Brown, L.G. A survey of image registration techniques. *Association for Computing Machinery* **1992**, *24*, 179–186.
11. Maintz, J.B.A.; Viergever, M.A. A survey of medical image registration. *Medical Image Analysis* **1998**, *2*, 1-36.
12. Hill, D.L.G.; Batchelor, P.G; Holden, M.; Hawkes, D.J. Medical image registration. *Physics in Medicine & Biology* **2001**, *46*, 1-45.
13. Barnea, D.I.; Silverman, H.F. A class of algorithms for fast digital image registration. *IEEE Transactions on Computers* **1972**, *100*, 1-36.
14. Padfield, D. Masked object registration in the Fourier domain. *IEEE Transactions on Image Processing* **2012**, *21*, 2706–2718.
15. Kennedy, J.; Eberhart, R. *Particle swarm optimization.Proceedings of IEEE International Conference on Neural Networks* **1995**, 1942–1948.
16. Kennedy, J. *Swarm Intelligence*. In *Handbook of Nature-inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*, ed. by Zomaya, A.Y., Springer **2006**, 187–219.
17. Wachowiak, M.P.; Smolikova, R.; Zheng, Y.; Zurada, J.M.; Elmaghraby, A.S. An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Transactions on Evolutionary Computation* **2004**, *8*, 289-301.
18. Bracewell, R. *Pentagram Notation for Cross Correlation*. In *The Fourier Transform and Its Applications*, ed. Bracewell, R., McGraw-Hill **1965**, 46–243.
19. Lesse, J.; Novak, C.; Taylor, V. An automated technique for obtaining cloud motion from geosynchronous satellite data using crosscorrelation. *Journal of Applied Meteorology and Climatology* **1971**, *10*, 118–132.
20. Lewis, P. Fast template matching. *Vision Interface* **1995**, *95*, 15-19.
21. Anuta, P.E. Spatial registration of multispectral and multitemporal digital imagery using fast Fourier transform techniques. *IEEE Transactions on Geoscience Electronics* **1970**, *8*, 353-368.
22. Okabe, A.; Boots, B.; Sugihara, K.; Chiu, S. *Spatial Tessellations Concepts and Applications of Voronoi Diagrams*. J.Wiley & Sons **2000**.
23. Šedivý, O.; Brereton, T; Westhoff, D.; Polívka, L.; Beneš, V.; Schmidt, V.; Jäger, A. 3D reconstruction of grains in polycrystalline materials using a tessellation model with curved grain boundaries. *Philosophical Magazine* **2016**, *96*, 1926-1949.

24. Petrich, L.; Furat, O.; Wang, M.; Krill III, C.E.; Schmidt, V. Efficient fitting of 3D tessellations to curved polycrystalline grain boundaries. *Frontiers in Materials* **2021**, *8*, 760602.

25. European Commission and Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs; Grohol, M .; Veeh, C. Study on the critical raw materials for the EU 2023 – Final report. *Publications Office of the European Union* **2023**, https://data.europa.eu/doi/10.2873/725585.

26. Pratt, W. *Digital Image Processing: PIKS Scientific Inside*. J. Wiley & Sons **2007**.

27. Ridler, **Thomas Wilhelm**; Calvard, S. Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man and Cybernetics* **1978**, *8*, 630-632.

28. Sezgin, M.; Sankur, B. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* **2004**, *13*, 146-165.

29. Dhanachandra, N.; Manglem, K.; Chanu, Y.J. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science* **2015**, *54*, 764–771.

30. Gan, G.; Ma, C.; Wu, J. *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics **2020**.

31. Maire, E.; Withers, P.J. Quantitative X-ray tomography. *International Materials Reviews* **2014**, *59*, 1-43.

32. Soille, P. *Morphological Image Analysis: Principles and Applications*. Springer **2013**.

33. Beucher, S.; Meyer, F. The morphological approach to segmentation: the watershed transformation. In *Mathematical Morphology in Image Processing*, ed. Dougherty, E.R., Marcel Dekker Inc. **1993**, 600–612.

34. Beare, R.; Lehmann, G. The watershed transform in ITK — discussion and new developments. *The Insight Journal* **2006**, *6*.

35. Roerdink, J.B.T.M.; Meijster, A. The watershed transform: Definitions, algorithm and parallelization strategies. *Fundamental Informaticae* **2000**, *41*, 187–228.

36. Zheng, J.; Hryciw, R.D. Segmentation of contacting soil particles in images by modified watershed analysis. *Computers and Geotechnics* **2016**, *73*, 142–152.

37. Ridder, A.; Prifling, B.; Hilger, A.; Osenberg, M.; Weber, M.; Manke, I.; Birke, K.P.; Schmidt, V. Quantitative analysis of cyclic aging of lithium-ion batteries using synchrotron tomography and electrochemical impedance spectroscopy. *Electochemica Acta* **2023**, *444*, 142003.

38. Spettl, A.; Wimmer, R.; Werz, T.; Heinze, M.; Odenbach, S.; Krill III, C.; Schmidt, V. Stochastic 3D modeling of ostwald ripening at ultra-high volume fractions of the coarsening phase. *Modelling and Simulations in Material Science and Engineering* **2015**, *23*, 065001.

39. Kuchler, K.; Westhoff, W.; Feinauer, J.; Mitsch, T.; Manke, I.; Schmidt, V. Stochastic model for the 3D microstructur of pristine and cyclically aged cathodes in Li.ion batteries. *Modelling and Simulations in Material Science and Engineering* **2018**, *26*, 035005.

40. Burger, W..; Burge, M. *Digital Image Processing: An Algorithmic Introduction Using Java*, Springer **2016**.

41. Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **2006**, *58*, 1-35.

42. Wang, Z.; Bovik, A.; Sheik, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* **2004**, *13*, 600–612.

43. Chiu, S.N.; Stoyan, D.; Kendall, W.S.; Mecke, J. *Stochastic Geometry and Its Applications*. J. Wiley & Sons, 3$^{\mathrm{rd}}$ ed. **2013**.

44. Englisch, S.; Ditscherlein, R.; Kirstein, T.; Hansen, L.; Furat, O.; Drobek, D.; Leißner, T.; Apeleo Zubiri, B.; Weber, A.P.; Schmidt, V.; Peuker, U.A.; Spiecker, E. 3D analysis of equally X–ray attenuating mineralogical phases utilizing a correlative tomographic workflow across multiple length scales. *Powder Technology* **2023**, *419*, 118343.

45. Furat, O.; Leißner, T.; Ditscherlein, R.; Sedivy, O.; Weber, M.; Bachmann, K.; Gutzmer, J.; Peuker, U.A; Schmidt, V. Description of ore particles from XMT images, supported by SEM-based image analysis. *Microscopy and Microanalysis* **2018**, *24*, 461–470.

46. Furat, O.; Kirstein, T.; Leißner, T.; Bachmann, K.; Gutzmer, J.; Peuker, U.A.; Schmidt, V. Multidimensional characterization of particle morphology and mineralogical composition using CT data and R-vine copulas. *Minerals Engineering* **2024**, *206*, 108520.

Appendix

The structural similarity measure SSM, considered in Section 4.2.1 for computing the value of MSSM in Eq. (19), is defined as follows. Recall that in Eq. (19) we considered images $I^{\mathrm{f}} : W_{\mathrm{intact}} \to \{0, \ldots, 65565\}$ and $\widetilde{I}^{\mathrm{f}} : W_{\mathrm{intact}} \to \{0, \ldots, 65565\}$ with domain $W_{\mathrm{intact}} \subset \mathbb{Z}^3$. Furthermore, for each $x \in \widetilde{D}$ of some subset $\widetilde{D} \subset W_{\mathrm{intact}}$, we considered the similarity $\mathrm{SSM}_{I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}}(x)$ of I^{f} and $\widetilde{I}^{\mathrm{f}}$ at $x \in \widetilde{D}$, which is given by

$$
\mathrm{SSM}_{I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}}(x) = \begin{cases} \dfrac{(2\mu_{I^{\mathrm{f}}}(x)\mu_{\widetilde{I}^{\mathrm{f}}}(x) + c_1)(2\sigma_{I^{\mathrm{f}}, \widetilde{I}^{\mathrm{f}}}(x) + c_2)}{(\mu^2_{I^{\mathrm{f}}}(x) + \mu^2_{\widetilde{I}^{\mathrm{f}}}(x) + c_1)(\sigma^2_{I^{\mathrm{f}}}(x) + \sigma^2_{\widetilde{I}^{\mathrm{f}}}(x) + c_2)}, & \text{if} (\mu^2_{I^{\mathrm{f}}}(x) + \mu^2_{\widetilde{I}^{\mathrm{f}}}(x) + c_1)(\sigma^2_{I^{\mathrm{f}}}(x) + \sigma^2_{\widetilde{I}^{\mathrm{f}}}(x) + c_2) \neq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{36}
$$

where $\mu_{I^f}(x) \in \mathbb{R}$ is the local mean grayscale value of I^f at $x \in \widetilde{D}$ with

$$\mu_{I^f}(x) = (I^f * k)(x), \tag{37}$$

and $*$ denotes convolution. The function $k : K = \{-5, \dots, 5\}^3 \rightarrow \mathbb{R}$ in Eq. (37) is a (truncated) Gaussian kernel [40], which is given by

$$k(x) = \frac{1}{\nu} \exp\left(\frac{\|x\|}{2\sigma^2}\right) \tag{38}$$

for any $x \in K$ and $\sigma \geq 0$, where the normalizing constant $\nu > 0$ is chosen such that $\sum_{x \in K} k(x) = 1$. In our case, it turned out that $\sigma = 1.5$ is a suitable choice. Note that the convolution of I^f and k considered in Eq. (37) is defined as

$$(I^f * k)(x) = \sum_{(i,j,l) \in K} I^f(x_1 - i, x_2 - j, x_3 - l) \, k(i, j, l),$$

for each $x = (x_1, x_2, x_3) \in \widetilde{D}$, where we put $I^f(x_1 - i, x_2 - j, x_3 - l) = 0$ if $(x_1 - i, x_2 - j, x_3 - l) \notin \widetilde{D}$ for some $(i, j, l) \in K$. For further details regarding convolutions, we refer to [40]. The local mean grayscale value $\mu_{\widetilde{I^f}}(x) \in \mathbb{R}$ of $\widetilde{I^f}$ at $x \in \widetilde{D}$ is defined analogously. Furthermore, the local standard deviation $\sigma_{I^f}(x) \in [0, \infty)$ of I^f at $x \in \widetilde{D}$ is given by

$$\sigma_{I^f}(x) = ((I^f \cdot I^f) * k)(x) - \mu_{I^f}^2(x), \tag{39}$$

where $\cdot$ denotes the voxelwise multiplication of two images. The local standard deviation $\sigma_{\widetilde{I^f}}(x)$ of $\widetilde{I^f}$ is defined analogously. Finally, the local covariance $\sigma_{I^f, \widetilde{I^f}}(x)$ is given by

$$\sigma_{I^f, \widetilde{I^f}}(x) = ((I^f \cdot \widetilde{I^f}) * k)(x) - \mu_{I^f}(x)\mu_{\widetilde{I^f}}(x), \tag{40}$$

for each $x \in \widetilde{D}$. The constants c_1 and c_2 in Eq. (36) are chosen as in [42], i.e.,

$$c_1 = 0.01 \cdot (\max_{x \in \widetilde{D}} I^f(x) - \min_{x \in \widetilde{D}} I^f(x)) \quad \text{and} \quad c_2 = 7.65 \cdot (\max_{x \in \widetilde{D}} I^f(x) - \min_{x \in \widetilde{D}} I^f(x)).$$