

Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt

Crystal Shape Characterization: From Single Crystals to Crystal Aggregates

Tijana Kovačević

Vollständiger Abdruck der von der Fakultät
Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt
der Technischen Universität München zur Erlangung des akademischen Grades
eines Doktor-Ingenieurs
genehmigten Dissertation.

Vorsitzende/-r: Prof. Dr. Kay H. Schneitz

Prüfende/-r der Dissertation:

1. Prof. Dr.-Ing. Heiko Briesen

2. Prof. Dr. Xue Z. Wang

Die Dissertation wurde am 24.10.2017 bei der Technischen Universität München
eingereicht und durch die Fakultät
Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt
am 06.02.2018 angenommen.

Acknowledgments

My deepest gratitude goes to Prof. Dr.-Ing. Heiko Briesen for giving me the opportunity to work on this topic as well as for his insightful guidance and mentoring over the years. My thanks goes further to Prof. Dr. Xue Z. Wang for taking time to evaluate this thesis and to Prof. Dr. Kay Schneitz for presiding over the examination committee.

Special thanks goes to cooperation partners that made parts of this work possible. I thank Prof. Dr.-Ing. Kai Sundmacher and Dr. Andreas Voigt for the opportunity to use the laboratories at OVGU Magdeburg, as well as for their comments and ideas regarding the cooperation. I am especially grateful to Viktoria Wiedmeyer from OVGU Magdeburg for performing crystallization experiments as well as for her insights regarding the interpretation of the results. My further thanks goes to Prof. Dr.-Ing. Gerhard Schembecker, TU Dortmund University, for the opportunity to cooperate and his input regarding the work. Special thanks goes to further colleagues from Dortmund: Stefan Heisel for providing the data and sharing his insightful ideas regarding the joint publication, Lisa-Maria Terdenge, for initiating the cooperation, and Kerstin Wohlgemuth, for her guidance during the project. I gratefully thank Jonathan Schock, TUM, Chair of Biomedical Physics, for performing CT measurements, as well as Prof. Dr. Franz Pfeiffer, for giving us the opportunity for this cooperation. Finally, I thank Dr. Stefan Schorsch, from ETH Zürich for providing potash alum crystals used in one of the studies.

I thank my predecessor Dr.-Ing. Alexander Reinhold, for creating the framework that represents the basis for this work, as well as for his help and support during the early stages of my PhD. My thanks goes further to Dr. Vesna Müller for her mentoring.

I am grateful for the support of my colleagues who made my work at SVT TUM enjoyable, especially Michael Kuhn and Jörg Engstle who shared my office during this time. Further thanks goes to Michael Kuhn and Prof. Briesen for performing classification of crystal contact types in one of the studies. I thank Carsten Choszcz, who, along with Simon Röhrer, Viktoria Wiedmeyer and Stefan Heisel, proofread parts of this thesis. I further thank all students who contributed to this project. Daniela Eixenberger grew potash alum crystals used in one of the studies. Beatriz Antonacio Povedano performed during her internship some of the simulation studies discussed in chapter 3, which were later re-done using a more efficient code. Her work inspired further simulations and the discussion on distance correction in section 3.1.3.1.

I thank laboratory assistants René Urban, TUM, for performing a CT measurement, as well as Stefanie Markstein, OVGU Magdeburg, for her help with crystallization experiments.

The financial support of Deutsche Forschungsgemeinschaft (DFG) through the Priority Program SPP 1679, Dynamic Simulation of Coupled Solids Processes, project IDs BR-2035/9-1 and BR-2035/9-2 is gratefully acknowledged. I further acknowledge the publishing houses Elsevier and American Chemical Society for giving me permission to re-use parts of already published work, namely Kovačević et al. [1, 2, 3] in chapters 3 - 5 and Heisel et al. [4] in chapter 6.

Finally, I thank my family and friends for their constant support over the years.

Abstract

Crystallization is an important step in many industries, used for purification and product recovery. The finally obtained crystalline product passes a set of steps such as filtration or tableting, where crystal size, shape and degree of agglomeration play an important role. Furthermore, aggregation leads to liquid inclusion which decreases product purity. This motivates the need for characterization methods capable of characterizing the crystal shape and measuring the degree of agglomeration. In particular, the ability to fully describe the shape, orientation and position of each primary particle in a crystal aggregate can lead to better understanding of the aggregation mechanism, which would facilitate designing a product with specific properties.

Recently, significant amount of work has been dedicated to designing image-based methods for crystal characterization, including both aggregate recognition and detailed shape measurement of single crystals. Two-dimensional (2D) imaging techniques can be implemented in-line or on-line, thus providing a large amount of data and reducing the issues regarding sampling. However, they rarely provide the complete geometric information available with the off-line three-dimensional (3D) images. In this work, both approaches were investigated regarding their capabilities to characterize certain aspects of a sample of crystals.

Microcomputed tomography (μ CT) provides 3D images and is used for full shape characterization of each particle in a sample. A method for identifying the shape of single crystals from such images is developed and further extended to finding the shape, position and orientation of each primary particle in a crystal aggregate. In cooperation with the Chair of Process Systems Engineering, Otto-von-Guericke-University of Magdeburg, this method was applied to measure the orientation between primary particles in potash alum aggregates. In the conducted experiments it was observed that primary particles tend to have the same orientation significantly more often than expected when random collision is assumed.

In contrast to this detailed characterization of individual particles, 2D imaging can be adopted for measuring the agglomeration degree. Agglomeration degree is a property of a sample of crystals and requires simpler processing of a large number of individual particles. The method involves automatic classification of the imaged objects based on a computed set of image descriptors. This study was conducted in cooperation with the Laboratory of Plant and Process Design, TU Dortmund University. The accuracy of the artificial neural network classifier with respect to the selection of image descriptors and the training set composition was investigated.

Kurzzusammenfassung

Kristallisation stellt in vielen Branchen einen wichtigen Prozessschritt dar und wird im Bereich der Produktaufreinigung und Produktrückgewinnung eingesetzt. Die erzeugten Kristalle durchlaufen weitere Prozessschritte wie Filtration oder Tablettierung, wobei Größe, Form und Agglomerationsgrad der Kristalle eine wichtige Rolle spielen. Des Weiteren kann Aggregation zum Flüssigkeitseinschluss führen und dadurch die Produktreinheit verringern. Dies dient als Motivation für die Entwicklung neuer Messverfahren zur Charakterisierung von Kristallform und Agglomerationsgrad des Produktes. Besonders die Möglichkeit, sowohl Form als auch Orientierung und Position jedes einzelnen Primärpartikels im Kristallaggregat beschreiben zu können, ist dabei wichtig. Dadurch können die Aggregationsmechanismen genauer untersucht werden, was die Entwicklung von Produkten mit gewünschten Eigenschaften ermöglicht.

In den letzten Jahren wurde intensiv an der Entwicklung von bildbasierten Methoden zur Charakterisierung von Kristallen gearbeitet. Dies beinhaltete sowohl die Erkennung von Aggregaten als auch die detaillierte Formcharakterisierung der einzelnen Kristalle. Zweidimensionale (2D) bildgebende Verfahren können dabei sowohl in-line als auch on-line in den Prozess implementiert werden. Dadurch können zwar die Schwierigkeiten einer Probennahme reduziert und eine große Menge an Daten generiert werden. Eine komplette geometrische Information, wie sie durch off-line erstellte dreidimensionale (3D) Bilder vorhanden ist, ermöglichen diese Verfahren jedoch kaum. In dieser Arbeit wurden beide Ansätze hinsichtlich ihrer Möglichkeiten zur Charakterisierung bestimmter Kristalleigenschaften untersucht.

Mikrocomputertomographie (μ CT) erzeugt 3D Bilder und ermöglicht eine detaillierte Strukturaufklärung jedes einzelnen Kristalls einer Probe. Basierend auf diesen Bildern wurde ein Verfahren zur Bestimmung der Form der Einzelkristalle entwickelt und so erweitert, dass die Form, Orientierung und Position jedes einzelnen Primärpartikels im Aggregat beschrieben werden kann. In einer Kooperation mit dem Lehrstuhl für Systemverfahrenstechnik der Otto-von-Guericke-Universität Magdeburg wurde dieses Verfahren für die Messung der relativen Orientierungen zwischen Primärpartikeln in Kalialaun Aggregaten angewandt. Die durchgeführten Versuche zeigen, dass die Primärpartikel deutlich häufiger die gleiche Orientierung besitzen, als durch eine Annahme zufälliger Kollision zu erwarten wäre.

Im Gegensatz zu dieser detaillierten Charakterisierung der individuellen Partikel, können 2D bildgebende Verfahren zur Bestimmung des Agglomerationsgrades eingesetzt werden. Der Agglo-

merationsgrad stellt eine Eigenschaft einer Kristallprobe dar und kann durch einfachere Verarbeitung einer größeren Anzahl an Partikeln erfasst werden. Das entwickelte Verfahren beinhaltet eine automatische Klassifizierung der abgebildeten Objekte auf der Basis von Bilddeskriptoren. Diese Studie wurde in Kooperation mit dem Lehrstuhl für Anlagen- und Prozesstechnik der TU Dortmund durchgeführt. Die Genauigkeit der Klassifizierung mittels künstlicher neuronaler Netze wurde in Zusammenhang mit den gewählten Bilddeskriptoren und der Zusammensetzung des Training Sets untersucht.

Contents

Acknowledgments	i
Abstract	iii
Kurzzusammenfassung	v
List of Figures	xviii
List of Tables	xxi
Abbreviations	xxiii
Notation	xxv
1 Introduction	1
1.1 Motivation	1
1.2 Crystallization Fundamentals	3
1.3 Focus and Outline	6
1.4 Notes on Terminology	7
2 Methods for Crystal Size and Shape Characterization	9
2.1 Crystal Size and Shape	9
2.2 Size Measurement Techniques	10
2.3 Image-based Size and Shape Measurement Techniques	13
2.3.1 Image Acquisition	13
2.3.2 Image Processing	18
2.3.3 Image Analysis	23
2.3.4 Statistical Evaluation	29
2.4 Orientation Measurement	35
2.5 Goals and Open Questions	36

3	Shape Identification of Single Crystals	39
3.1	Algorithm	39
3.1.1	Identifying Face Normals	39
3.1.2	Establishing a Correspondence Between Measured and Modeled Faces	44
3.1.3	Identifying Face Distances	48
3.1.4	Fit Quality Measures	52
3.2	Validation	53
3.2.1	Simulated Data	53
3.2.2	Experimental Data	63
3.2.3	Conclusion	65
4	Shape Identification of Primary Particles in Crystal Aggregates	67
4.1	Algorithm	67
4.1.1	Pre-processing	67
4.1.2	Segmentation into Primary Particles	68
4.1.3	Shape Identification of Primary Particles	74
4.1.4	Fit Quality Measures	80
4.2	Validation	82
4.2.1	Simulated Data	82
4.2.2	Experimental Data	86
4.2.3	Conclusion	91
5	Orientation of Primary Particles in Crystal Aggregates	95
5.1	Aggregation and Growth Experiments	96
5.2	Preparation and Imaging	100
5.3	Image Processing	101
5.3.1	Image Pre-processing	101
5.3.2	Segmentation into Primary Particles	101
5.3.3	Shape Identification of Primary Particles	105
5.3.4	Fit Quality Measures	106
5.4	Disorientation Angle Distribution Measurement	107
5.4.1	Classification of Primary Particle Contact	107
5.4.2	Disorientation Angle	108
5.4.3	Disorientation Angle Distribution	111
5.5	Validation by Simulation	111
5.6	Results	112
5.7	Conclusion	116

6	Classification of Imaged Objects	119
6.1	Crystallization Experiments	120
6.2	Image Processing and Analysis	122
6.3	Training and Test Set Creation	124
6.4	Image Descriptor Ranking	127
6.5	ANN	128
6.6	Results	130
6.6.1	ANN Size	130
6.6.2	Combined Data Sets	131
6.6.3	Separate Data Sets	133
6.7	Conclusion	136
7	Conclusion and Outlook	139
	Bibliography	154
A	Projecting a Point onto a Polytope	155
B	Disorientation Angle Distribution	158

List of Figures

1.1	Different types of nucleation, adapted from [29, p. 172].	3
1.2	Different types of aggregation, according to [31].	4
2.1	Examples of three different shapes of potash alum crystals, using the model that defines 26 possible faces, divided into three face groups represented by different colors. In Figure 2.1c, a case where the green and orange face groups are grown out, is presented. Their face distances are marked by dots. In this case, the shape is invalid and the face distances of orange and green faces must be re-scaled to coincide with the vertices and edges of the polytope.	11
2.2	"Work-flow" for obtaining size and shape information from imaging methods.	14
2.3	Principle of backprojection, adapted from [81, p. 207].	17
2.4	Examples of backprojection for different number of directions. Adapted from [81, p. 207].	17
2.5	Separating touching objects by a combination of watershed transform and distance transform. (a) Original image. (b) Distance transform of the inverted image. (c) Sign of distance transform changed. (d) Background set as basin. (e) Watershed transform: obtained regions illustrated by different colors, whereas watershed pixels are shown in black. Note that distance transform images were adapted for illustration purposes.	20
2.6	The task of classification is to determine the class (here, triangle or square) of an object based on the value of the considered variables. In this case, classes are linearly separable, as depicted by the line.	31
2.7	An illustration of a neural network with an input layer, several hidden layer and an output layer.	33
3.1	Procedure for identifying the shape of single crystals.	40
3.2	HEALPix discretization of a sphere into elements of equal area [134]. The 12 principal fields are presented as bold lines. Each principal field is divided into N_{side}^2 sub-fields drawn by thin lines, leading to a total of $12 \cdot N_{\text{side}}^2$ points. Borders of the HEALPix discretization were drawn using the HEALPix library for MATLAB by Y. Naruse [135]. In the right figure, points marked by blue represent the spherical cap centered at the orange point, where $\alpha_{\text{cap}} = 15.87^\circ$	42

3.3	An illustration of the reduced Hough transform space $\tilde{b}(\phi_i, \theta_i)$ for a simulated and a real crystal, given in the bottom row. Peaks corresponding to identified face normals are marked by red dots. Crystals are octahedral, whereas the considered shape model contains 26 crystal faces, resulting in $\alpha_{\text{cap}} = 15.87^\circ$ and a spherical cap illustrated in Figure 3.2b. The figure was obtained by interpolating the angle values to a uniform grid. .	43
3.4	An illustration of the matching process. Crystal model containing 26 face normals that belong to three face groups is shown on the left. A real crystal is shown on the right, where the identified face normals are marked by arrows. All identified face normals belong to the "blue" face group; other face groups from the model crystal have not been identified on the surface of the real crystal. Reprinted with permission from [2]. Copyright (2016) American Chemical Society.	44
3.5	The procedure for matching identified to model face normals and finding the rotation matrix [1].	47
3.6	A procedure for finding the face distance in cases where the rotated model normal has no match among the identified normals. The plane corresponding to the most-populated Hough bin is presented by black triangles. The plane corresponding to the highest point density is represented by black squares. It can be seen that while there are more triangles than squares, the most populated plane would underestimate the true face distance. .	48
3.7	Illustration of the distance correction issue in 2D. The object is represented in gray, the pixels in white and pixel centers as blue points. In the left figure, an infinitely discretized Hough transform procedure will identify lines through pixel centers, so that the object bounded by orange lines is finally obtained. The distance of each line from the orange middle point is, therefore, half a pixel smaller than the true facet distance. In the right figure, the object bounded by orange lines would be too small, as well. However, if the distance is increased so that the lines touch the outmost pixel boundary points, the obtained object would be too large, as shown by green lines.	50
3.8	Potash alum crystal shapes used in the validation of the shape identification procedure. .	55
3.9	Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1-1, given in Table 3.1.	56
3.10	Examples of shape identification results for one crystal orientation and different volume-equivalent sphere diameters D . Simulation and shape identification parameters correspond to case S1-1, given in Table 3.1. The simulated crystals are rendered in white, while the identified shapes are presented as black polytopes.	57
3.11	Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1-5, given in Table 3.1.	57

3.12 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1f-5, given in Table 3.1.	58
3.13 Examples of shape identification results for one crystal orientation and different volume-equivalent sphere diameters D . Simulation parameters for the corresponding cases are given in Table 3.1. The simulated crystals are rendered in white, while the identified shapes are presented as black polytopes.	59
3.14 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S2-1, given in Table 3.1.	59
3.15 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S2-5, given in Table 3.1.	60
3.16 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S2-8, given in Table 3.1.	60
3.17 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3-1, given in Table 3.1.	61
3.18 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3-5, given in Table 3.1.	62
3.19 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3e-1, given in Table 3.1.	62
3.20 Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3e-5, given in Table 3.1.	62
3.21 Experimentally measured potash alum particles used for validating the method for identifying the shape of single crystals. Visualization done by MyVGL [139]. Reprinted with permission from [1]. Copyright (2014) American Chemical Society.	64
3.22 Shape identification in case of single crystals: examples where the procedure performed well and the crystals were highly regular.	65
3.23 Shape identification in case of single crystals: examples where the procedure performed well and crystals contained small particles at the surface.	66
3.24 Shape identification in case of single crystals: examples where the underlying crystal did not correspond to a well-defined single particle.	66

3.25	Shape identification in case of an aggregate, obtained with the algorithm for single crystals (left) and with the algorithm for aggregates(right). Error measures regarding the right crystals are defined in chapter 4 on crystal aggregates.	66
4.1	A crystal before (a) and after (b) the filtering procedure	68
4.2	Identifying the shapes of primary particles in crystal aggregates [2].	69
4.3	Segmentation of an aggregate into primary particles [2]. (a) Watershed segmentation before concavity expansion. (b) Concavity expansion. (c) Watershed segmentation after concavity expansion. (c) Watershed voxels are re-assigned. (d) Small and large watershed regions are merged. (e),(f) Two iterations of the procedure that recombines large watershed regions. (g) Shape identification. Concavity points are shown as blue crosses in Figure 4.3a, while the corresponding concavity search masks are presented as black boxes.	70
4.4	(a) Concavity expansion procedure deletes foreground voxels in the concavity direction, defined between the concavity point \mathbf{p}_c (orange) and the middle point of the background part of the mask \mathbf{p}_b (blue). The mask is shown in black. The procedure results in a thin hole through the object near each concavity point (b). Reprinted (adapted) with permission from [2]. Copyright (2016) American Chemical Society.	72
4.5	Faulty region, presented in orange in the left image is identified and deleted, resulting in the right image. Concavity points are shown in red in the left image.	77
4.6	The blue and green regions (upper left) belong to the same primary particle, so that the identified polytopes overlap strongly (upper right). The algorithm detects such situations and merges the regions (lower left) to obtain only one identified polytope (lower right).	79
4.7	Before (a) and after (b) moving and re-sizing the smaller primary particle shape fit.	80
4.8	An illustration of the effect of primary particle growth on one aggregate from data sets A1-0, A2-0 and A3-0, defined in Table 4.1.	82
4.9	An illustration of the effect of surface roughness on one crystal from different data sets of type A2, defined in Table 4.1.	83
4.10	Segmentation (left) and shape identification (right) for one aggregate from sample A1-0. Two green primary particles were wrongly concatenated. The aggregate is considered to have an unsuccessful shape identification by both sets of quality measures.	86
4.11	Segmentation (left) and shape identification (right) for one aggregate from sample A3-0.2. The orange primary particle was wrongly concatenated with the part of another particle. However, the shape of the orange particle was identified correctly in spite of this segmentation error. The aggregate is considered to contain primary particles where no shape identification was possible by the quality measures developed for simulated data. It is considered to have a successful shape identification by the quality measures for real data.	87

4.12 Segmentation (left) and shape identification (right) for one aggregate from sample A3-0.4. The segmentation is almost correct, but a part of the orange particle was assigned to the blue one. However, the shape of the blue particle was identified as too small. The aggregate is considered to have an unsuccessful shape identification by the quality measures for simulated data. The shape identification is considered successful by the quality measures for real data, as $q_{v,r} = 0.088$ and $q_{\text{surf},r,\text{all}} = 0.05$	87
4.13 3D volume of the particles used for validating the aggregate shape identification procedure (a), as well as one slice through it (b). Rendering was performed by Jonathan Schock, at the Chair of Biomedical Physics, TUM, using AVIZO Fire [146]. Reprinted with permission from [2]. Copyright (2016) American Chemical Society.	89
4.14 Shape identification in case of crystal aggregates: examples where the procedure performed well. Error measure $q_{\text{surf},r,\text{fit}}$ was left out in cases where it was equal to $q_{\text{surf},r,\text{all}}$	92
4.15 Shape identification in case of crystal aggregates: examples where the aggregate is considered to contain primary particles where no shape identification is possible.	93
4.16 Shape identification in case of crystal aggregates: examples where the procedure performed badly.	93
5.1 Crystallization vessel and the stirrer geometry.	98
5.2 Concentration and supersaturation profiles measured in the experiments, based on the data computed by Viktoria Wiedmeyer. The time of solution and crystal sampling is marked by black circles and crosses, respectively. Seeding was performed at time $t = 0$ s for all experiments. Reprinted from [3] with permission from Elsevier.	98
5.3 An image of the crystals taken 37s after adding seeds in the experiment R (a), and 270s after adding seeds in the experiment M3 (b). The images in the lower row contain particles where the area-equivalent circle diameter is greater than $200\mu\text{m}$ marked in blue. The images in the upper row are reprinted from [3] with permission from Elsevier.	99
5.4 μCT preparation procedure. Crystals are placed on an adhesive tape so that there is enough space between separate particles (a). The adhesive tape is then rolled around the toothpick, ensuring that the crystals remain separated (b). Reprinted from [3] with permission from Elsevier.	100
5.5 One horizontal slice from experiment M5. The porous structure is the toothpick, the onion-like layers are the layers of adhesive tape and the white areas are crystals. Reprinted from [3] with permission from Elsevier.	102
5.6 Examples of particles from experiment M10 that were excluded from the shape identification procedure as their shape did not correspond to either agglomerated or single crystals [3].	102

5.7	A symmetrical (a) and an asymmetrical crystal (b,c), both observed in the experiment M3 [3]. In case of (a) and (b), the identified shapes presented in black are obtained using the asymmetrical crystal model, as explained in section 5.3.3. The shape identification in (c) is performed using the method based on symmetry from chapter 4. Note that the symmetrical crystal model contained 26 faces, while the asymmetrical model consisted of 8 faces.	103
5.8	The procedure for identifying the shapes of primary particles in crystal aggregates using user interaction.	104
5.9	(a) The result of the segmentation procedure is displayed twice, as rendered crystals (left) and as a point cloud (right). Different colors are used to ensure sufficient contrast. The user can choose to concatenate the regions or add new regions. In this case, the user chose to concatenate the green and orange regions, upon which the segmentation was accepted. The result is shown in (b). Reprinted from [3] with permission from Elsevier.	106
5.10	Examples of contact classes. Reprinted from [3] with permission from Elsevier.	108
5.11	Reference orientation of cube and regular octahedron. Reprinted from [3] with permission from Elsevier.	110
5.12	Examples of disorientation angles for pairs of primary particles observed in the experiment M10 [3].	111
5.13	Mean error measures for 100 simulated potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1-1, given in Table 3.1, except that no symmetry conditions are imposed, in accordance to the shape identification procedure presented in this chapter.	113
5.14	DAD computed on the simulated polytopes (a) and DAD computed on the polytopes identified from the simulated 3D images (b). The blue line is the analytical solution for uniform orientations of constituting primary particles [131, 132]. The bar graphs contain data on both grown-in and lightly-touching primary particle pairs. Reprinted (adapted) from [3] with permission from Elsevier.	113
5.15	DAD for the four conducted experiments given in Table 5.1 is presented in orange. The first column represents grown-in agglomerate pairs, the second are the pairs of lightly-touching crystals, and the third column is the distribution obtained when considering grown-in agglomerates and lightly-touching pairs together. The blue line represents the theoretically expected distribution [131, 132]. The number of primary particle pairs included in the computation can be seen below the graphs. Reprinted (adapted) from [3] with permission from Elsevier.	115

- 5.16 Volume-equivalent sphere diameter measured using μ CT images (upper row) and a projected area-equivalent sphere diameter measured using a QICPIC video towards the experiment end (lower row), both for experiment M5. Number-based histograms are shown to the left, while volume-based ones are shown to the right. Object areas in pixels, needed to compute the figures in the lower row, were calculated by Viktoria Wiedmeyer and provided to the author. 117
- 6.1 Finding concavity depth and concavity points. (a) Red line: object boundary. Green line: boundary of the convex hull image. (b) Yellow dotted line: one segment of the convex hull boundary. Large green circles: ends of the convex hull boundary segment. Light blue dots: one object boundary segment. Large light blue circles: concavity points. Yellow rhomboids: corresponding nearest points on the convex hull boundary. Orange dotted lines: correspondences between boundary points and the nearest convex hull points; their length is the concavity depth. Reprinted from [4] with permission from Elsevier. . . . 125
- 6.2 Artificial neural network used in this study, containing one input, one hidden and one output layer. 129
- 6.3 Average classification accuracy obtained for different numbers of neurons in the hidden layer. Error bars represent standard deviation obtained upon repeating the training process 10 times for each number of hidden neurons. Classification was performed using the training set "C-Ala-200-TR" and the corresponding test sets of type "C". All 19 image descriptors were employed. Reprinted from [4] with permission from Elsevier. 132
- 6.4 Bubbles from data set Ala1 that were correctly (a) and wrongly (b) classified using DFA, as observed by the Dortmund research group [4]. Reprinted from [4] with permission from Elsevier. 132
- 6.5 ANN results based on the training set C-Ala-200-TR using different numbers of image descriptors. Image descriptors are ranked according to "sa" (a), "cb" (b) as well as "mn" (c). The error bars represent the standard deviation of the values for different test sets. The target value $\overline{PI} = 0.9$ is visualized by a horizontal line. Reprinted from the supplementary material of [4] with permission from Elsevier. 134
- 6.6 ANN results based on the training set C-AA-200-TR using different numbers of image descriptors. Image descriptors are ranked according to "sa" (a), "cb" (b) as well as "mn" (c). The error bars represent the standard deviation of the values for different test sets. The target value $\overline{PI} = 0.9$ is visualized by a horizontal line. Reprinted from the supplementary material of [4] with permission from Elsevier. 135

6.7 ANN results based on the training set C-Ala- N -TR using different training set sizes ($N_{\text{total}} = 9N = 27, 90, 180, 360, 1800$) and numbers of image descriptors; for better visualization, the image descriptor subset selected ("sa" ranking order) is highlighted with a gray bar. The error bars represent the standard deviation of the values for different test sets. The target values $\overline{PI}_{\text{All}} = 0.9$ and $\delta Ag = 0.1$ are visualized by solid horizontal lines. Reprinted from [4] with permission from Elsevier. 135

6.8 ANN results based on the training set C-AA-n-TR using different training set sizes ($N_{\text{total}} = 9N = 27, 90, 180, 360, 1800$) and numbers of image descriptors; for better visualization, the image descriptor subset selected ("mn" ranking order) is highlighted with a gray bar. The error bars represent the standard deviation of the values for different test sets. The target values $\overline{PI}_{\text{All}} = 0.9$ and $\delta Ag = 0.1$ are visualized by solid horizontal lines. Reprinted from [4] with permission from Elsevier. 136

6.9 ANN results based on the "separate training and test sets" for L-alanine, denoted by "S" in Table 6.5. First three image descriptors from the ranking order "sa" were selected for the analysis. Reprinted from [4] with permission from Elsevier. 137

6.10 ANN results based on the "separate training and test sets" for adipic acid, denoted by "S" in Table 6.5. First three image descriptors from the ranking order "mn" were selected for the analysis. Reprinted from [4] with permission from Elsevier. 137

A.1 A point can be projected onto a face (left), edge (middle) or vertex (right), depending on its location. The influence region of a face, edge and vertex is illustrated in blue, green and orange, respectively. Influence regions are defined by their face normals, illustrated as arrows with different colors, as well as corresponding points on each region face, illustrated with the same colors as the normals. An example of a point being projected onto a face and edge is presented using white stars. Any point in the vertex influence region is projected to that vertex, so that no examples are illustrated for that case. 156

List of Tables

2.1	An overview of investigated particle properties, variables and classification methods in case of automatic particle classification.	36
3.1	Parameters for simulation and shape identification of single crystal data sets where the shapes are illustrated in Figure 3.8.	55
3.2	Parameters of the μ CT measurement.	63
3.3	Parameters for shape identification of experimental data: validation of the single crystal procedure.	64
4.1	Parameters for simulation and shape identification of aggregate data sets. p represents the surface roughness parameter $p_{\text{roughness}}$. There were four different surface roughness levels for each data set, so that 12 data sets are finally obtained.	83
4.2	Shape-identification success evaluated for the 12 simulated data sets using the quality measures for simulated data, defined in Section 4.1.4.1, and the quality measures for experimental data, defined in Section 4.1.4.2. First three columns show the simulated case and the parameters used in sample generation. The fourth and seventh columns represent the percentage of aggregates where the shape of all primary particles was correctly identified. The fifth and eighth columns represent the percentage of aggregates containing at least one badly identified primary particle. The sixth and ninth columns show the percentage of aggregates where the shape of some primary particles could not be identified, while it was identified correctly for the remaining particles.	85
4.3	Percentage of successful shape identifications per aggregate class consisting of two to five primary particles. The values are computed for each group of simulated data sets by considering all four data sets obtained by varying the surface roughness parameter. . . .	85
4.4	Parameters of the μ CT measurement for validation of the aggregate shape identification procedure.	88
4.5	Parameters for shape identification of experimental data: validation of the aggregate procedure.	89
5.1	Parameters of the conducted seeded batch cooling crystallization experiments [3].	96

5.2	Parameters of the μ CT measurements performed by Jonathan Schock, Chair of Biomedical Physics, TUM. Reprinted (adapted) from [3] with permission from Elsevier.	101
5.3	Number of crystals where the shape identification was successful for each primary particle, no shape identification was possible for some regions ("missing") and the shape identification was unsuccessful for some primary particles. Reprinted (adapted) from [3] with permission from Elsevier.	107
5.4	Number of primary particle pairs from experiments R, M3, M5, and M10 that were classified as "grown-in", "lightly-touching" and "none" by three different experts, denoted by H, M and T. Also, the number of pairs that were classified into the same class by all three experts, denoted by "all" [3]. ¹	109
5.5	Number of different crystal types in each sample. Columns 2-4 show single crystals, single crystals with attached small particles, and single crystals with particles for which the fitting procedure failed to produce a result. Columns 5-7 show aggregates for which a fit was possible for all primary particles, aggregates that also contained small particles, and aggregates that contained primary particles for which no fit was possible. Columns 8 and 9 show particles for which no shape fit was possible because each particle was too small and the fitting procedure failed to produce a result. Note that "single" refers to crystals for which the shape identification was possible for only one particle, whereas in case of "aggregates" it was possible for multiple particles. Reprinted (adapted) from [3] with permission from Elsevier.	116
6.1	Parameters of the L-alanine cooling crystallization and adipic acid gassing crystallization experiments. † Static drying. ‡ Fluidized bed drying. *Mixture of ethanol/water, volume fraction 4/1. **Pure ethanol. ◇ Rough estimation (process was performed until sample was completely dry). Data provided by Stefan Heisel. Reprinted (adapted) from the supplementary material of [4] with permission from Elsevier.	121
6.2	Product characteristics of the examined samples of the chemical systems L-alanine/water and adipic acid/water, calculated in Dortmund using image analysis and preliminary DFA classifiers not presented here. Data provided by Stefan Heisel. Reprinted (adapted) from the supplementary material of [4] with permission from Elsevier.	121
6.3	Examples for processed objects (after binarization), imaged by QICPIC. Reprinted (adapted) from [4] with permission from Elsevier.	122
6.4	All 19 image descriptors, their definitions and their type: 1. "Size"; 2. "Roughness"; 3. "Proportion"; 4. Area fraction. † A_{convex} is the convex hull area which is used for calculation but not used as an additional image descriptor. *Calculations based on values provided by ImageJ. **Calculated with MATLAB. Reprinted (adapted) from [4] with permission from Elsevier.	123

-
- 6.5 "Combined" ("C") and "separate" ("S") training ("TR") and test ("TE") set configurations for L-alanine ("Ala") and adipic acid ("AA"). "s" refers to single crystals, "a" to aggregates, "b" refers to bubbles, and N is the number of objects in the set (N varies between 3 and 200). Reprinted (adapted) from [4] with permission from Elsevier. 126
- 6.6 Different \overline{PS} ranking orders for L-alanine (Ala) and adipic acid (AA) crystals, based on data provided by Stefan Heisel [4]. "sa" and "cb" refer to the \overline{PS}_{sa} and \overline{PS}_{cb} values obtained from discrimination of single crystals/agglomerates and crystals/bubbles, respectively. "mn" is the arithmetic mean of "sa" and "cb". The ranking is from best to worst. 129
- 6.7 Lowest number of image descriptors N_{desc} needed so that $\overline{PI}_{All} \geq 0.9$ and \overline{PI}_{All} does not change for more than 2 % upon adding an additional image descriptor. Results are presented for different image descriptor ranking orders. Computations regarding DFA are provided by Stefan Heisel. Ranking orders leading to the smallest number of needed descriptors are presented with bold letters. Reprinted (adapted) from [4] with permission from Elsevier. 131

Abbreviations

μ CT	Microcomputed Tomography
2D	Two-dimensional
3D	Three-dimensional
ANN	Artificial Neural Network
CT	Computed Tomography
DFA	Discriminant Factorial Analysis
DAD	Disorientation Angle Distribution
FBRM	Focused Beam Reflectance Measurement
OVGU	Otto-von-Guericke-University Magdeburg
PCA	Principal Component Analysis
PS	Proportional Similarity
PSD	Particle Size Distribution
PSSD	Particle Size- and Shape Distribution
SEM	Scanning Electron Microscopy
SVM	Support Vector Machine
TEM	Transmission Electron Microscopy
TU	Technische Universität / Technical University
TUM	Technical University of Munich

Notation

Accents

- Averaged quantity
- ~ Quantities/vectors that have been moved/rotated/modified
- ^ Quantities/vectors that have been moved/rotated/modified

Greek Symbols

α	Angle between two face normals / Euler angle
β	Euler angle
γ	Euler angle
δA_g	Error in the measurement of agglomeration degree
ϕ	Polar angle
ρ	Distances used in Hough transform
$\Delta\rho$	Distance discretization step in Hough transform
σ	Supersaturation
Θ	Disorientation angle / Azimuthal angle (chapter 3)

Latin Symbols

Ag	Agglomeration degree
\mathbf{a}_i	i -th face normal of a polytope or crystal, column vector
\mathbf{A}	A matrix of face normals, $\mathbf{A} = [\mathbf{a}_i^T]$, dimensions $N_H \times 3$
b	Number of points in a Hough transform bin
c	Correction factor (chapter 3) / concavity (chapter 4) / concentration (chapter 5)
d	Distance
\mathbf{d}	Movement direction
e	Tolerance
\mathbf{e}	Polytope edge, defined by the difference of the coordinates of its vertices
h_i	i -th face normal of a polytope or crystal (chapters 3 to 5) / neuron input (chapter 6)
\mathbf{h}	A column vector of face distances h_i
\mathbf{h}_C	A column vector containing one face distance h_i per face group
I	Image
L	Objective function
$M_{\mathbf{h}_C \rightarrow \mathbf{h}}$	A matrix that transforms a vector \mathbf{h}_C into the corresponding vector \mathbf{h}
$M_{\mathbf{h} \rightarrow \mathbf{h}_C}$	A matrix that transforms a vector \mathbf{h} into the corresponding vector \mathbf{h}_C
N	number of objects of a certain type
\mathbf{n}	Directions of face normals used in Hough transform
o	Degree of overlap
PI	Performance index of the classification
PS	Proportional similarity
r	Resolution of a 3D image given as voxel length in μm / Particle convexity
\mathbf{R}	Rotation matrix
s	Input into a neuron in the output layer
\mathbf{S}	A mapping or filter matrix (chapter 3) / symmetry operation matrix (chapter 5)
t	Target output of a neuron
V	Volume
\mathbf{v}_i	Coordinates of the i -th vertex of a polytope or crystal, column vector
\mathbf{V}	A matrix of vertex coordinates, $\mathbf{V} = [\mathbf{v}_i^T]$, dimensions $N_V \times 3$
\mathbf{p}	Coordinates of a point
q	Fit quality measure
\mathbf{x}	Origin of the coordinate system
y	Output of a neuron in the output layer

Chapter 1

Introduction

1.1 Motivation

Crystallization, a process of crystal formation and growth, is an important step in pharmaceutical, chemical and food industry. Crystals represent highly ordered solid structures of compound-specific atoms, ions or molecules where the incorporation of impurities is unlikely. Therefore, crystallization can be used to separate the desired chemical compound from a mixture or to increase the purity of the compound. The importance of crystallization is exemplified by the fact that most small-molecule pharmaceuticals are delivered to patients in crystalline form [5]. In food industry, crystals are used to improve the food texture, so that an ice-cream without ice crystals would taste chalky, whereas cocoa butter crystals provide snap upon breaking chocolate [6]. Particle size has traditionally been considered as one of the most important characteristics of the crystalline product. A well-known everyday example again involves ice cream, where ice crystals larger than 50 μm created during storage lead to unpleasant coarse texture [7][8, p. 321-323], [9, p. 264-267]. Particle size furthermore influences the ease and efficiency with which the crystallized particles are processed in order to obtain the final product, such as a drug tablet or a bar of chocolate. In pharmaceutical industry, small crystals can be advantageous due to their better solubility and bioavailability properties, while being more difficult to process [5, 10]. A wide particle size distribution (PSD) containing a large amount of fine particles can cause clogging of filters and pneumatic conveyors or reduce the porosity of the filter cake, leading to long filtration times [11]. Apart from particle size, the impact of particle shape on both the final product and downstream processing has been gaining attention with the advancements in measurement technologies based on image processing [12]. Particle shape influences powder flowability and tableting behavior, so that the related properties are better for plate-like ibuprofen crystals than for needle-like ones [13]. Similarly, needle-like particles are more difficult to filter than equant particles [5]. Particle shape can be indirectly related to several other particle properties, such as product purity. Purity may be reduced through aggregation, a process where individual crystals stick together to form larger particles, by allowing inclusions of the impure mother liquor [14] or antisolvent [10]. Physico-chemical properties of crystals are further influenced by polymorphism. Polymorphs are crystals of the same chemical compound that

have different molecular structure, thus leading to a different shape. Polymorphs can have different solubility and bioavailability, so that transitioning between polymorphs or sudden crystallization of the undesired polymorph can lead to withdrawal of the drug and the need for reformulation [5, 15].

Previous examples served to motivate the importance of particle size and shape in different industries. Gaining control over these properties can lead to the manufacturing of the desired product at reduced costs. Achieving this goal requires knowledge that can be divided into four basic categories. First, mechanistic understanding of the fundamentals of both crystallization and post-processing steps must be gained. This involves investigating the influence of controllable parameters, such as the solution composition, temperature and the hydrodynamic conditions, on the properties of the obtained product. The second step is the measurement of kinetics, involving nucleation, growth, aggregation and breakage rates. The third category involves developing models and modeling strategies that enable further study of the influencing parameters. Finally, methods for process optimization and control based on the knowledge from the previous three stages should be applied to find the path towards the desired product. The first two categories rely on the ability to characterize different aspects of the process. Studying the fundamentals of crystallization typically requires detailed characterization of the size, shape and surface of single particles. Kinetic measurements, on the other hand, are often conducted for the whole crystal population, so that less detailed characterization of these properties on the single particle scale is often sufficient. The size and shape information must be complemented by the measurements of other relevant properties, such as the solution concentration, or the polymorphic state of the particles. A large amount of effort has been invested in the recent years in the development of such characterization methods, known as process analytical technologies (PAT)s [16]. Special importance regarding particle shape is attributed to the methods based on image analysis, enabling direct shape observation. There is a wide range of available image acquisition technologies, enabling both off-line imaging of particle samples [17, 18, 19, 20] as well as in-situ imaging [20, 21, 22, 23] of the particles inside the crystallizer vessel, where intermediate solutions involving sampling loops [24, 25, 26, 27] are also possible. The vast majority of the developed methods is based on two-dimensional (2D) images, leading to loss of information regarding three-dimensional particles. The alternative are three-dimensional (3D) imaging techniques which have recently become available [19]. These methods provide detailed, otherwise unattainable information, especially in the case of complex structures such as aggregates containing several primary particles. However, they can only be used off-line and require particle sampling. Therefore, a trade-off is sought between the richness of the particle shape information and the ability to obtain it fast, without introducing any impurities or inaccuracies through sampling. This work presents methods for extracting particle shape information on both scales. Microcomputed tomography (μ CT) is used to obtain the full 3D crystal polytope, both in case of single crystals and crystal aggregates, where each of the primary particles is thus described. On the population scale, a method for extracting the agglomeration degree of the product from an in-situ imaging system is investigated.

The following sections of this chapter give a brief introduction regarding the milestones in the quest for particle size and shape control. Special focus is given to crystal aggregation, as the methods

presented in this work enable further insights into aggregate formation, as illustrated in the final case study. A detailed literature review regarding imaging methods is given in chapter 2.

1.2 Crystallization Fundamentals

Crystallization literature typically considers nano- to centimeter-sized crystal particles, while natural crystals can grow up to several meters in length [28]. Internally, crystal structure is a rigid lattice, incorporating the constituting entities, namely atoms, ions or molecules, in an order manner [29, p. 1]. Macroscopically, ideal single crystals are convex polytopes exhibiting one or more groups of symmetrical faces, also called crystal forms [29, p. 11], discussed in more detail in section 2.1. Crystallization involves four basic mechanisms. Nucleation is the process of crystal formation. A prerequisite is either supersaturation, meaning that an excess amount of solute is dissolved in the solvent, or supercooling, where the solution is cooled under the freezing temperature. Nucleation can be subdivided into primary and secondary [29, p. 172] as illustrated in Figure 1.1. In case of primary nucleation, no crystal particles are present in the solution prior to the nucleation event. The excess constituting entities will assemble together upon collision and re-dissolve, until some critical size of the obtained particle is reached, known as the nucleus [29, p. 173-175]. This is known as homogeneous nucleation and is extremely rare in a technical environment as some amount of foreign particles, such as dust, is always present in the solution. These particles act as nucleation centers leading to the phenomenon known as heterogeneous nucleation [29, p. 182]. Heterogeneous nucleation can also be induced by introducing gas bubbles into the system, which is known as gassing crystallization [30]. Secondary nucleation involves nuclei formation through abrasion of existing crystal particles by fluid shear or collision with other particles, vessel walls or stirrer. It can therefore occur at a smaller level of supersaturation than primary nucleation [29, p. 185-187].

Upon nucleation, newly formed crystals continue to grow in the supersaturated solution. This is the second and most studied mechanism occurring during crystallization. Crystals grow by incorporating the constituting elements into the crystal lattice. During growth, the crystal shape changes, where the

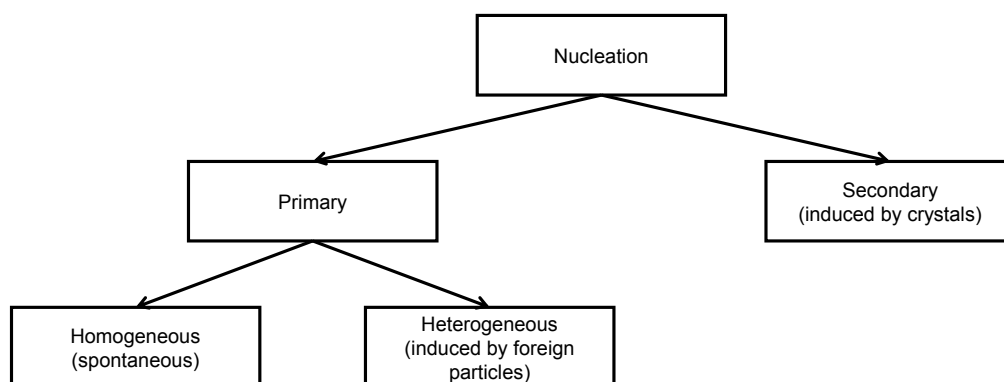


Figure 1.1: Different types of nucleation, adapted from [29, p. 172].

fastest growing crystal faces may eventually disappear [29, p. 203]. Several mechanistic explanations have been suggested in the literature and confirmed by experimental observation. They differ in the explanation of the sites where the new atom, ion or molecule is integrated into the lattice, as well as in variables that influence this incorporation. The resulting kinetic models can be either mechanistic or purely empirical. An often used model is the supersaturation (σ) dependent growth rate $G = k_g \sigma^g$, where k_g may further incorporate temperature dependence through the Arrhenius law [31, p. 129]. The growth rate can be defined using different particle size measures so that determining the growth rate relies on techniques for measuring particle size. Values k_g and g are fit parameters, whereas the value of g suggest whether the growth rate is limited by the diffusion of the constituting entities to the crystal surface or by their integration into the crystal lattice [31, p. 129]. The growth rate may further depend on crystal size [29, p. 237-238]. In order to obtain a narrow PSD, thus achieving favorable behavior during processing, crystallization is often seeded by adding crystals into the prepared solution at the beginning of the crystallization run. The crystallization conditions are then kept within the metastable zone, defining the zone of supersaturation where nucleation is improbable [29, p. 117].

The third crystallization phenomenon is crystal breakage, occurring upon collision of a crystal with other crystals or equipment parts. The amount of breakage depends on the material parameters and the hydrodynamic and mechanical conditions. It alters the particle size and shape, which can be desirable in case of needle-like particles as it reduces their aspect ratio. Crystal breakage need not be done in a crystallizer and is often achieved after crystallization by milling or sonication in order to achieve particles with smaller aspect ratio and the target PSD [5]. Attrition can be considered as a type of crystal breakage, leading to rounded particles [32].

The fourth crystallization phenomenon is aggregation, also known as agglomeration, where two or more particles join together to form a larger particle. This phenomenon is given special focus in this work. Aggregation can be divided into primary and secondary [31, p. 156], as illustrated in Figure 1.2. The typically used terms aggregation and agglomeration refer to the secondary phenomenon, where an aggregate is formed upon collision of two particles. Secondary aggregation is further subdivided into orthokinetic and perikinetic, the former referring to particles larger than 1 μm moving due to fluid shear

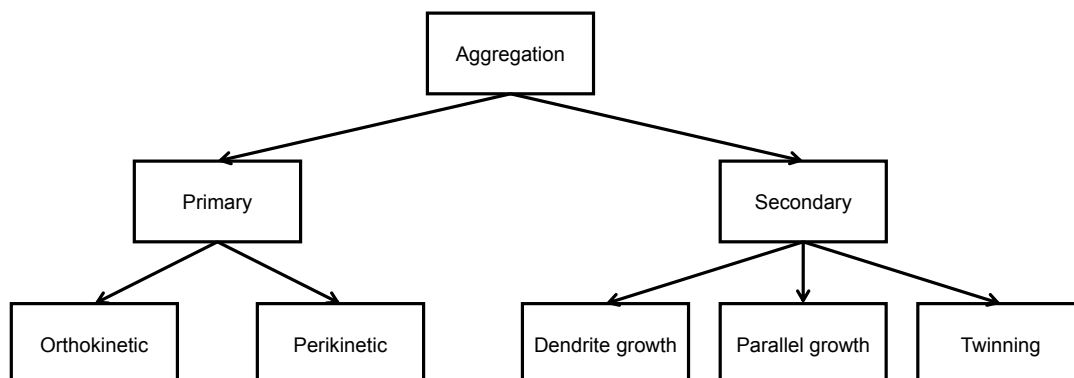


Figure 1.2: Different types of aggregation, according to [31].

and the latter to smaller particles in Brownian motion [31, p. 160]. Crystallization therefore typically considers orthokinetic aggregation. The ability of a pair of particles to form an aggregate depends on whether they can be cemented together fast enough to withstand the disruptive hydrodynamic forces [33, 34, 35]. This depends on the material parameters and hydrodynamic conditions as well as on the size, shape and orientation between the colliding particles [34, 36]. Pratola et al. [37] measured the strength of aggregates grown under varying supersaturation conditions and found that the strength increases with increasing supersaturation. Furthermore, aggregation behavior changes with a change of solvent composition [38, 39, 40]. In contrast to the secondary mechanism, primary aggregation is caused by mal-growth where the final crystal appears to consist of more than one primary particle [31, p. 157]. Three types of primary aggregation can be differentiated according to Jones [31, p. 157-160] and are briefly introduced here. Some crystals exhibit dendrite growth, where several levels of needle-like particle branches are visible. A typical example are snow flakes. Parallel growth is a term given to structures where several crystals are inter-grown on top of each other so that their faces and edges are parallel. The phenomenon was observed for alums and it is possible for two different substances to form such structures together. The final type of primary aggregation are twin crystals, where two or three particles are joined together in a symmetrical, ordered manner. Particles may also grow through each other, creating the so-called penetration twins. The formation of such twins can be encouraged by the presence of certain impurities [41]. Twinning can be confirmed by looking at the interface between the primary particles by a transmission electron microscope (TEM) and studying the TEM diffraction patterns [34, 42]. It is often not possible to differentiate between the primary and secondary aggregation based on simple observation [31, p. 157]. Collier et al. [34] investigated the orientations between primary particles during precipitation and found alignment tendencies for experiments conducted at lower ionic strength, where twinning could be excluded as a mechanism in some of the cases.

Crystal size can be controlled by adjusting the parameters that influence the supersaturation, such as the temperature profile, solution composition or the PSD and the amount of the seed crystals. In case of crystals exhibiting more than one face group, these parameters can also be used to control crystal shape. A common approach involves a temperature profile that leads to cycles of growth and dissolution, which was shown to broaden the domain of achievable crystal shapes [28, 43]. Shape can be further influenced by the choice of solvent [39] or through additives and impurities [24].

The strategies for controlling crystal size and shape can be divided into open-loop and closed-loop. In open-loop approaches, the optimal variables, such as the temperature profile, are computed beforehand. In closed-loop control, the target and the current state are periodically compared and the process variables are manipulated to stir the process in the desired direction. These variables are either the temperature profile or the amount of some compound such as the solvent, antisolvent or additive [44]. The new values for the process variables can be computed either using classical proportional-integral (PI) controller [24], not requiring any information on the crystallization kinetics, or a controller based on the underlying process model [43]. Therefore, both open- and closed-loop control may rely on crystallization modeling.

The goal of crystallization modeling is often to describe the change of particle size- and/or shape

distribution (PSSD) over time. This is achieved with a population balance model. In case of a perfectly mixed batch reactor, this is described by [45]:

$$\frac{\partial \mathbf{n}(t, \mathbf{L})}{\partial t} + \sum_{i=0}^N \frac{\partial G_i \mathbf{n}(t, \mathbf{L})}{\partial L_i} = B(t, \mathbf{n}) - D(t, \mathbf{n}), \quad (1.1)$$

followed by an appropriate set of initial and boundary conditions, as well as a mass balance. Similar equations for continuous crystallization or spatial-dependent scenarios can be derived. Here, \mathbf{n} is the PSD/PSSD, defined over the chosen set of N particle size measures L_i , t is the time and G_i are the growth rates. B and D are the birth and death terms, describing the creation and loss of particles through nucleation, aggregation and breakage. The latter two involve integrals and thus complicate the numerical solution. The equation can be simplified by ignoring certain terms, so that a simulation of particle growth can be obtained by setting B and D to 0. Models of different complexity are obtained depending on the dimensionality of \mathbf{n} and the underlying kinetic models. High-dimensional models are obtained when considering crystal growth in face directions, thus providing a full description of the shape of single crystals [46, 47]. Faces that grow out and thus disappear from the crystal surface may pose further challenges in this case [47]. In aggregation models, aggregation rate is defined using an aggregation kernel that models the frequency of collision between particles, scaled by the efficiency of these collisions, representing the probability that the aggregate survives the collision [33, 35, 36, 48, 49, 50]. Aggregation kernels typically consider spherical particles and an aggregation efficiency based on an appropriate bridge geometry [33, 35, 48, 50] and hydrodynamics. While the orientation of the particles towards the shear field can be taken into account [33, 35, 50], the actual particle shape and the orientation between primary particles are ignored. Notable exceptions include the work of Ochsenbein et al. [36], where a shape-dependent kernel is proposed and the work of Briesen [49, 51], modeling particle geometry upon aggregation using a Monte Carlo simulation. However, neither allows for a full consideration of particle geometry and mutual particle orientation after collision along with the implications concerning the mass balance.

1.3 Focus and Outline

The previous sections motivated the need for detailed characterization of particle size and shape. In the recent years, significant development in this field was achieved, as will be reviewed in Chapter 2. The available methods focus on the characterization of single crystals based on 2D imaging techniques. As introduced above, crystallization processes often result in aggregated particles. Aggregation is often undesired as it reduces the product purity through liquid inclusions [10, 14] but it can also be deliberately used to obtain larger particles with favorable downstream processing [10]. Characterizing crystal aggregates based on 2D imaging is difficult due to inherent loss of 3D information through projection. Newly emerging 3D imaging techniques provide the full 3D shape information that can enable analyzing particle shape in detail in case of both single crystals and crystal aggregates. The drawback of these

techniques is the long time needed for preparation and measurement, so that on-line particle characterization is not possible. Therefore, the choice of suitable method depends on the required level of detail on the one side and, on the other side, the necessity to extract less-detailed information regarding a large number of particles without the need for withdrawing them from the solution.

In this thesis, both approaches are investigated. The detailed 3D particle characterization is based on μ CT measurements, conducted in cooperation with the group of Prof. Pfeiffer (Technical University of Munich). Chapter 3 presents a method for extracting the full 3D crystal polytope from the obtained 3D images. The method is extended towards the analysis of crystal aggregates in Chapter 4. In Chapter 5, this method is applied to analyze the orientations between primary particles in aggregates of potash alum crystals, based on experiments conducted in cooperation with the group of Prof. Sundmacher (Otto-von-Guericke-University Magdeburg). Chapter 6 investigates a method for measuring the agglomeration degree of a product based on a less-detailed processing of a large amount of 2D images. The method uses artificial neural network (ANN)-based particle classification. The underlying crystallization experiments using L-alanine and adipic acid were conducted by the group of Prof. Schembecker (Technical University of Dortmund), and a comparison with another classification method investigated by their group is performed. Finally, Chapter 7 concludes the work.

1.4 Notes on Terminology

In the following, the terminology that is used in this work is presented. We consider the crystal shape, or morphology, to refer to the relative combination of its face distances. Knowing the full crystal shape and size is equivalent to having the full information about the crystal polytope. The crystal polytope can be described either as a set of vertices, which is called V -representation, as it will be introduced in the following chapter, or a set of face directions and their distances from a chosen origin, which is known as the H -representation. The term face group describes faces that grow symmetrically and is equivalent to the term crystallographic form used in the literature [29, p. 11]. Aggregate is used to describe a group of touching or overlapping primary particles, where agglomerate refers to an aggregate where primary particles have grown into each other. The opposite of an aggregate is a single crystal, while crystal can refer to both types of particles. The term agglomeration degree is used to describe the relative amount of aggregates in a crystalline product, in order to remain consistent with the term used in the prior publication [4] and the literature.

In discussions on particle imaging and classification, the term "image descriptor" refers both to shape descriptors and size measures.

In equations, bold symbols are used for vectors and matrices. Italic symbols in indices refer to numbers, so that d_i is the i -th variable of type d or the i -th element of vector \mathbf{d} . Non-italic symbols represent simple names. Unless stated otherwise, vectors are considered to be column vectors. Dimensions of a matrix with a rows and b columns are denoted by $a \times b$, and may be indicated in the index of the matrix symbol, such as $\mathbf{A}_{a \times b}$.

Chapter 2

Methods for Crystal Size and Shape Characterization

2.1 Crystal Size and Shape

Single crystals can be modeled as convex polytopes. Thus, having the full 3D information about the crystal size and shape is equivalent to knowing the parameters of the corresponding polytope. The parameters depend on the used crystal representation. Here, we consider three crystal representations, described in detail by Reinhold et al. [52]. A V -representation is the set of points defining the convex hull of the polytope. Given that a crystal is convex, it consists of the vertices \mathbf{v}_i of the polytope, gathered in a matrix $\mathbf{V}_{N_V \times 3} = [\mathbf{v}_i^T]$, where N_V is the number of vertices. An H -representation is obtained by noting that a crystal consists of all points \mathbf{p} fulfilling:

$$\mathbf{A}\mathbf{p} \leq \mathbf{h}, \quad (2.1)$$

where $\mathbf{A}_{N_H \times 3} = [\mathbf{a}_i^T]$ is a matrix containing N_H face normals \mathbf{a}_i , $\|\mathbf{a}_i\| = 1$, and \mathbf{h} is a vector containing the corresponding face distances h_i . The H -representation is therefore defined by \mathbf{A} and \mathbf{h} . Note that, as in the case of V -representation, the same crystal shape is obtained by translating the crystal for some vector \mathbf{x}_0 [52, 53]:

$$\mathbf{h}' = \mathbf{h} + \mathbf{A}\mathbf{x}_0. \quad (2.2)$$

The third crystal representation is based on crystal symmetry. We assume that the crystal is positioned so that its center of mass is in the origin of the coordinate system. In this case, face distances of all crystal faces in the same face group are identical due to symmetry. The vector \mathbf{h} can therefore be reduced in dimension by considering only one face distance for each of the C face groups, yielding the N_C -dimensional vector \mathbf{h}_C [52, 53]:

$$\mathbf{h} = \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}} \mathbf{h}_C. \quad (2.3)$$

Here, $\mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}$ is a mapping matrix of size $N_H \times N_C$, containing values 1 where face distances from \mathbf{h}_C should be copied to \mathbf{h} and zeros otherwise. The H_C -representation is then defined by \mathbf{A} , $\mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}$ and \mathbf{h}_C .

In order to facilitate computation, it is often necessary to convert between different crystal representations. This functionality is enabled in the MATLAB framework of Reinhold [52], which is used as the basis for this work. This framework uses the concepts of convex geometry and the Cddlib library [54]. The framework considers the H - and V -representations, whereas the H_C -representation can be obtained by translation and applying the appropriate group mapping matrix. Among others, it enables conversion between the two representations, measure computation for the two representations and the reduction of the H -representation to a set of non-redundant inequalities.

During growth, crystal faces may disappear or new faces may appear. This creates challenges in both crystallization modeling and experimental characterization. The chosen crystal model should cover all crystal faces that may arise during the considered experiment. This can be of special importance in the case where the number of different achievable shapes is large, as illustrated in Figure 2.1. Crystals that change shape in terms of changing the set of visible edges can cause problems in measure computation if it is based on an assumed crystal shape [47]. Furthermore, when modeling crystal growth, crystal faces that grow out may lead to an invalid set of face distances. This is demonstrated in Figure 2.1c, where the green and orange face groups are grown out and their distance is such that the faces do not touch the crystal polytope. These face distances must be reduced so that the faces touch the polytope at vertices and edges, which may additionally require adapting the growth rate of these faces in simulation studies [47]. The morphology domain describing possible \mathbf{h} -values can be decomposed into regions where different edge combinations occur [53, 55].

In case of experimental characterization, the chosen method should be able to find the subset of faces that appear in the observed crystal. Furthermore, validity of the obtained shape should be ensured in order to obtain correct measures and use the the information for crystallization modeling. Most size and shape measurement techniques are incapable of measuring the crystal polytope and provide some other size and shape information instead. In the following section, an overview of the available measurement techniques and measurable properties is presented.

2.2 Size Measurement Techniques

If all particles in the considered population have the same shape, the population can be completely described by a particle size distribution (PSD). The PSD can be expressed over some suitable characteristic length, such as the particle diameter in case of perfect spheres or the side length in case of cubes. The choice of the characteristic length depends on the particle shape as well as on the underlying measurement principle.

A commonly used approach is to represent particles by the diameter of a sphere with an equivalent volume. This is especially suitable for techniques that measure a quantity that directly depends on the particle volume. One such technique is the Coulter counter, used for sizing particles between 0.6 μm

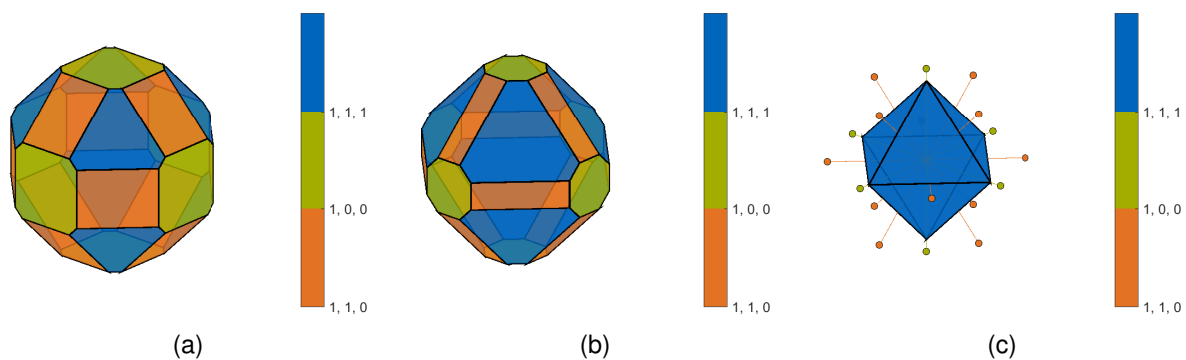


Figure 2.1: Examples of three different shapes of potash alum crystals, using the model that defines 26 possible faces, divided into three face groups represented by different colors. In Figure 2.1c, a case where the green and orange face groups are grown out, is presented. Their face distances are marked by dots. In this case, the shape is invalid and the face distances of orange and green faces must be re-scaled to coincide with the vertices and edges of the polytope.

and $1200 \mu\text{m}$ [56, p. 186]. Particles are suspended in an electrolyte liquid and pass through an orifice between two electrodes. The change of the impedance between the electrodes is proportional to the particle volume [56, p. 186]. The suspension must be dilute enough to minimize the probability of two particles passing through the orifice simultaneously and thus being counted as one.

A popular in-line measurement technique used for monitoring crystallization processes is laser reflectance, typically involving the Focused Beam Reflectance Measurement (FBRM) probe from Mettler Toledo. The probe sends a beam of light that scans particles in a circle whose diameter is assumed to be significantly larger than the particle length [57]. The light reflected off a particle is captured by a sensor. The resulting measure is called the chord length and corresponds to the length of the light beam segment that was intersected by the particle. It therefore depends both on the particle orientation and its location towards the beam, so that different chord lengths can be observed for the same particle [58]. Chord length distribution can be obtained from PSD by adopting either an empirical approach [59, 60] or a model based on particle geometry and further assumption regarding the measurement system [57, 61, 62]. The inverse problem of estimating the PSD from the measured chord length distribution is ill-posed [57]. The quality of the estimation depends both on the chosen approach regarding the forward-problem [62], as well as on the chosen numerical procedure [59].

Laser diffraction methods are based on the fact that the intensity of the light diffracted by a particle is related to the particle size. The methods use either Mie or Fraunhofer light scattering theory, both assuming spherical particles [63, p. 97]. Fraunhofer theory represents an approximation applicable for particles larger than several micrometers and does not require information about the refraction index, while Mie theory enables the measurement of particles as small as $0.1 \mu\text{m}$ [63, p. 92-100]. Particle size distribution can be obtained numerically from the measured intensity distribution of diffracted light by solving an inverse problem [58]. The measurement is performed in a dilute suspension and is fast enough to be performed on-line [56, p. 182]. The particle size distribution is expressed in the equivalent sphere diameter and is affected both by the fact that the particles are not spherical as well as by their

orientation at the time of measurement [58].

Ultrasound attenuation spectroscopy technique correlates particle size to the ultrasound attenuation coefficient measured at different frequencies. Similarly to optical techniques described above, particle size distribution is obtained by numerically solving an inverse problem [64]. The advantage of the method is its applicability on a broad size range (10 nm to 3 mm) in highly concentrated suspensions [56, p. 190]. However, in order to perform the measurements, several mechanical, thermodynamic and transport parameters must be known, rendering the method more difficult to apply for monitoring dynamic systems with changing parameters such as crystallization processes [64].

Sieving is another commonly used technique where particles are allowed to pass through a stack of sieves with different mesh sizes. In order to facilitate passage through the sieves, vibrations can be employed or the particles can be suspended in a liquid [56, p. 162]. The particle size is characterized by the equivalent sieve aperture diameter, representing the diameter of the largest sphere that would pass through a sieve [29, p. 63]. The used sieve mesh widths are between 5 μm and 125 mm [56, p. 162]. The technique is better suited for equant particles than for elongated or plate-like ones [56, p. 163]. In crystallization sieving is typically used to obtain a narrow size distribution of the seed particles, which are then characterized by the sieve mesh width, as for example seen in [25, 65].

Sedimentation techniques are based on the fact that the settling velocity of spherical particles in a fluid depends on the fluid properties and the particle density and diameter [56]. Therefore, under certain conditions, for known fluid properties and particle density, the diameters of the particle can be computed from the settling velocity measurements. If the particle is not spherical, the obtained value is called the Stokes diameter and is the diameter of an equivalent sphere that would sink with the same velocity [56, p. 166]. Particle size measurements that use this principle monitor the amount of settled particles over time, using different types of forces to control the settling behavior.

Even in case of an accurate measurement of the equivalent sphere diameter, such as using the Coulter counter, care must be taken when measuring growth rate parameters based on one size measurement only. This was shown by simulation for needle-like particles [58]. The change of shape during growth leads to a broadening or shrinking of the PSD, which may be misinterpreted as evidence of agglomeration, breakage, growth-rate dispersion or size-dependent growth [58]. Additionally, as discussed above, many measurement techniques are ill-suited for particles that deviate strongly from a spherical shape.

Specialized techniques have been developed for nanometer-sized particles. A popular method is dynamic light scattering. It is based on the fact that suspended nanoparticles exhibit Brownian motion. As small particles move faster than the large ones, particle size can be correlated to the fluctuations of scattered laser light intensity signal [56, p. 189]. Obtaining the particle size distribution involves solving an inverse problem and is problematic in case of multi-modal and wide distributions [56, p. 189]. The method assumes that particles are spherical and the obtained measure is an equivalent hydrodynamic radius [66]. Furthermore, sample parameters such as temperature, solvent viscosity and refractive index must be known in order to perform the calculations, while the sample must be dilute [66].

The size of micrometer- and millimeter-sized particles can also be determined by analyzing images

of particles if the imaging is performed in a controlled manner and the size of the pixel is known. However, such techniques are often used to extract not only size, but also shape information, which is why they are discussed in detail in the following section.

2.3 Image-based Size and Shape Measurement Techniques

Imaging represents a direct measurement technique as size and shape are directly observed. The process of obtaining the information about particle shape can be viewed as involving four steps, namely image acquisition, image processing, image analysis and statistical evaluation, even though the distinction between these steps may be fuzzy. As illustrated in Figure 2.2, the process can be ended at each stage, depending on the required information. Image acquisition is sufficient in cases where a human operator observes the images and reaches conclusions about the process. An example involves observing the effect of the preparation procedure on the seed crystals and studying its impact on the PSD and quality of the final product [67]. The image processing step involves converting the image to grayscale or binary, detecting individual particles and finding significant features within the image such as corners, lines or planes. While this is typically not the case, the procedure could be ended at this stage by, for example, discarding all images taken by an in-line microscope that do not contain any particles. Thus, the goal of image processing is to provide input for the analysis that follows. The image analysis step can be viewed as the process of obtaining the final information about each imaged particle. This could be the chosen particle size measures, such length and width, a set of shape factors or descriptors [68, 69], the fully reconstructed 3D shape [1, 19] or the size and shape of each primary particle in an aggregate [2]. The procedure ends here when the detailed information about one imaged particle is needed, such as the full 3D shape of a newly crystallized compound or the polymorph of an exemplary crystal. The last step compiles the information obtained from a population of particles and may correlate it to other measured or simulated quantities, such as the solution concentration. This can involve simply gathering the chosen measures into a distribution [17, 26], determining the growth rate [25, 45], computing the agglomeration degree [68] or detecting the polymorph based on the images of other particles [70]. Depending on the type of information that is to be extracted, this step can require parameter optimization [25, 45] or the use of statistical and pattern recognition tools [68, 69]. The four steps presented in Figure 2.2 and the resulting methods are reviewed below in more details.

2.3.1 Image Acquisition

The available imaging methods can be divided into two-dimensional (2D) and three-dimensional (3D), depending on the type of image that is finally obtained. In certain cases, however, it is possible to reconstruct full 3D shape information from a single 2D image [25, 71]. Stereoscopic methods use two images taken from different positions in order to obtain depth information. For the sake of simplicity, these methods are here discussed together with the 2D methods.

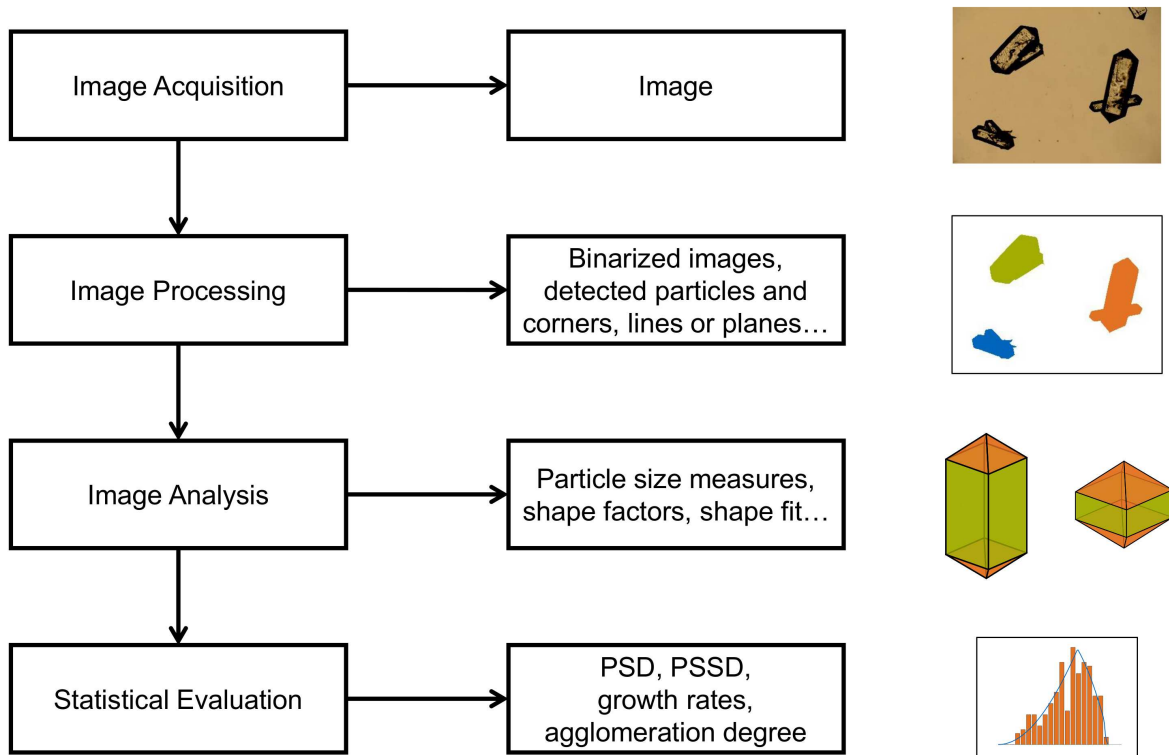


Figure 2.2: "Work-flow" for obtaining size and shape information from imaging methods.

2.3.1.1 2D Imaging Methods

In 2D imaging, the image of a particle can be considered to represent a 2D projection of the 3D particle onto the imaging plane. Therefore, the projection will depend on the orientation of the particle towards the camera, unless the particle is a perfect sphere, in which case the projection is always a circle. This is the reason why it is difficult to extract information about the 3D particle shape from 2D images. The developed imaging methods can either be based on always imaging particles from the same direction, or imaging particles from different directions and relying on statistics to obtain the 3D information.

Static 2D imaging methods take images of particles that are not moving [56, p. 177]. A typical example is the optical microscope, where the particles are fixed on the microscope slide. The advantage of this technique is that particles settle in a stable position [12]. Needle-like particles will therefore lie on their longest side, so that the analysis of static images allows one to correctly estimate particle length [72]. Some particles, such as potash alum, position themselves on the largest face and the symmetry of the particles allows one to estimate the full 3D shape from these images [71]. The main drawback of the static image analysis is that it requires a method to separate particles on the microscope slide so that they do not touch or overlap. Failing to do so can result in erroneously identifying a pair of touching particles as a single particle and wrongly measuring the size and shape information. Such dispersion is not easy to achieve and must in some cases be performed manually [17]. Moreover, many microscopes require an operator to manually move the stage and focus the particles, so that imaging

a particle distribution becomes a time-consuming process. However, automatic setups which perform both dispersion and focusing, such as Morphologi G3 of Malvern, are available [23] and enable measurement of a larger number of particles than manually operated microscopy procedures. A further drawback is that particles must be sampled during the experiment. Sampling can introduce impurities into the system and result in a non-representative sample of the particle population. The obtained particles must be prepared for imaging, which typically involves washing and drying. These processes may change the product properties, such as the amount of aggregates, thus increasing the dispersion challenge. Patience and Rawlings [24] attempted to overcome this issue in an on-line setup by continuously pumping the suspension with the particles through a flow-cell during a crystallization experiment. The flow through the cell was stopped during imaging, allowing the particles to settle on the cell bottom.

Further static imaging methods involve scanning electron microscopy or transmission electron microscopy. However, due to complicated preparation procedures involved, these techniques are typically used for visual observation of the product and not for quantitative size and shape analysis.

Contrary to static techniques, dynamic imaging methods take images of moving particles [56, p. 177]. Particles therefore have varying orientations towards the camera. Camsizer (Retsch) takes images of falling particles in air and uses a vibratory feeder to disperse the particles. The amount of vibration of the feed can result in different particle size distributions [73]. If the same convex particle is imaged from different orientations, the 3D particle area S can be related to the average projection area \bar{A}_p by the Law of Cauchy [56, p. 14] as:

$$S = 4\bar{A}_p. \quad (2.4)$$

According to this law, the area-equivalent circle diameter based on \bar{A}_p is equal to the area-equivalent diameter of the spherical particle, based on S [56, p. 14]. Similarly, for a given PSSD, it is possible to derive the probability density function for the projected area under the assumption that all particle orientations are equally likely. Then, the quality of a measurement device can be evaluated by comparing the predicted to the obtained area distribution [73].

QICPIC (Sympatec) is capable of imaging particles in a wet, additionally to a dry, dispersion. The slurry is pumped by a peristaltic pump through a measurement cuvette where the images are taken. The setup requires a low suspension density [74]. It can also be implemented on-line, where an external sampling loop sends solution samples through the QICPIC cuvette during the crystallization experiment [25].

Further imaging techniques that can be considered dynamic are those that take in-situ images of the particles in the crystallizer. These techniques involve either a probe inserted directly into the vessel [61], or a camera mounted in front of the reactor wall [20, 21, 22]. As the distance of the particles towards the camera varies, the images may contain rather blurry particles along with the ones that are in focus [20]. Edges of the particles may be partially badly visible [20]. The background intensity may vary with the lighting and hydrodynamic conditions in the reactor, rendering segmentation difficult [20, 44]. The quality of the resulting images is therefore lower than in case of the on-line techniques involving an external sampling loop [25]. Furthermore, high suspension densities can lead to particles overlapping

in the image plane, so that it is difficult to detect and measure single crystals [21, 22, 25, 44].

More information about the true 3D size and shape of a particle can be obtained if two images of the particle are taken from different directions. Such setups are referred to as stereo vision [23, 70, 75, 76, 77] or stereoscopic [26, 27]. Different approaches have been explored in the literature. The group of Mazzotti used a self-constructed on-line setup that samples the slurry during the experiment and takes images in a flow-cell. The first setup [26] involved a dilution loop to ensure low suspension density. A mirror was used to split the light beams so that the particle is illuminated from two perpendicular directions at the same time. Another mirror ensures both images are conducted to the same imaging plane where they are simultaneously captured by one camera. The setup was further adapted by improving the optical properties, removing the dilution loop and using two cameras instead of mirrors [27]. A stereoscopic setup involving two cameras imaging the particle simultaneously was suggested by Wang et al. [75]. The setup, using an angle of 22° between the two cameras, was applied both for monitoring the growth of a single crystal in a flow-cell [77], as well as for monitoring a crystal population, where the cameras were placed just outside the reactor wall [23, 76].

2.3.1.2 3D Imaging Methods

Tomographic techniques construct a 3D image of an object by considering a set of 2D slices or projections. The techniques differ in the types of 2D slices and the underlying physical principles used to obtain them. Unlike a 2D image, a 3D image provides full geometric information about the object. However, 3D imaging methods require larger measurement times and elaborate sample preparation, so that they currently cannot be used on-line or in-situ.

Confocal microscopes enable obtaining a precisely focused image of a certain slice of the specimen, while the specimen must be coated with fluorescent dye. A 3D image of the specimen can therefore be obtained by stacking slices obtained on various heights in the imaging (z) direction, where the resolution in the z -direction is usually lower than that in the xy -plane [78]. Singh et al. [19] used this technique to image crystals. It enabled computing Miller indices of the faces and determining the crystal polymorph.

Computed tomography (CT), also known as x-ray tomography, uses a set of conventional 2D x-ray images to reconstruct the full 3D structure of the object [80, p. 1]. An overview is given in the article by Barigou and Douaire [79] and the book by Stock [80], whereas the basic information is summarized here. 2D x-ray images, referred to as projections, are obtained from different directions and represent the attenuated version of the incident radiation.

A set of projections can be obtained either by rotating the imaging setup, as is the case in conventional medical CTs, or by rotating the object which is to be imaged, typically around one fixed axis. Microcomputed tomography (μ CT) is the high-resolution variant of the technique, using voxels whose length is at least as small as $50\text{-}100\mu\text{m}$ [80, p. 1]. It can thus be used to obtain structural and geometric information in engineering [82], geology [83], biology, materials- [84] or food science [7, 79, 85], as well as to measure particle size distributions [82, 83, 84]. The attenuation of monochromatic x-rays after traveling some distance x through a homogeneous object can be described by the Beer-Lambert

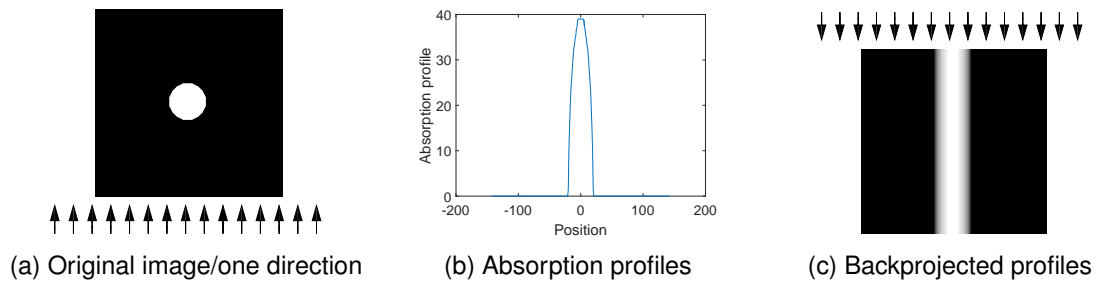


Figure 2.3: Principle of backprojection, adapted from [81, p. 207].

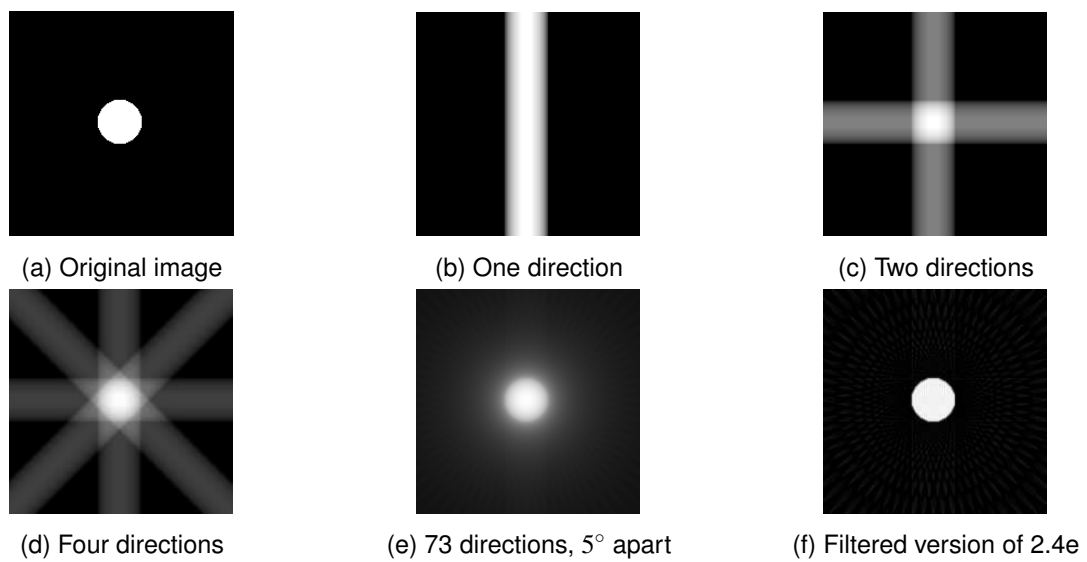


Figure 2.4: Examples of backprojection for different number of directions. Adapted from [81, p. 207].

equation:

$$I = I_0 e^{-\mu x} \quad (2.5)$$

where I_0 is the initial intensity and μ is the attenuation coefficient, depending on the material properties and the x-ray wavelength [80, p. 13][79]. In case of an inhomogeneous object, an integral $\int \mu(x) dx$ is considered instead of the constant value μ [79]. This is further complicated by the fact that x-rays exhibit scattering or refraction [80, p. 14]. Furthermore, conventional μ CT systems use polychromatic radiation, so that the equation 2.5 must be further considered for each wavelength present in the radiation [79]. The goal of the imaging is to retrieve the absorption coefficient $\mu(\mathbf{p})$ at each 3D voxel position \mathbf{p} from a set of 2D absorption profiles. This is typically done by the filtered backprojection process [80, p. 24-31]. A simplified illustration of this principle is given in the books by Stock [80, p. 24-28] and Gonzalez et al. [81, p. 205-208] and is briefly summarized here. It is considered that the image of a 2D object is to be reconstructed from a set of 1D absorption profiles obtained for different directions. For each considered direction, the absorption profile is measured along a set of parallel lines that cover the image, as shown in Figure 2.3. Therefore, for each direction this results in a function depicting the absorption value for each considered line position. The back-projection principle is based on the image obtained by projecting this function back to the image plane for each direction. In the resulting image, high absorption values are depicted by bright pixels. Superimposing such "backprojected" images for different projection directions results in very bright pixels inside the object and very dark pixels otherwise, as shown in Figure 2.4. The approximation is improved with the number of used projections [81, p. 207]. However, some blurring still remains in the image and must be resolved by suitable filtering [80, p. 24-28]. The quality of the obtained μ CT image therefore depends on the number of parameters, among others the number of projections and the difference in absorption coefficients of the phases that are to be discriminated in the image.

The advantage of μ CT is the ability to obtain full information about the particle size and shape. This can be used to measure a particle size distribution or compute a shape factor distribution for a sample of particles [84]. When measuring a population of particles, results depend on the chosen segmentation algorithm ensuring individual particles are correctly recognized [84]. This source of error also exists in 2D imaging methods and is reduced by an adequate sample preparation such as manual or automatic dispersion. However, such automatic dispersion techniques cannot be applied to 3D samples.

2.3.2 Image Processing

The goal of image processing, as considered here, is to isolate different particles and prepare them for size and shape measurement, which will be the task of image analysis. The procedure may start by filtering in case of a noisy image. This is followed by segmentation, where the objects of interest are isolated. Finally, the objects may be further processed to prepare them for size and shape measurement. In the following, classical image processing procedures are discussed. Their application to obtaining information about crystal size and shape is presented in the section 2.3.3 on image analysis. The presented algorithms can be applied to both 2D and 3D images. For the sake of convenience, the

smallest element of an image is termed pixel in both cases, whereas voxel is used when 3D images are discussed specifically.

2.3.2.1 Image Filtering

The simplest filters are linear filters, where a mask, typically of size $n \times n$, where n is odd, is centered on each image pixel. More information can be found in an image processing textbook, e.g. [81, p. 64-71]. The value of the current pixel is changed to correspond to the weighted sum of the pixel values marked by the mask, where the weights are provided by the mask. A mean filter uses equal weights so that each pixel in the resulting image represents an average value of the masked pixels in the original image. A Gaussian filter uses weights derived from the Gaussian distribution, so that the pixel itself, being the center of the mask, has the highest value and the weight values decrease further from the center. Masks can also be designed to approximate the gradient or the Laplacian of the image [81, p. 75], which is used in edge detection and image enhancement.

While these filters reduce image noise, a typical artifact of linear filtering is image blurring. A notable filter which is very effective for salt-and-pepper noise is the median filter [81, p. 80]. The gray value of each pixel is modified to the median of the gray values marked by the mask. This filter is a non-linear filter as finding the median is not a linear operation.

2.3.2.2 Image Segmentation

Thresholding Segmentation typically starts by assigning each pixel of the image either to foreground or background, thus creating a binary image with only two pixel values. Pixels are assigned based on comparison of their gray values with some threshold t . Thresholding therefore represents the simplest type of classification problem, where the samples (pixels) are to be classified into two classes (foreground and background) based on one variable (gray value). Classification techniques will further be discussed in section 2.3.4.3, where they are employed to determine the particle type.

A threshold for the given image can be selected manually by the operator or it may be computed automatically, based on the gray value histogram of the given image. Several techniques for automatic thresholding can be found in image processing textbooks [81, 86]. A notable example is the Otsu threshold which minimizes the within-class variance:

$$\sigma_W(t)^2 = w_0(t)\sigma_0(t)^2 + w_1(t)\sigma_1(t)^2 \quad (2.6)$$

and maximizes the between-class variance [87] [86, p. 14-18], [81, p. 516-519]:

$$\sigma_B(t)^2 = w_0(t)(\mu_I - \mu_0(t))^2 + w_1(t)(\mu_I - \mu_1(t))^2. \quad (2.7)$$

Here, w_0 and w_1 are the probabilities of classes 0 (background) and 1 (foreground), after applying the computed threshold t , and μ_0 , μ_1 and μ_I are the mean gray values of the two classes and the whole image, respectively, while σ denotes variance of the appropriate class. This means that the obtained

classes should be narrow, with mean values as far apart as possible [86, p. 14]. The threshold is obtained by computing σ_B for all possible gray value levels t as threshold and picking the one producing the maximal value.

Another popular method is hysteresis thresholding [88, p. 125]. A pixel is set to foreground if its gray value is higher than or equal to a high threshold t_H . Then, pixels with a gray value higher than a low threshold t_L , that are connected to the foreground pixels, are also set to foreground. The advantage of this method is that connected foreground elements are obtained [88, p. 125]. It is thus less likely that a relevant object in an image has been split into several segments.

Labeling Upon obtaining a binary image, separate objects, which in the considered case represent particles, need to be detected. In the simplest case, particles do not touch or overlap and can be separated by labeling the connected components in the image. A connected component is a set of pixels where each pixel can be reached from each other pixel of the component by following a path consisting of adjacent pixels [81, p. 469]. The results therefore depend on the type of adjacency used, where the adjacency defines the neighbors of a pixel. In 2D images, one can consider a 4-neighborhood, where the neighbors of a pixel are found above, below, left and right, or the 8-adjacency which additionally includes the four diagonal neighbors. Similar relationships are defined for 3D images. 6-neighborhood consists of the six voxels whose centers are at the distance of 1 voxel from the current voxel. 26-neighborhood comprises all diagonal neighbors as well, so that the possible distances between the current voxel and the neighbors are 1, $\sqrt{2}$ or $\sqrt{3}$ voxels. The term "labeling" refers to the fact that the procedure creates a gray value image where all pixels of one connected component are given the same gray value (label).

Watershed Transform A labeling procedure will identify a group of touching particles as one connected component, which may lead to errors in size and shape computation [23]. In case of grown-in agglomerated crystals, it is not possible to separate the primary particles by an adequate dispersion procedure and further techniques must be employed.

Watershed transform is a classical image processing procedure explained in detail in the literature

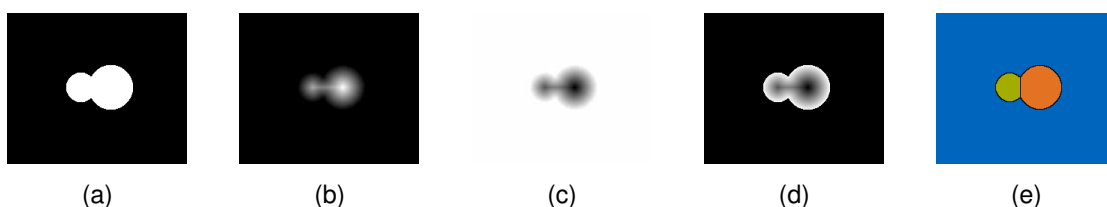


Figure 2.5: Separating touching objects by a combination of watershed transform and distance transform. (a) Original image. (b) Distance transform of the inverted image. (c) Sign of distance transform changed. (d) Background set as basin. (e) Watershed transform: obtained regions illustrated by different colors, whereas watershed pixels are shown in black. Note that distance transform images were adapted for illustration purposes.

[81, p. 542-550][89]. It is performed on a gray value image, where gray values can be thought of as representing height values in a landscape. An example can be seen in Figure 2.5d. If the landscape is flooded, several catchment basins, also termed regions, will form and will be separated by the watersheds. Watersheds represent landscape points that belong to neither catchment basin as water falling onto them is equally likely to flow to either of the neighboring basins [81, p. 542]. In this analogy, the obtained basins can be thought of as separate objects and the watersheds as boundaries between them. This is illustrated in Figure 2.5e, where watershed pixels are shown in black. Catchment basins or objects will therefore be created around the local minima [88, p. 143]. Watershed by immersion [89] is an algorithm that computes the watershed transform by simulating the flooding process. The process can be illustrated by imagining that the image is "pierced" at the positions of local minima and is then immersed into a pool of water. As the water flows into the catchment basins the watersheds are built to separate them. The algorithm [89] therefore starts by assigning basins to the pixels with the lowest gray value. Pixels that have the next highest gray value are then considered and can either be added to the existing basin, assigned as new basins or left out of consideration if they belong to a connected component that contains more than one existing basin. The procedure is iterated up to the highest gray value level, resulting in a set of labels for different catchment basins and a label with all watershed pixels corresponding to those that were not assigned to any basin. The concept holds for 2D as well as n-dimensional images [89].

In order to separate touching or overlapping objects, watershed segmentation can be used together with the distance transform [81, 88, 89]. One of the possible procedures is illustrated in Figure 2.5. For each pixel, distance transform gives the shortest distance to a foreground pixel [81, p. 543]. If the image is inverted beforehand, this method will compute the shortest distance to the background for each pixel in the foreground. Therefore, the obtained gray value image will often have high values in the center of each of the touching objects. Changing the sign of these values so that they become negative will prepare the image for the watershed transform, ensuring minima in the object center. In order to enforce watershed pixels along the boundary between background and foreground, background should be assigned the value $-\infty$. If this step is skipped, the obtained regions would extend into the background. In this case, correct foreground regions can be obtained by masking the watershed image with the original binary image [84].

A further segmentation technique involves using watershed transform on the image gradient, so that high gray values are obtained near object edges, thus forcing watersheds pixels at these locations [81, p. 545].

Watershed transform finds application for separating touching particles [84, 90] or biological cells and nuclei [91, 92, 93, 94, 95]. A common problem is over-segmentation where the desired object is split into several parts [81, 84, 88]. Under-segmentation may also occur when splitting overlapping objects [94]. Over-segmentation can be resolved by pre-processing the image, modifying the watershed transform algorithm or merging the obtained regions [88, p. 144].

Pre-processing involves modifying the image before applying the watershed segmentation. An example is the H-minima transform [84, 88, p. 111] which attempts to resolve the issue of too many local

minima. The H-minima technique is an iterative procedure, introduced here based on [81, p. 472,484-485] and [88, p. 111]. In each iteration, the pixel value is replaced by the maximal value in its neighborhood, where the neighborhood is defined by the connectivity. This is also known as grayscale dilation and is described in section 2.3.2.3. The obtained image is then compared with a mask image and the minimum of the pixel values in two images at each location is taken. The mask image is obtained by increasing the gray values of the original image by a parameter h , called the "dynamic" [88, p. 111]. The procedure ends if there is no change of the image in successive iterations. The effect of the operation is that some local minima are deleted as several local minima may be joined into a common valley. An analogous operation on local maxima is known as the H-maxima transform [96, p. 63]. Another pre-processing approach [90] consists of considering a sphere around each local minimum with a radius equal to the distance of that local minimum to the background. Any other local minima within such a sphere are deleted. This method was designed for splitting touching spherical particles [90].

A further pre-processing technique for eliminating over-segmentation involves using watershed markers or seeds in form of a mask image that contains foreground pixels at the locations that are known to belong to the watershed basins or to the background [81, p. 547]. The mask image can be used to enforce minima on these desired locations in the image on which the watershed transform is performed, such as the gradient or the distance transform image [81, p. 549]. Various methods can be used to obtain watershed markers and are often designed with the specific application in mind. Zhang et al. [94] defined markers as local minima in a special distance transform image obtained after performing a procedure that shrinks the overlapping objects near the points where they overlap.

Finally, over-segmentation may be resolved by merging watershed basins. Umesh Adiga and Chaudhuri [91] merged small with large watershed regions whereas Long et al. [93], apart from size, also considered the gray values and the convexity of the watershed region obtained upon merging.

Edge Detection In the here discussed 2D applications, it is typically sufficient to find the outer boundary of the particle. This problem statement is known as edge detection. The algorithms are based on the fact that gray values exhibit a strong change at the edges so that the gradient image will have high gray values at these locations. However, using a simple filter that computes the image gradient and thresholding the resulting image usually does not provide satisfactory results [86, p. 103]. A popular algorithm is the Canny edge detector, explained in detail in [86, p. 103-105] and summarized here. The original gray value image is first filtered with a Gaussian filter whose standard deviation is some chosen value σ . The gradient of this smoothed image, as well as the gradient magnitude, are computed. The algorithm then finds pixels that represent local maxima in the gradient magnitude image along the directions specified by the gradient. Edges are formed in the final step by performing hysteresis thresholding on the image consisting of these local maxima. The Canny edge detector algorithm therefore has three parameters, σ and two thresholds for the hysteresis operation [81, p. 500]. Calderon De Anda et al. [20] used a two-scale method combining the edges detected by two Canny operations with different σ parameters. This edge detection operation was the basis for segmenting in-situ images of crystals taken by a camera positioned near the reactor wall. The method used by other researchers segmenting

in-situ images of crystals [61, 71, 97] and was integrated into the Stereovision^{NI} software which is used to process images obtained by a stereoscopic setup [23].

2.3.2.3 Morphological Operations

The segmented objects often need to be processed further before size and shape measurements can be undertaken. Classical tools for this task are morphological operations, described in detail in [81]. The two basic operations are morphological erosion and dilation. The operations can be applied both on binary [81, p. 444-447, 451-454] and grayscale [81, p. 475-478] images. In both operations, a mask, also called structuring element, is centered at each pixel of the image. In case of binary erosion, the foreground pixel is kept only if all pixels marked by the structuring element are foreground pixels. Grayscale erosion, in the simplest case, sets the pixel value to the minimum of the pixel values in the mask, so that binary erosion represents its special case. Similarly, grayscale dilation changes the pixel value to the maximum of the values seen in the mask, where the mask is reflected around the origin beforehand. Binary dilation is a special case where the pixel is set to foreground if at least one foreground pixel is covered by the structuring element. Therefore, binary erosion shrinks the object whereas binary dilation expands it. In order to change the object size as little as possible, combinations can be applied [81, p. 454-457]. Morphological opening is an erosion followed by a dilation, whereas a closing is a dilation followed by an erosion. Binary closing is often applied to close discontinuous edges that may result from uneven illumination or blurring, such as in [20]. Binary opening can be used to reduce contour roughness or to delete small or thin objects that represent noise, also seen for example in [20]. Grayscale variants of the operations can be used to filter the image or correct uneven illumination [81, p. 480-482].

A further morphological operation is hole-filling, where enclosed background pixels are set to foreground [81, p. 474].

2.3.3 Image Analysis

Before starting image analysis, it is assumed that separate particles have been identified. The goal is now to measure their size or fit and measure the shape. This section provides an overview of measurable quantities as well as of methods and algorithms.

2.3.3.1 Object Size

The most intuitive size measure for an imaged object is the object area in case of binary 2D images or the object volume in case of binary 3D images. Area of a 2D object can be determined by counting the foreground pixels [98, p. 34] and multiplying by the area of each pixel expressed in μm^2 . Volume of a 3D object can be computed in a similar manner. Based on these values, one can calculate the equivalent diameter of a circle with the same area (2D) or a sphere with the same volume (3D). Area is

the zeroth-order moment, $m_{0,0}$ of a 2D object, where the moment of order p, q is defined as [98, p. 37]:

$$m_{p,q} = \sum_{(u,v)} u^p v^q, \quad (2.8)$$

where u and v are the coordinates of foreground pixels. The moments $m_{1,0}$ and $m_{0,1}$ represent the coordinates of the centroid and can be used to compute central moments [98, p. 38]. The object can then be approximated by an ellipse with the same central moments as the object [98, p. 43]. A similar idea was used by the group of Mazzotti, who described the imaged particle by the major and minor axis length of an equivalent ellipse [61, 72, 99, 100]. Similarly, in 3D, long, intermediate and short axis length of a best-fit ellipsoid can be used [83]. Length and width can further be approximated by the maximal and minimal Feret diameter [18] or by using morphological operations [17]. In 3D images, length of a particle can furthermore be defined as the length of the longest straight line connecting points on the surface [82]. Width and thickness are defined similarly, while being mutually perpendicular and perpendicular to the length direction [82].

A further commonly used measure of particle size is the perimeter. In order to compute the perimeter, one must first find the boundary of the object, defined by the foreground pixels that have background pixels as neighbors [81, p. 552]. Note that the obtained boundary depends on the choice of connectivity. In order to calculate the perimeter, therefore, one must determine the length of the line that connects the boundary pixel centers. The so obtained length can overestimate the real perimeter, even for a well-chosen connectivity, so that a scaling factor of 0.95 can be used to correct the result [98, p. 33]. Đuriš et al. [101] have investigated three different softwares and have shown that different algorithms are used for perimeter calculation. The difference in the obtained perimeter measures further influencing shape factors that are computed using the perimeter. Moreover, they show that image resolution has a strong influence on the calculation of the perimeter.

Imaging with a higher resolution reduces the discretization error. Objects containing very few pixels should be left out of consideration as they cannot be differentiated from the image noise and the error in their size measures is large. The resolution and the measurement system should therefore be chosen in such a way that the discarded particles represent a neglectable portion of the particle size distribution. A brief discussion on the topic of image resolution can be found in Gamble et al. [12].

2.3.3.2 Shape Descriptors

Shape Factors Shape factors are values derived from the size measures. Classical shape factors are circularity, defined as $\frac{4\pi A}{P^2}$ [98, p. 34], where A and P are the particle area and perimeter, respectively, or the aspect ratio and elongation, representing the ratio of some chosen length and width measures. Certain crystal shapes are more desirable than others so one or more shape factors can be used to monitor [17] or control the crystallization process [24].

A large number of shape factors, combined with other shape descriptors and size measures, can be used as input for pattern recognition tools that recognize the amount and degree of agglomeration

or the polymorphic form of crystals. This is discussed in more detail in section 2.3.4.3.

Fourier Descriptors In case of 2D images, particle shape can be described by a curve showing the distance of contour points from the object centroid. In case of a perfect circle, this curve would be flat, whereas a curve for a square would show four peaks. The curve can be approximated or smoothed by using Fourier transform and considering only a chosen number of Fourier coefficients [81, p. 582]. Furthermore, if the curve is appropriately sampled at a constant number of points, this approach can be used to provide a signature and compare different objects. Considering only the magnitude of the Fourier coefficients further ensures independence of object rotation, while the coefficients can also be scaled to discard the information about object size [25].

Borchert et al. [25] created a pre-computed database of Fourier descriptors to describe the shape of KDP crystals. To obtain the database, they simulated crystals with different orientations and aspect ratios and projected them onto a plane. The database was used to measure growth rates in an experiment where crystals were imaged by the QICPIC dynamic image acquisition method. Imaging were taken on-line in a flow-cell through which samples of the suspension were periodically pumped. The descriptors of imaged crystals are compared with the database. The obtained match, upon re-scaling the size, provides the full 3D reconstruction of the imaged crystal.

A similar idea, involving comparison with a database of Fourier descriptors obtained from different orientations of the crystal towards the projection plane had also been pursued by Li et al. [102]. It was used to discriminate between the two polymorphs of L-glutamic acid in images obtained on-line.

Concavity Descriptors In certain crystallization applications it is necessary to discriminate between single crystals and agglomerates or recognize primary particles in aggregates. In case of a binary image, this can be done based on object convexity, as primary particles would appear convex while aggregates would exhibit concavities. Most concavity descriptors are based on the convex hull concept. A convex hull of a region is the smallest polygon that is convex and contains all points of that region [98, p. 35]. Therefore, the convex hull of a set of pixels can be computed exactly by considering each pixel as a filled square. However, the image itself is a discretized version of the true object area, so that the convex hull should be discretized as well to yield a so-called convex image. The solidity is then computed by considering the areas of these two images [81, p. 597]. Another approximation of the convex hull is obtained by approximating pixels through their center points.

The simplest concavity-based descriptor is the convexity or solidity, defined as the ratio of the actual object area and the area of its convex hull [69, 103]. Concavity index is defined as the ratio of the largest concavity segment area and the object area [69]. The largest concavity segment is the largest connected component in the convex hull image obtained after subtracting the object itself [69]. Concavity-based descriptors can also be defined using morphological operations [104, 105, 106].

2.3.3.3 Shape Fitting

If the combination of the imaging method and the particle shape permits it, the ultimate goal of the image analysis procedure is to obtain the parameters of the polytope that best fits the imaged crystal. In case of 2D imaging, it is sometimes sufficient to fit an appropriate 2D shape on the recognized particle, whereas for simple shapes a full 3D reconstruction can be achieved. In either case, shape fitting requires detecting lines and corners, in case of 2D images, or planes, in case of 3D images.

Fitting 2D shapes Shape fitting in case of 2D images typically starts by line detection. There are several methods in the literature. Hough transform [32, 107] considers a set of directions \mathbf{d}_i in space and a set of distances ρ_j . Each direction-distance pair (\mathbf{d}_i, ρ_j) defines one bin. Bins are then filled with points \mathbf{p} that fulfill

$$\rho_j - \frac{\Delta\rho}{2} \leq \langle \mathbf{d}_i, \mathbf{p} \rangle < \rho_j + \frac{\Delta\rho}{2}, \quad (2.9)$$

where $\Delta\rho = \rho_j - \rho_{j-1}$ is the step size for the distance discretization. Lines are then found by a suitable maxima search over the space defined by the bins. Reinhold et al. [32] used Hough transform as a basis of an algorithm that fits a polytope to each of the 2D images obtained by a stereoscopic setup. Then, Minkowski-addition principles are adopted in order to fit a rounded shape and find the roundness of the abraded potash alum particles. An alternative method to Hough transform is the Burns line finder, used by Larsen et al. [21]. Burns line finder is applied to grayscale images and produces line segments which are further concatenated into lines. Pairs of parallel lines are then identified and clustered together. Based on these clusters a rectangle is fitted and used to estimate the particle length, width and orientation. The approach was applied to in-situ crystal images and allows detected rectangles to overlap. It was further extended by incorporating a 3D crystal model [22]. Here, the detected lines are clustered into viewpoint invariant groups (VIGs), meaning that their properties are kept regardless of the orientation of the particle towards the camera. VIGs are then compared to the possibilities arising from the crystal model in order to estimate both the model and viewpoint parameters and finally fit a 3D shape. The success of both methods [21, 22] was found to be dependent on the solids concentration, which is an inherent problem of the in-situ imaging methods, as discussed above. Ahmad et al. [97, 108] also fitted rectangles and polygons on overlapping particles in in-situ images. The method is based on detecting "salient" corners, representing corners where exactly two edges intersect. Groups of three corners are then formed and symmetry is used to compute the remaining corners, depending on the assumed polygonal shape. Length and width can then be computed based on these shapes to yield the particle size distribution [108].

It is furthermore possible to fit an ellipse based on the moments of the object image, as explained in section 2.3.3.1 and implemented by the group of Mazzotti to measure the object length and width [61, 72, 99, 100]. Note that in this case, it is not necessary to detect lines in the image, rendering the computation simple and efficient.

Fitting 3D shapes As discussed in section 2.3.1.2, 3D crystal shape can reliably be measured by 3D imaging methods. Singh et al. [19] used confocal microscopy to image crystals which were coated with fluorescent dye. The shape reconstruction algorithm extracts the contours of the 2D slices and determines the convex hull of the obtained stack of contour points. In order to determine crystal faces, convex hull points are triangulated. Triangles with similar orientation are grouped together and boundary points close to the triangle groups are assigned to them. Planes are fitted through these points and the face distances are determined, providing a 3D polytope. In order to determine the crystal polymorph, angular patterns of the obtained polytope are computed and compared with the theoretical angular patterns of different polymorphs. Finally, Miller indices are assigned to faces.

Full 3D shape can in some cases also be extracted from 2D images. In section 2.3.3.2, it was stated that a full 3D shape of a crystal can be obtained by comparing the Fourier descriptors of a particle with those in a pre-computed database [25]. That approach [25, 102] assumes that the particles are randomly oriented towards the imaging plane as the imaging is done on-line or in-situ. In case of off-line imaging, particles settle in a stable position under the influence of gravity, enabling in certain cases a direct fit of the 3D shape. Ma et al. [71] investigated potash alum crystals grown in a hot-stage reactor. The morphology of the crystals is such that crystals settle on one of the faces belonging to the largest face group. The distances of certain line pairs are related to the 3D crystal shape and its projection on a plane from the assumed angle. Therefore, the 3D shape can be obtained from some geometrical considerations. It is used to obtain the face-specific growth rates.

Several methods for obtaining the 3D shape from stereoscopic imaging setups have been developed. Apart from the presented image processing steps, stereoscopic imaging requires a matching step where the desired features from the images are paired [44]. Upon establishing the matching, further image analysis steps can be performed. The group of Mazzotti [26, 27, 100, 109] matched labeled objects in the two images based on the coordinates of their centroids and bounding boxes. Images were taken in a flow-cell with a stereo angle of 90° , where on-line operation is possible by continuously sampling the slurry and returning the samples into the reactor after imaging [26, 27, 109]. Different levels of detail can then be extracted from these images. In [26, 27], particles were approximated by spheres, cylinders or cuboids. Particle class was determined based on the curve that describes the distance of the boundary pixels to the centroid by analyzing the extrema of this curve. The length measures of the approximated shapes are then obtained from the boundary curve or by analyzing the extreme pixels in both projections. In their future work [109], a shape model corresponding to a polytope based on the crystal morphology was adopted. The polytope is measured by solving an optimization problem that considers possible orientations and shapes of the particle in the imaging setup. In each step, the outlines of the particle in the two stereoscopic images are compared to the outlines that would be obtained when projecting the particle with the currently considered orientation and shape. As each considered particle type can crystallize into shapes with several different combinations of edges, different regions of the morphology domain are considered separately. The approach was validated by simulation of different particle morphologies, showing that method is capable of measuring the crystal polytope with a reasonable accuracy in most cases. A simulation of different stereo angles ($12^\circ - 90^\circ$) shows that a

polytope that matches the given pair of projections is found for the whole range of angles. However, when comparing the actual simulated size and shape with the result of the optimization procedure, large errors were found for stereo angles lower than 60° . Furthermore, the authors see little advantage with respect to the shape approximation approach [27]. The particles observed in a real experiment were of too poor a quality, thus not fitting the polytope model perfectly, whereas a simulation study has shown that the generic particles can also capture the volume and aspect ratio well at a much lower computational effort.

The stereoscopic setup proposed and investigated by Wang and co-authors [75] can be used to monitor on-line static particles grown in a flow-cell [77], as well as particles grown in a reactor by placing the cameras next to the vessel wall [23, 70, 76]. The advantage of the latter setup with respect to that of the Mazzotti group is that the sampling step is eliminated [23]. A description of the reconstruction steps can be found in [76]. The matching of crystal pairs in the two images is done based on the centroid coordinates. Then, corners and edges for each pair of particles are matched. The reconstruction of the 3D coordinates uses the triangulation method. The basic idea of this method is that lines connecting an imaged point to the camera for the two images will intersect in the point which was imaged, thus providing its 3D coordinates [110, p. 263]. In [23], the setup was used to measure the 3D length of needle-like particles and estimate the growth rate in length direction. The length measurement was validated by comparison to the result of a static off-line imaging setup. The applicability to other crystal morphologies and to face-specific growth rate measurement was shown in [76]. The setup was further applied [77] to measure the face-specific growth rates of static crystals in a flow-cell and investigate the influence of supersaturation, solution velocity and seed size.

Bujak and Bottlinger [111] designed a method for reconstructing the 3D shape of irregular particles based on three orthogonal projection images taken during particle fall. As no shape model for the irregular particles can be adopted, the reconstruction is done purely based on the contours extracted from the projection images. Two of the three contours are taken to form a simple wire-frame, while the third contour is approximated by a polygon. Then, the wire-frame is intersected by planes parallel to the third contour. Each plane intersects the wire-frame in four points that define a rectangle. The approximated polygon is then scaled within this rectangle in order to provide one slice of the particle. The reconstructed shape consists of a stack of such slices and approximates the particle volume well.

Chakraborty et al. [112] presented a method for reconstructing the 3D shape of a single crystal from several images taken from different angles. The crystal was imaged by an ordinary digital camera coupled with a light source. The idea is based on the fact that regular light reflection occurs from exactly one crystal face if the crystal is imaged from a certain angle, allowing the shape of that face to be easily extracted by suitable image processing steps. The 3D reconstruction is then performed by subsequently attaching reconstructed faces using the information about the angular orientation of the face and its adjacency to other faces over edges. The technique is not easily applicable for crystal populations. It is time consuming as the camera must be re-positioned to take images from different angles and the information about camera movement and face adjacency must be noted and specified for reconstruction.

2.3.4 Statistical Evaluation

The image analysis step results in the chosen size or shape measures for each imaged particle. In certain cases, the result is a full 3D reconstruction of the imaged particle, so that any size or shape measure can be computed. The task of the statistical evaluation step is to process this data for a large amount of particles to obtain the final information of interest. To this end, size and shape data is often correlated to other measured quantities or simulation results. This section presents a brief overview of the possibilities in this area.

2.3.4.1 Measurement of Particle Size and Shape Distribution

We here consider particle size distribution (PSD) to be a one-dimensional distribution determining the amount of particles for a given size measure. A particle size and shape distribution (PSSD) is a multi-dimensional distribution over more than one size measure. In the ideal case, a PSSD has as many dimensions as there are face groups for the considered crystal and it is defined over the face distances h_i . As these values are rarely experimentally measurable, PSSD can be reported in terms of length and width [17, 27, 58, 69], or one length measure and two aspect ratios [100].

In the simplest case, the PSD/PSSD is obtained by gathering the measured quantities into a histogram and scaling them appropriately. Puel et al. [17] thus obtained a 2D distribution over length and width, where these quantities were measured by static image analysis. Measuring length of needle-shaped particles by dynamic image analysis, however, results in smaller values due to particles' random orientation towards the camera [23]. Particle width is affected similarly. Therefore, even though one can create a distribution of so obtained length and width values, they do not correspond to the true PSD. This issue can be overcome by modeling the imaging process, as shown by the group of Mazzotti [61, 72, 99, 100]. They chose to fit ellipses to images of particles taken in a flow-cell setup, yielding major and minor axis length measures. A relationship between the obtained axis length distribution (ALD) and the true PSD was then established. A Monte Carlo approach was adopted to compute the ALD of a single particle and weighing was employed to find the ALD of the entire distribution [61]. The inverse problem of extracting the PSD from the observed ALD is solved by optimization [72]. The approach was also applied to the stereoscopic setup, where a 4D ALD was related to a 3D PSD [100]. This idea is similar to extracting the PSD from the chord length distribution [57, 59, 113], as discussed above.

2.3.4.2 Kinetics Measurement

Measuring the kinetics of a certain process is often a first step towards its optimization. In crystallization, a lot of effort is invested in measuring the growth and nucleation kinetics. Studying the nucleation kinetics typically requires the measurement of the induction time and therefore the detection of very small particles. While this can be done by detecting the particles through FBRM [114] or optical microscopy in case of small volumes [115], it typically involves measuring some macroscopic property of the whole slurry, such as for example the turbidity [65, 115]. Therefore, the focus of this section is on

the growth rate measurement. Dissolution rate measurements can be obtained similarly, as they can be considered to represent negative growth rates.

Growth rate measurement is based on tracking the change of crystal size over time. Different size measures can be used, ranging from the sphere equivalent diameter to face distances from the crystal origin, corresponding to the H -representation [25, 71]. Underlying kinetic models also differ, ranging from a linear equation describing the change of mean particle length over time, as for example used in [23], to models that incorporate the dependence on the supersaturation. The latter methods, therefore, require a concentration measurement and usually employ on-line infrared spectroscopy probes. Furthermore, the methods can be employed both to study a single crystal and the whole population. The growth rate estimation can involve either a deterministic curve-fitting of the desired parameters, or numerically solving an optimization problem. The first case is employed when the underlying equations are simple enough that an explicit solution for the growth parameters can be derived by setting the gradient of the objective function to zero. The latter approach is non-deterministic and the solution depends on the chosen starting point as the typically-used gradient-based algorithms are only capable of finding a local minimum of the objective function. The objective function relates the measured crystal size to the crystal size obtained under the chosen set of growth parameters. It can thus be defined using the mean of the measured distributions or by using the entire distribution. Therefore, there is no single approach for the growth rate measurement and the chosen approach depends on many different factors. In the following, three examples are presented, based on different information about crystal size and shape extracted from the imaging and image analysis procedures described above.

Ma et al. [71] conducted hot-stage experiments where a small amount of crystals nucleated and were then allowed to grow. As discussed above, it was possible to extract the full 3D shape from the obtained microscopy images as the orientation of the crystals towards the camera is known. The concentration, and thus supersaturation, are determined by measuring the volume of the crystals through image analysis. A second order polynomial was fitted to the curve describing average face distances of different face groups over time, so that a linear growth rate can be computed as its derivative. These growth rates can now be correlated with the supersaturation σ which changes over time to obtain equations of type

$$G_i = k_{g,i} \sigma^{g_i}, \quad (2.10)$$

where i denotes different face groups, whereas $k_{g,i}$ and g_i are fit parameters.

Borchert et al. [25] obtained the full 3D crystal shapes from the 2D on-line imaging setup by comparison of the Fourier descriptors of the measured particle with a pre-computed database. They conducted seeded cooling crystallization experiments with two different constant cooling rates, yielding two different supersaturation profiles. A PSSD was estimated from the 3D reconstructions of the particles and the mean shape could be computed at different points in time. The growth rate law from equation 2.10 was assumed for each of the two face groups i . The parameters $k_{g,i}$ and g_i were found by solving the optimization problem with two least-squares objective functions summing the errors of the average face distance h_i for each face group at different times. This work was further extended by Eisenschmidt et

al. [116], who incorporating the dependence on temperature, considered the effect of impurities and also measured dissolution rates.

Ochsenbein et al. [45] monitored the seeded batch desupersaturation experiments using the stereoscopic setup of the Mazzotti group [27] to obtain the 2D PSSD. The needle-like particles were approximated as cylinders where the length and diameter were measured. Several growth rate expressions were investigated, with and without size dependence, both mechanistic and purely empirical. The objective function for the optimization problem is defined over the concentration and moments of the PSSD. A population balance equation is solved in order to get the concentration and moments for each chosen set of parameters. The method was validated by simulating the imaging process where particles are described as true 3D polytopes as well as by additional experiments. A discussion of different growth expressions and potential causes of the observed PSSD broadening were also provided.

Alternatively to using image analysis, growth rate measurement can be done by model identification where an optimization problem is solved to obtain the growth parameters. The optimization function can compare the initial and final particle sizes, but it can also be written over quantities other than the crystal size. For example, Kim and Yang [65] assumed that the nucleation will occur at the same supersaturation level for a certain cooling rate. The objective function used to optimize the growth rate parameter for different face directions was therefore based on the difference between the metastable zone width and the supersaturation at the point of nucleation, measured for different seed amounts.

2.3.4.3 Particle Classification

Particle images can be used to classify particles according to criteria such as shape [24, 117], polymorphic form [70, 118], whether the particle is an aggregate [68, 69] or the complexity of an aggregate regarding the number of underlying primary particles [105, 106]. Based on this classification, other properties of the particulate product can be computed, such as the agglomeration degree or the amount of different polymorphs. In the following, necessary tools for particle classification are presented and their properties are discussed before giving particle classification examples from literature.

Classification is a classical problem in the field of pattern recognition. It considers a set of objects,

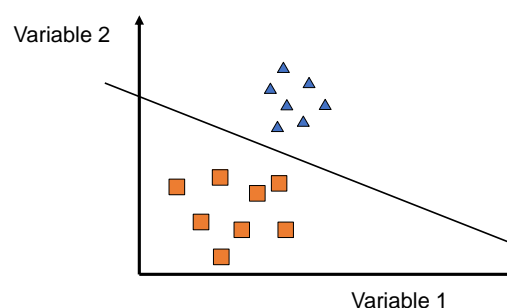


Figure 2.6: The task of classification is to determine the class (here, triangle or square) of an object based on the value of the considered variables. In this case, classes are linearly separable, as depicted by the line.

each characterized by some variables, as illustrated in Figure 2.6. In case of particle imaging, the objects are the imaged particles, whereas the variables are typically the computed size and shape descriptors, together termed image descriptors. The goal of classification is to sort each object into one of the classes based on the measured variables. The classes are typically pre-defined by a human operator, so that in case of particle imaging, the operator may define the classes as single crystals and aggregates [69] or circular, square and rectangular and irregular shapes [117]. A set of mathematical operations that result in a decision regarding the class of a given object is called a classifier. It is inferred from a training set containing objects that were manually classified by a human expert. The accuracy of the classifier can be determined using another set of manually classified objects known as the test set. Finally, the classifier can be applied to new objects for which the underlying class is unknown. The described approach is known as supervised learning [119, p. 3] in the pattern recognition literature. In the case of unsupervised learning, no manual classification is performed and the underlying classes are unknown. The resulting procedure is called clustering as the algorithm searches for clusters of similar objects [119, p. 3].

In case of particle imaging, determining whether a particle is an aggregate by thresholding its solidity value can be considered as a trivial classification problem with one variable. In case of two variables, a simple classifier can be visualized as a curve that splits the 2D variable space into two parts belonging to the two classes. A generalization to n dimensions and N classes is obtained by considering a set of surfaces that partition the n -dimensional space into at least N parts. The goal of the classification procedure, therefore, is to fit such curves or surfaces. If lines or hyperplanes are obtained, the classifier is called linear, otherwise it is known as non-linear. Classifiers can furthermore be considered as parametric or non-parametric, depending on whether they fit the parameters of some assumed probability distributions that describe different classes.

Discriminant factorial analysis (DFA) is an algorithm for obtaining a linear discriminant function, which is a linear parametric classifier. It is assumed that each class is normally distributed over the variable space, where the classes have different means but the same covariance matrices. The classifier finds the parameters of these normal distributions [120, p. 343] and classifies each new object in a class where the probability of misclassification is minimized. This is equivalent to maximizing the posterior probability [120, p. 336, 394]. It is furthermore possible to form a quadratic classifier by relaxing the assumption that the covariance matrix of the two classes is the same, leading to the quadratic discriminant function [120, p. 340]. However, such classifiers are less stable and often more sensitive to the size of the training set [120, p. 347].

Artificial neural network (ANN) represents a non-linear, non-parametric method. A feed-forward neural network, also known as multilayer perceptron, consists of interconnected processing nodes, also called neurons, typically divided into three or more layers. More information about multilayer perceptrons can be found in the book by Bishop [119, p. 225-261], while the most important aspects are introduced here. An illustration of such a neural network is given in Figure 2.7. The input layer contains one node for each of the considered variables. Each node in each of the hidden layers performs a non-linear, typically sigmoidal, function over the weighted sum of the outputs of the nodes from the

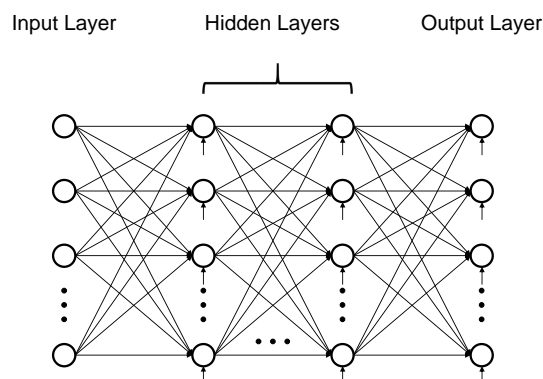


Figure 2.7: An illustration of a neural network with an input layer, several hidden layer and an output layer.

previous layer. Each node in the output layer corresponds to one class and the object is sorted into the class corresponding to the output node with the highest value. The training process consists of finding the weights that minimize an objective function describing the difference between the obtained and the desired output values for the objects in the training set. Therefore, training an artificial neural network is an optimization problem that can be solved by classical optimization techniques such as steepest descent and other gradient-based methods. These methods modify the value of each parameter, in this case each weight within the network, in the direction of the negative derivative of the objective function with respect to that parameter. The objective function, however, depends directly only on the weights that connect the last output layer to the hidden layer. Its derivative with respect to other weights can be derived mathematically and contains an expression for the output error of the upcoming layer. Therefore, in order to modify the weights of inner layers, the error is propagated back from the output layer towards the input layer. This is known as the backpropagation principle and was derived for the least-square objective function by Rumelhart et al. [121].

The classification accuracy depends on the chosen pattern recognition tool, such as DFA or ANN, whereas the ANN introduces further degrees of freedom in terms of the network structure, such as the number of neurons and hidden layers as well as the implementation of the optimization procedure [122, 123]. The quality of the obtained classifier furthermore depends on the training set and therefore on the amount of objects it contains [105, 122, 124], the relative amounts of objects from different classes [124] and the chosen variables [68, 105, 123]. Increasing the amount of objects in the training set improves the classification accuracy but also increases the costs for creating the classifier as it implies more measurements and more time invested in manual object classification. If the measurements are costly, one can attempt to compensate for a small amount of objects in the training set by simultaneously measuring a large amount of variables. In case of particle classification by image processing this can easily be achieved by using as variables a large number of Fourier descriptors or all image descriptors computed by the used image processing software. However, this strategy does not always give the desired results and can lead to the so-called peaking phenomenon. Peaking phenomenon is viewed as the decrease of classification accuracy with an increasing number of considered variables [122,

125]. The explanation is based on the fact that certain variables contribute more information to the classification than others as different classes exhibit different values for these variables. Therefore, a large amount of useless variables can introduce unnecessary noise into the classification process [126]. Hamamoto et al. [126] showed that ANN classifiers are more robust to peaking phenomenon than some other investigated classifiers in case where the number of used variables increases linearly with the number of training samples. This is explained by the capability of ANN to ignore useless variables. However, ANN can still exhibit peaking behavior [122]. In case of particle classification with DFA [68], it was observed that quality of the obtained classification depends on the chosen set of image descriptors, where a larger set does not necessarily lead to better results. In such cases, a significant amount of effort is invested in selecting an appropriate subset of variables that leads to the best classification accuracy.

Automatic particle classification has been implemented by many research groups. A wide variety of image acquisition setups, classification methods, classes and variables were considered. Image acquisition can be done off-line statically [68, 105, 117, 127] or dynamically in suspension [74], as well as on-line using a flow-cell setup [69, 106]. An overview over the investigated particle properties, variables and classification methods is given in Table 2.1. Particles can be classified either into single crystals and aggregates [69], additionally including waste particles [68, 74], or into several complexity classes based on the appearance and the amount of primary particles in an aggregate [39, 40, 105, 106]. Further possible classifications are done based on shape [24, 117] where the different shapes can also represent different polymorphic forms [70]. The used variables most often involved a set of size-independent image descriptors [39, 69, 105, 106, 123], typically involving classical shape descriptors and concavity descriptors. Terdenge et al. [68, 74] additionally used size measures. Some researchers employed Fourier descriptors, defined in multiple manners [117, 127, 128], where Fourier descriptors were shown to be slightly inferior to shape and concavity ones [117]. The most popular classification methods are DFA [68, 74, 105, 106, 127] and ANN [117, 123, 128]. Ochsenbein et al. [69] and Huo et al. [70] employed a non-linear classifier called support vector machine (SVM). Some of the discussed approaches first performed a dimension reduction on the selected variables [39, 40, 70, 118, 123]. A popular tool is principal component analysis (PCA). In [39], a function of principal components that is correlated to the number of primary particles was derived through the visual observation of the PCA score plot for the training set. This represents a scatter plot of the image descriptor values in the principal component space. A correlation function was fitted and used to estimate the number of primary particles for the data in a test set as well as the unseen data. An even simpler classification procedure was designed by Patience and Rawlings [24]. They considered two variables, plotted the particles in this variable space and fitted two non-intersecting ellipses to clusters of the two classes, representing tetrahedral and cubic particles. Particles that fall outside of these ellipses were later discarded. Apart from the mentioned supervised learning approaches, Zhang et al. [118] used an unsupervised clustering method to detect crystal polymorphs.

Finally, the designed classifiers can be used to extract the information about the crystal product or the underlying crystallization process. The amount of agglomerates [68, 69, 74] can be used to

characterize the product after washing and drying steps [74] or extract information necessary to model agglomeration [36, 69]. An agglomeration degree on the particle-level, describing the amount of primary particles an aggregate consists of, can be used to study the effect of different impurities on particle morphology [105, 106]. Finally, the classifications of particle shape can be used to design process control strategies with the goal of obtaining the desired particle shape [24].

2.4 Orientation Measurement

In the previous section, methods for extracting particle size and shape were discussed, with a focus on imaging techniques. The general approach was divided into four steps presented in Figure 2.2, involving image acquisition, processing, analysis and statistical evaluation. These methods enable a detailed study of single crystals, as their full size and shape information can be obtained. However, there are still limitations with respect to crystal aggregates and their structure. Information about the primary particles that form the crystal aggregates are contained in crystal images, but are difficult to extract. The statistical methods discussed in section 2.3.4.3 enable classifying aggregates based on their complexity [105, 106, 127] and estimating the amount of primary particles [39, 40], the obtained values are only an estimate and no information about primary particle orientation is gained. Any procedure for measuring orientation between primary particles requires segmenting an aggregate into the constituting particles, which is not a trivial task. Furthermore, primary particles overlap and are not completely visible in the image which can cause ambiguities in the shape reconstruction. In case of static 2D imaging, no assumptions about the most stable position under the influence of gravity can be made as this depends on the structure of primary particles. In case of dynamic 2D imaging, even if resolving overlapping particles is in some cases possible [21, 97, 108], one would have to distinguish between separate single crystals that overlap in the projection image only and true aggregates. 3D imaging methods presented in this work [1, 2], as well as stereoscopic techniques [69] enable this distinction and may represent promising tools to study these phenomena.

The only currently used method for studying particle orientation based on imaging, excluding the work presented in this thesis and the related publications [1, 2, 3], involves visual observation. Collier and co-authors [34, 42] studied calcite particles using scanning electron microscopy (SEM) and transmission electron microscopy (TEM). Visual observation of the produced SEM and TEM aggregate images showed that primary particles tend to be aligned, but no quantitative information could be obtained. TEM is furthermore capable of producing diffraction patterns of the imaged particles. Electrons diffracted off different parts of the crystal lattice form characteristic interference patterns. These patterns can be indexed to identify crystal planes and the direction of the incident electron beam with respect to the crystal. Quantitative information about the orientation between primary particles can be obtained by performing such computations for each primary particle. Collier et al. [34] used TEM diffraction patterns to study the orientation of particles in calcite aggregates formed in an experiment at low ionic strength. Aggregates were classified according to the alignment of their primary particles into those with "perfect", "almost perfect", "partial", and "no alignment". It was found that 40% of the aggre-

	Single particle or agglomerated	Particle complexity	Particle shape	Polymorphic form
Size-independent descriptors	DFA [105], SVM [69], ANN [123]	DFA [105, 106], PCA-fit [39, 40]	DFA [105, 117]	unsupervised learning [118]
Including size-dependent descriptors	DFA [68, 74]			
Fourier descriptors		DFA [127]	DFA/ANN [117], ANN [128]	unsupervised learning [118]
Image and Fourier descriptors				SVM [70]

Table 2.1: An overview of investigated particle properties, variables and classification methods in case of automatic particle classification.

gates had either "perfect" or "almost perfect" alignment. Crystals produced at high ionic strength were visually observed by SEM and no sign of preferential orientation was found. This lead Collier et al. [34] to conclude that aggregates grown at low ionic strength probably have time to re-align themselves into an energetically more favorable state upon collision, forming aligned aggregates. This was explained by a thicker electrical double layer in case of low ionic strength, slowing the approach of the crystals. An alternative explanation stated that this layer holds crystals further apart so that only those aligned in a favorable position can grow a bridge strong enough to withstand the hydrodynamic forces.

Orientation between crystallites is often studied in material science as it influences the final properties of polycrystalline material [129, p. 1]. It is of special interest in bio- and nanomaterials where the observed structures cannot be explained by traditional crystallization models [130]. The orientations are measured using either electron (TEM)[34, 42] or x-ray [130] diffraction data. Mathematical expressions for expected orientation-related probability distribution functions can be computed in certain cases [131, 132]. Similar ideas can be applied to study a population of individual crystal aggregates, as shown by Collier et al. [34, 42], and further exploited in this work.

2.5 Goals and Open Questions

This chapter reviewed the established methods for particle size and shape measurement, with a focus on methods based on particle imaging. In crystallization, these methods represent tools for obtaining both kinetic information and a deeper, more fundamental understanding of the underlying mechanisms in order to achieve control of the PSD, purity, and other properties of the final product. Depending on the requirements, different imaging techniques can be employed to achieve this goal. Simple 2D methods can provide one-dimensional PSD, shape factors and the agglomeration degree of the product, which are sufficient for many practical applications. A more detailed investigation of the crystallization pro-

cesses requires more sophisticated measurement techniques. Face-specific growth rates of crystals that exhibit more than one face group can only be obtained from a 3D reconstruction of the particle shape. While this can be achieved using classical 2D imaging in case of simple morphologies, it in general requires more sophisticated tools such as stereoscopic methods using two 2D images, or 3D tomographic techniques. The complexity of the methods further increases with the complexity of the imaged particles. Apart from growth, crystals can also exhibit aggregation upon collision, leading two or more particles to assemble into a new particle and further grow together. This phenomenon can lead to the deviation from the required product properties. However, detailed study of aggregation has so far been difficult given that 2D imaging methods cannot provide the information about the size, shape and orientation of each involved primary particle. This type of information is necessary in order to understand which experimental and geometric conditions would lead to a formation of a stable aggregate when two crystals collide.

As it was shown in this chapter, little attention has been given so far to the development of characterization methods based on 3D imaging. These methods have a great potential in providing the necessary level of detail as no information about the particle shapes is lost through projection. In this work, methods of extracting the full 3D particle shape from μ CT images are developed and investigated on the example of potash alum crystals. They enable the measurement of the full 3D polytope in case of both single crystals and primary particles in crystal aggregates. Furthermore, it is shown how the orientation between primary particles can be measured, thus opening the possibility for detailed study of parameters that lead to aggregate formation without the need to resort to the study of TEM diffraction patterns. While the 3D imaging methods are unique in their ability to provide shape information, their drawback lies in the associated time efforts and cost, as well as in the need for particle sampling and preparation. 2D methods are still sufficient for practical applications, even in case of aggregation. While they may not facilitate fundamental understanding of aggregation, these methods can be used to compute the amount of aggregates in the product and thus find crystallization and post-processing conditions that lead to least aggregation. Thus, classifying 2D-imaged objects into single crystals and aggregates is also investigated in this work.

The experimental information, regarding both single crystals and aggregates, could be coupled with simulations to further study crystallization phenomena. Some of the required simulation and modeling techniques are already available, exemplified by multi-dimensional population balance solvers for crystal growth, reviewed in the work of Reinhold and Briesen [47]. The ability of the modeling techniques to predict crystal behavior can only be evaluated through comparison with experiments, leading back to the methods for sophisticated particle characterization and providing further motivation for methods studied in this work.

Chapter 3

Shape Identification of Single Crystals

3.1 Algorithm

This chapter presents a procedure for extracting the full 3D polytope from a 3D image of the crystal and is based on the corresponding publication [1]¹. The considered images are obtained by μ CT or by simulation, although the method is generally applicable to images acquired using similar tomographic techniques. It is assumed that each particle is stored in a separate image and that each particle represents a single crystal. Furthermore, the procedure is based on a comparison with a given crystal model, defining the directions of face normals that could potentially be found on the crystal surface. The finally obtained polytope consists of a subset of faces defined by the model. It is therefore possible that some of the model faces disappeared from the crystal surface, thus allowing different crystal morphologies of the modeled compound. The obtained polytope is symmetrical, using the symmetry defined by the crystal model. This step may be left out in order to better match the realistic experimental data, as it will be discussed in Chapter 5.

The procedure consists of three basic steps and is illustrated in a flow-diagram given in Figure 3.1. First, the directions of face normals in the crystal image are identified using Hough transform. Then, a correspondence between the identified face normals and the face normals given in the crystal model is established. Apart from deciding which identified faces are indeed present in the model and which model faces are visible on the imaged crystal, this step also involves finding the rotation matrix. Finally, the algorithm identifies the distance of each rotated model face from the particle center and applies symmetry conditions in order to obtain the final crystal shape. These steps are described in more detail in the following.

3.1.1 Identifying Face Normals

Hough transform is a classical tool for identifying lines in 2D images [107] [98] and has been discussed in the section 2.3.3.3 on shape fitting. The idea can be extended to finding planes in 3D point clouds,

¹In the previous publication, x - and y -axis of the crystals were permuted, which is corrected here. Further small bugs were fixed.

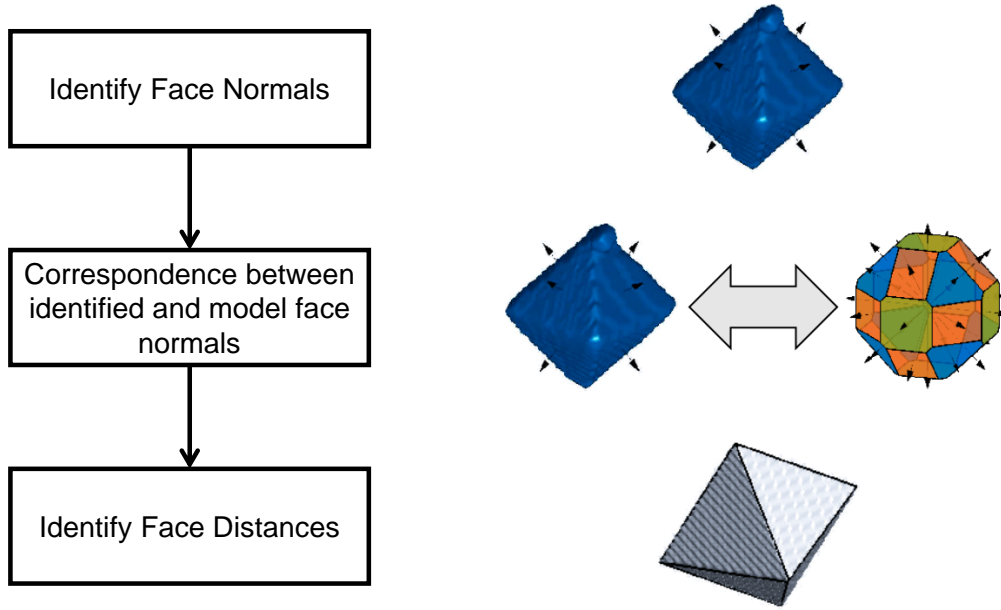


Figure 3.1: Procedure for identifying the shape of single crystals.

as presented by Borrmann et al. [133]. In the 3D case, this is achieved by discretizing the 3D space into a set of bins, where each bin defines a plane with a specified normal direction and a specific distance from the chosen middle point. Each bin is filled with points from the surface point cloud that are in the vicinity of the plane defined by that bin. Finally, bins with the highest number of contained points are detected, resulting in a set of planes that appear in the point cloud. The success of this procedure depends on the implementation details [133]. The procedure adopted in this work is given in the following.

The point cloud is given by the coordinates \mathbf{p}_k of foreground voxels on the crystal surface. These are foreground voxels that are connected to at least one background voxel using the 6-connectivity. Face normal directions \mathbf{n}_i are described by the polar angle ϕ_i and the azimuthal angle θ_i :

$$\mathbf{n}_i(\phi_i, \theta_i) = [\cos(\theta_i) \sin(\phi_i) \sin(\theta_i) \sin(\phi_i) \cos(\phi_i)]. \quad (3.1)$$

We furthermore consider a set of distances ρ_j from the chosen crystal middle point and define bins $b(\phi_i, \theta_i, \rho_j)$ that fully discretize the 3D space. Each so created bin represents the space between two planes with normals \mathbf{n}_i and distances $\rho_j - \frac{\Delta\rho}{2}$ and $\rho_j + \frac{\Delta\rho}{2}$, where $\Delta\rho$ is the discretization step for the distances ρ_j . Each bin is then filled with surface points \mathbf{p}_k that occupy this space.

To fully parametrize the face identification, the discretized face directions \mathbf{n}_i and face distances ρ_j are necessary. This is a non-trivial matter and is therefore discussed below in detail. As each considered object is a convex polytope, face distances measured from the polytope center of mass are always positive. Furthermore, distances are bounded by the smallest and largest distance of the surface points from the center. Thus, to simplify the computations, the point cloud is translated for the vector $-\mathbf{x}_{\text{mean}}$, where \mathbf{x}_{mean} is the arithmetic mean of the coordinates of the surface points. The

resulting translated surface points are denoted by $\tilde{\mathbf{p}}_k$, and their mean point coincides with the origin of the coordinate system.

The face distances ρ_j are discretized in the interval $(\rho_{\min}, \rho_{\max})$ with a distance $\Delta\rho$, so that the first discretization point is $\rho_1 = \rho_{\min} + \frac{\Delta\rho}{2}$, the i -th point is given by $\rho_i = \rho_{i-1} + \Delta\rho$, whereas the last discretization point is $\rho_{N_\rho} = \rho_{\max} - \frac{\Delta\rho}{2}$.² This ensures that the lower and upper bound of the considered bins lie inside the given interval. The bounds of the interval are given by

$$\rho_{\min} = c_{\min} \min_k d_k, \quad \rho_{\max} = c_{\max} \max_k d_k, \quad (3.2)$$

where d_k are distances of points $\tilde{\mathbf{p}}_k$ from the origin. The values c_{\min} and c_{\max} are equal to 1 in the case where both the normal directions and the surface points are infinitely finely discretized. However, the discretized face normal directions may not coincide with the actual face normals perfectly, so that the constants c_{\min} and c_{\max} may be set to values slightly smaller and larger than 1, respectively in order to compensate for this [52, p. 100]. The discretization step $\Delta\rho$ can be chosen in two manners. The first option is to fix the $\Delta\rho$ to some value, such as $\Delta\rho = 1$, so that it is equal to the voxel length, as the computation is done in voxel coordinates. In this case the discretization step does not depend of the size of the crystal and the imaging resolution, so that the same value $\Delta\rho$ is obtained for both small and large crystals. The second option is to define $\Delta\rho$ as

$$\Delta\rho = \frac{\rho_{\max} - \rho_{\min}}{N_\rho}, \quad (3.3)$$

where N_ρ is the given number of discretization points. The accuracy of the face distance computation is given by $\Delta\rho$, so that in this case the accuracy would be relative to the crystal size, instead of constant. Furthermore, $\Delta\rho$ depends on the crystal shape, as $\min_k d_k$ and $\max_k d_k$ differ very little for spherical particles and differ strongly for long crystals.

The discretized directions \mathbf{n}_i can be visualized as points on a unit sphere. An intuitive approach would be to sample the angles ϕ and θ uniformly. However, as discussed by Borrmann et al. [133], this would lead to the same number of points at the poles as on the equator, so that the pole is oversampled and the equator undersampled. We instead choose the HEALPix approach of Gorski et al. [134], where the sphere is divided into fields of equal area, each represented by one point which defines a direction \mathbf{n}_i in the 3D space. The obtained discretization is illustrated in Figure 3.2. The sphere is divided into 12 fields, each of which is further split into N_{side}^2 equally-sized sub-fields, finally resulting in $12 \cdot N_{\text{side}}^2$ points [134].

The Hough transform consists of traversing surface points and bins and incrementing the number of points $b(\phi_i, \theta_i, \rho_j)$ in the bin i by 1 for each surface point $\tilde{\mathbf{p}}_k$ that fulfills

$$\rho_j - \frac{\Delta\rho}{2} \leq \langle \mathbf{n}_i(\phi_i, \theta_i), \tilde{\mathbf{p}}_k \rangle < \rho_j + \frac{\Delta\rho}{2}. \quad (3.4)$$

²If $\Delta\rho$ is such that the value cannot be exactly satisfied, the closest point satisfying $\rho_i = \rho_{i-1} + \Delta\rho \leq \rho_{\max} - \frac{\Delta\rho}{2}$ is chosen as ρ_{N_ρ} . This can occur if $\Delta\rho$ is specified directly, instead of specifying N_ρ , see the discussion on choosing $\Delta\rho$.

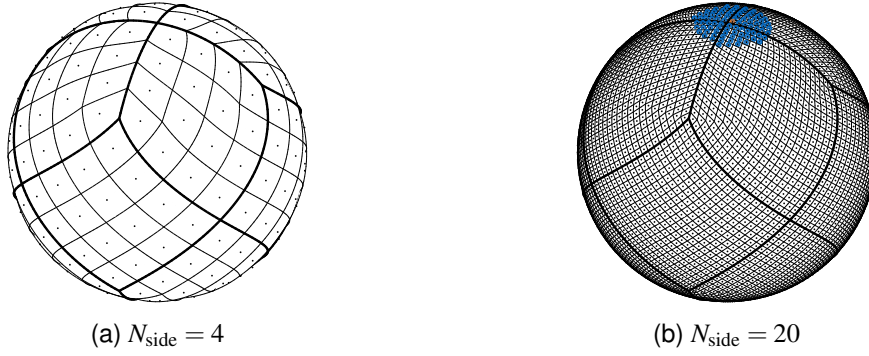


Figure 3.2: HEALPix discretization of a sphere into elements of equal area [134]. The 12 principal fields are presented as bold lines. Each principal field is divided into N_{side}^2 sub-fields drawn by thin lines, leading to a total of $12 \cdot N_{\text{side}}^2$ points. Borders of the HEALPix discretization were drawn using the HEALPix library for MATLAB by Y. Naruse [135]. In the right figure, points marked by blue represent the spherical cap centered at the orange point, where $\alpha_{\text{cap}} = 15.87^\circ$.

As the considered crystals are convex, there will be only one face in each considered face direction. Therefore, a new, reduced Hough space $\tilde{b}(\phi_i, \theta_i)$ is defined by considering only the bin with the maximal number of points in each face normal direction:

$$\tilde{b}(\phi_i, \theta_i) = \max_{\rho_j} b(\phi_i, \theta_i, \rho_j). \quad (3.5)$$

Finally, face normals represent the locations of maxima in this newly defined bin space. This can be viewed as finding maxima on a spherical grid defined by the HEALPix discretization and is done using the principle of non-maximum suppression [98, p. 59]. At the beginning, each grid point is considered to contain a maximum. A spherical cap is placed at each grid point, covering points that make an angle less or equal than α_{cap} with it. This is illustrated in Figure 3.2b. A grid point can only keep containing a maximum if its bin value is the maximal bin value when considering all points covered by the cap. This procedure ensures that two points within the spherical cap cannot simultaneously be maxima, unless the corresponding bins contain the exact same amount of points, which is unlikely to occur. Therefore, the angle between two identified face normals is larger than $2\alpha_{\text{cap}}$. The value α_{cap} is chosen as

$$\alpha_{\text{cap}} = 0.45 \min_i \alpha_i, \quad (3.6)$$

where α_i are all possible angles between pairs of different face normals in the considered crystal model.

Figure 3.3 illustrates an example of the reduced Hough space $\tilde{b}(\phi_i, \theta_i)$, obtained using a simulated and a real potash alum crystal. The crystals are octahedral and the crystal model considered 26 possible faces, as illustrated in Figure 2.1a. It can be seen that using a fixed $\Delta\rho = 1$ results in more points in Hough-transform bins.

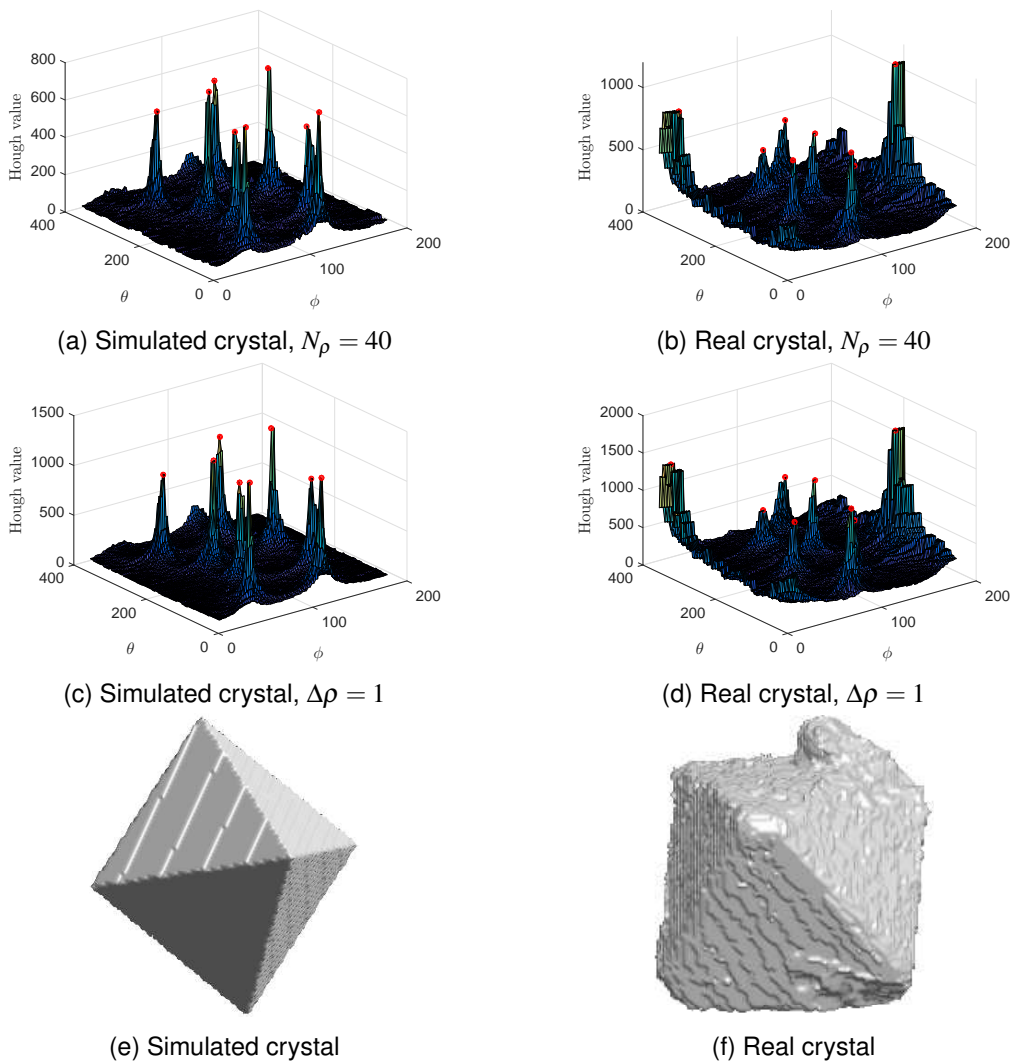


Figure 3.3: An illustration of the reduced Hough transform space $\tilde{b}(\phi_i, \theta_i)$ for a simulated and a real crystal, given in the bottom row. Peaks corresponding to identified face normals are marked by red dots. Crystals are octahedral, whereas the considered shape model contains 26 crystal faces, resulting in $\alpha_{\text{cap}} = 15.87^\circ$ and a spherical cap illustrated in Figure 3.2b. The figure was obtained by interpolating the angle values to a uniform grid.

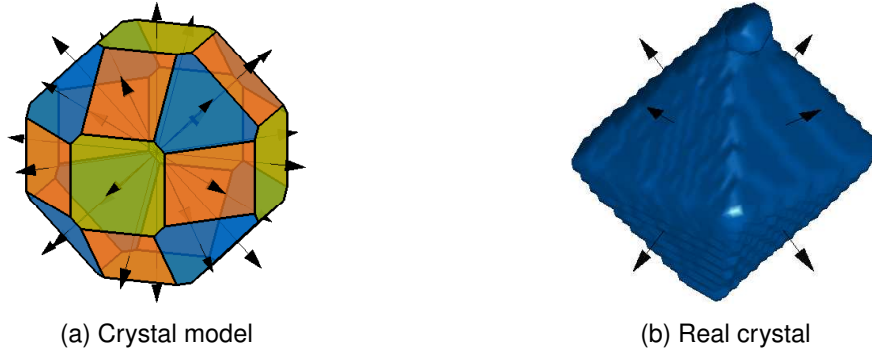


Figure 3.4: An illustration of the matching process. Crystal model containing 26 face normals that belong to three face groups is shown on the left. A real crystal is shown on the right, where the identified face normals are marked by arrows. All identified face normals belong to the "blue" face group; other face groups from the model crystal have not been identified on the surface of the real crystal. Reprinted with permission from [2]. Copyright (2016) American Chemical Society.

3.1.2 Establishing a Correspondence Between Measured and Modeled Faces

The procedure from the previous section results in a set of N_F identified face normals, gathered in the matrix \mathbf{A}_F . The task of this algorithm step is to match the identified face normals to the N_H face normals defined by the crystal model and represented by the matrix \mathbf{A} . Furthermore, the model crystal has some nominal orientation so that a rotation matrix \mathbf{R} , rotating the normals of the model crystal to coincide with the observed ones, is necessary. Often, fewer face normals are identified than what is expected according to the model, so that $N_F < N_H$. This is illustrated in Figure 3.4 for potash alum crystals modeled by 26 faces. Some faces, such as the green and orange model faces in Figure 3.4a may have grown out and are not visible on the surface of the observed crystal, presented in Figure 3.4b. It is also possible that the faces are present in the observed crystal, but are too small to lead to significant peaks in the Hough transform and were thus not identified. Similarly, the matrix of identified face normals \mathbf{A}_F may contain extra normals that do not correspond to those in the model crystal. This can happen in case of breakage, aggregation or attachment of small particles. Therefore, it is necessary to establish a correspondence between the N_H model normals in \mathbf{A} and N_F identified normals in \mathbf{A}_F . The matching process consists of three components explained in more detail below: a matching matrix \mathbf{S}_M , a filter matrix \mathbf{S}_D and a rotation matrix \mathbf{R} . The goal is to find these matrices and fulfill the following equation [1]:

$$\mathbf{S}_D \cdot \mathbf{A}_F \approx (\mathbf{S}_M \cdot \mathbf{A}) \cdot \mathbf{R}^T. \quad (3.7)$$

The matching matrix \mathbf{S}_M defines a match in \mathbf{A}_F for each model face normal in \mathbf{A} . This matrix has a size of $\tilde{N}_F \times N_H$, where \tilde{N}_F is the number of identified face normals that have a match among the model normals. Each row i in \mathbf{S}_M contains the value 1 at the position j corresponding to the match \mathbf{a}_j for the identified face normal $\mathbf{a}_{F,i}$. The filter matrix \mathbf{S}_D removes identified face normals from \mathbf{A}_F that have no match among the model normals in \mathbf{A} . Therefore, this matrix has a size of $\tilde{N}_F \times N_F$. Its j -th row has a value 1 in the j -th position if the j -th identified face normal $\mathbf{a}_{F,j}$ has a match among model normals.

All other values are equal to 0. Finally, the rotation matrix \mathbf{R} rotates the model face normals into the identified ones.

The equation 3.7 defines the goal of this algorithm step. However, the approximation sign still leaves some ambiguity regarding the three sought matrices. We can define an objective function, searching for a rotation matrix that gives the smallest error in the least-square sense:

$$L(\mathbf{R}) = \frac{1}{2\tilde{N}_F} \sum_{i=1}^{\tilde{N}_F} \left\| [\mathbf{S}_D \cdot \mathbf{A}_F]_i - [(\mathbf{S}_M \cdot \mathbf{A}) \mathbf{R}^T]_i \right\|^2. \quad (3.8)$$

This corresponds to the so-called Wahba problem [136, 137], where the task is to find a rotation matrix \mathbf{R} that rotates N 3D unit column vectors \mathbf{r}_i into the unit column vectors \mathbf{b}_i :

$$L(\mathbf{R}) = \frac{1}{2} \sum_{i=1}^N \omega_i \|\mathbf{b}_i - \mathbf{R}\mathbf{r}_i\|^2. \quad (3.9)$$

Here, ω_i are normalized weights, and are set to $\frac{1}{N}$ in the normal-matching problem defined in equation 3.8. Markley [137] gave a solution to this problem, based on singular value decomposition. It involves creating a matrix \mathbf{B} :

$$\mathbf{B} = \sum_{i=1}^N \omega_i \mathbf{b}_i \mathbf{r}_i^T, \quad (3.10)$$

and performing singular value decomposition into orthogonal matrices \mathbf{U} , \mathbf{V} and a diagonal matrix \mathbf{S} :

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \quad (3.11)$$

The rotation matrix minimizing the equation 3.9 is obtained as [137]

$$\mathbf{R} = \mathbf{U}[\text{diag}(1, 1, d)]\mathbf{V}^T, \quad (3.12)$$

where

$$d = \det(\mathbf{U}) \cdot \det(\mathbf{V}). \quad (3.13)$$

According to Markley [137], at least two linearly independent vectors are necessary to obtain a unique solution.

It is evident from equation 3.8 that the filter and mapping matrices \mathbf{S}_M and \mathbf{S}_D must be known in order to find the rotation matrix. A potential solution would be to test all possible mapping and filter matrices that can result from the given set of identified and model face normals. This is computationally very inefficient, given that the considered crystals tend to have a large number of faces, such as 8 to 26 in case of potash alum crystals, as illustrated in Figure 3.4. The approach adopted instead is to approximate the rotation matrix using only three most prominent identified face normals and consider all possible mappings of these three normals. If this leads to a reasonable approximation, a matching of the remaining normals is attempted and the case leading to the maximal number of matched normals is

chosen. The final rotation matrix is then computed using all matched normals. Details of this procedure are explained in the following and summarized in the flow-diagram in Figure 3.5.

First, three most prominent identified face normals are chosen as vectors \mathbf{b}_i . We denote these normals by $[p, q, r]$ and the matrix containing them by $\mathbf{A}_{F,[p,q,r]}$. Here, the prominence of a face normal is measured by the amount of points in the corresponding Hough transform bin. The face normals must not be colinear. This is achieved by ensuring that the angle between each pair of normals is between α_{\min} and $180^\circ - \alpha_{\min}$, where

$$\alpha_{\min} = 0.9 \min_i \alpha_i. \quad (3.14)$$

α_i are all possible angles between model face normals. The algorithm now considers all possible triplets $[m, n, l]$ of face normals from the crystal model and assigns them as matches \mathbf{r}_j for the chosen three normals. This thus sets three rows of the candidate mapping and filter matrices, $\mathbf{S}_{M,\text{cand}}$ and $\mathbf{S}_{D,\text{cand}}$. The matrix containing the assigned face normals is denoted as $\mathbf{A}_{[m,n,l]}$. Now, a rotation matrix candidate \mathbf{R}_{cand} can be computed using the equation 3.12. It minimizes the objective function³

$$L(\mathbf{R}_{\text{cand}}) = \frac{1}{6} \sum_{i=1}^3 \left\| [\mathbf{A}_{F,[p,q,r]}]_i - [\mathbf{A}_{[m,n,l]} \mathbf{R}_{\text{cand}}^T]_i \right\|^2. \quad (3.15)$$

The rotation matrix candidate is discarded if the error value $L(\mathbf{R}_{\text{cand}})$ is smaller than some threshold value e_{tol} .

A surviving rotation matrix candidate is used to rotate all model face normals. The algorithm proceeds by searching for a match among the N_H model normals for the remaining $N_F - 3$ identified face normals. Here, the angles between an identified normal $\mathbf{a}_{F,i}$ and the model normals rotated by \mathbf{R}_{cand} is computed. The match is accepted if the minimal such angle is smaller than the tolerance α_{tol} , meaning that:

$$\min_j \arccos(\langle \mathbf{a}_{F,i}, (\mathbf{R} \cdot \mathbf{a}_j) \rangle) < \alpha_{\text{tol}}. \quad (3.16)$$

This results in one row of the candidate matching- and filter matrices $\mathbf{S}_{M,\text{cand}}$ and $\mathbf{S}_{D,\text{cand}}$. In case that more than one identified normal was matched to the same model normal, the match is assigned to the most prominent of those normals. This procedure results in a set of candidate matching and filter matrices, $\mathbf{S}_{M,\text{cand}}$ and $\mathbf{S}_{D,\text{cand}}$ for each surviving candidate rotation matrix \mathbf{R}_{cand} . The candidates are accepted as \mathbf{S}_M and \mathbf{S}_D and the search is stopped if all identified face normals were matched, meaning that \mathbf{S}_D is an identity matrix. Otherwise, further triplets of $[p, q, r] \rightarrow [m, n, l]$ are tested and the candidate matrices leading to the largest number of matched normals are finally chosen as \mathbf{S}_M and \mathbf{S}_D . The final rotation matrix is then computed by solving Wahba's problem using the equation 3.12 for the obtained mapping and filter matrices, \mathbf{S}_M and \mathbf{S}_D . For the sake of clarity, the whole matching procedure is illustrated in Figure 3.5.

³Note that the vectors taken as rows of matrices \mathbf{A} and \mathbf{A}_F are row-vectors. Wahba's problem assumes column vectors in order to derive the solution in equation 3.12. Thus, vectors must be transposed before computing the solution.

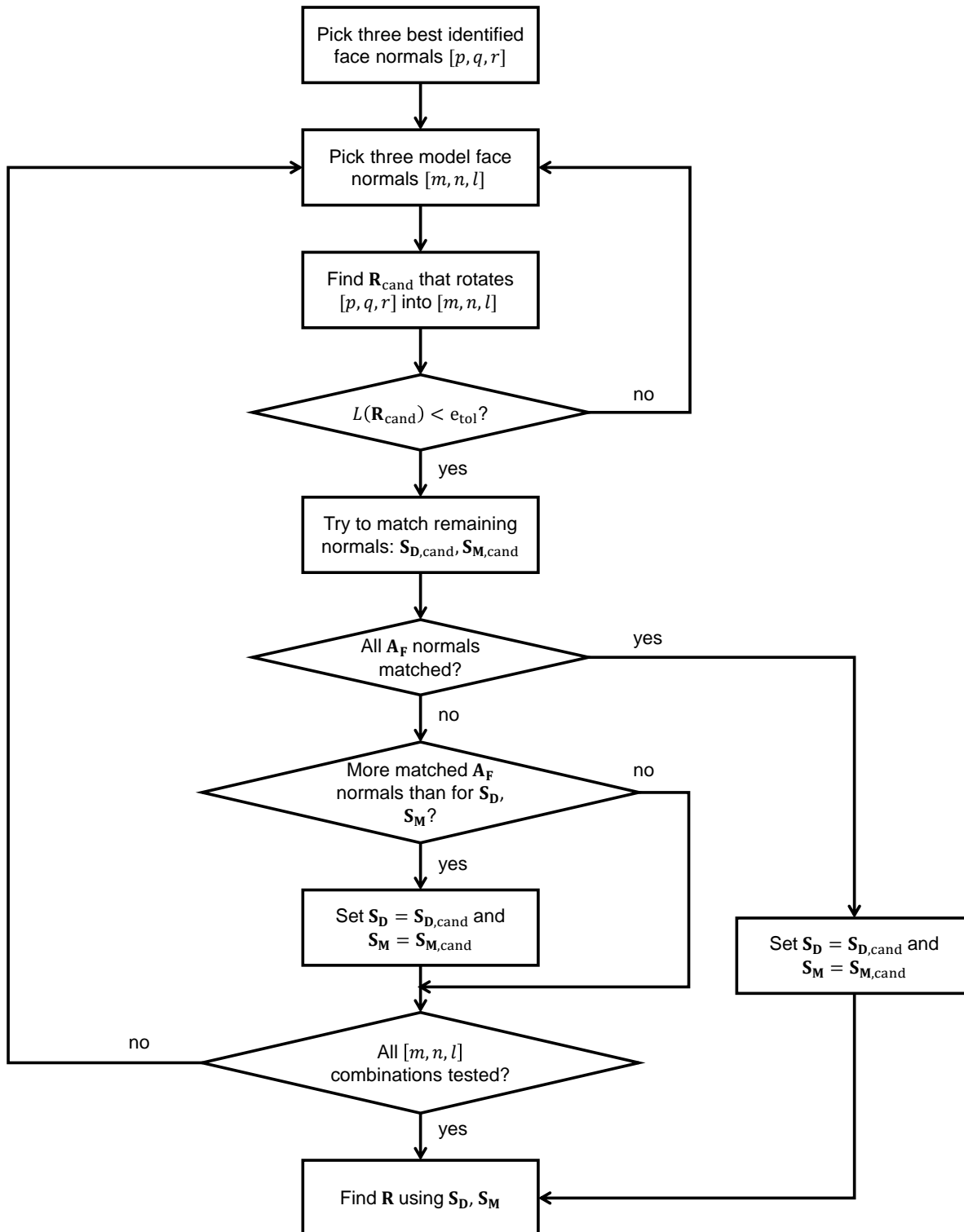


Figure 3.5: The procedure for matching identified to model face normals and finding the rotation matrix [1].

3.1.3 Identifying Face Distances

The previous step resulted in a rotation matrix \mathbf{R} and a correspondence between the model and the identified face normals. Therefore, the crystal can now be represented using the rotated version of the model face normals, \mathbf{AR}^T . It remains to find a face distance for each so defined face direction \mathbf{Ra}_i . These face distances are measured from the origin $(0,0,0)$ that coincides with the arithmetic mean of translated surface points \tilde{p}_i .

If there exists a match among the identified face normals \mathbf{A}_F for the currently considered rotated model normal \mathbf{Ra}_i , the face distance is obtained from the Hough transform. The algorithm finds the closest considered HEALPix normal \mathbf{n}_j , enclosing the smallest angle with \mathbf{Ra}_i . The face distance \tilde{h}_i is set to the value ρ_k corresponding to the most populated Hough-bin in the direction \mathbf{n}_j :

$$\rho_k = \max_k b(\phi_j, \theta_j, \rho_k). \quad (3.17)$$

If the considered rotated model normal \mathbf{Ra}_i has no match among \mathbf{A}_F , it is either very small or not present on the crystal surface. Thus, the Hough transform bin with the highest number of points will likely underestimate the face distance, as shown in Figure 3.6. In these cases, the algorithm searches for the Hough-transform bin with the highest point density r_j in the corresponding direction \mathbf{n}_j

$$r_j = \max_k \frac{b(\phi_j, \theta_j, \rho_k)}{A_{j,k}}. \quad (3.18)$$

Here, all crystal surface points \tilde{p}_l in the bin $b(\phi_j, \theta_j, \rho_k)$ are projected into the plane defined by the normal \mathbf{n}_j . To perform the projection into such a plane, the orthonormal basis of the vector \mathbf{n}_j is found, resulting in two vectors \mathbf{d}_1 and \mathbf{d}_2 in a plane normal to \mathbf{n}_j . The 2D coordinates of the projected points $\tilde{p}_{l,\text{proj}}$ in the plane spanned by $\mathbf{d}_1, \mathbf{d}_2$ are found by scalar product: $\tilde{p}_{l,\text{proj}} = (\langle \mathbf{d}_1, \tilde{p}_l \rangle, \langle \mathbf{d}_2, \tilde{p}_l \rangle)$. The value $A_{j,k}$ is the area of the convex hull of the points $\tilde{p}_{l,\text{proj}}$ from the bin $b(\phi_j, \theta_j, \rho_k)$ when projected to a plane

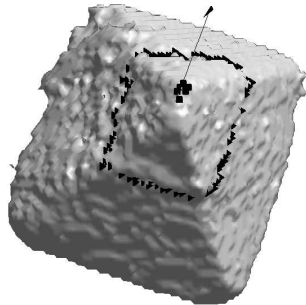


Figure 3.6: A procedure for finding the face distance in cases where the rotated model normal has no match among the identified normals. The plane corresponding to the most-populated Hough bin is presented by black triangles. The plane corresponding to the highest point density is represented by black squares. It can be seen that while there are more triangles than squares, the most populated plane would underestimate the true face distance.

normal to \mathbf{n}_j . The face distance \tilde{h}_i is finally set as

$$\tilde{h}_i = \rho_k, \quad (3.19)$$

using the value ρ_k that maximized equation 3.18.

Note that it is possible for the obtained face distances \tilde{h}_i to be such that the face does not touch the crystal polytope. This corresponds to the problem discussed in Figure 2.1. In this case, the H -representation contains redundant face normals that do not contribute to the crystal shape. These redundant normals can be identified and removed using the functionality present in the framework of Alexander Reinhold [52] and the Cddlib library [54]. The reduced polytope can be converted to the vertex V -representation, resulting in a set \mathbf{v}_k of polytope vertices. Face distances \tilde{h}_j corresponding to the redundant face normals can then be scaled to touch the polytope by so that [1]:

$$h_i = \max_k \langle \mathbf{v}_k, \mathbf{Ra}_i \rangle. \quad (3.20)$$

This results in a set of valid face distances h_i for the rotated model face normals \mathbf{Ra}_i , corresponding to a valid H -representation. However, face distances for the faces of the same face group may still be different. If the goal is to obtain a symmetrical, H_C representation, symmetry conditions must be applied.

3.1.3.1 Distance Correction

As shown above, the computed face distances represent Hough planes with either the highest number of points or the highest point density. Note that a point is assigned into a Hough bin if it fulfills equation 3.4. Each point represents the center of a surface voxel, moved so the mean of all voxel centers lies in the origin of the coordinate system. Therefore, each identified plane traverses the vicinity of surface voxel centers and the so obtained polytope is inherently smaller than the underlying crystal. This is illustrated in 2D in Figure 3.7. Since the computations are done in voxel coordinates, voxels having a dimensionless length of 1, a correction factor between $1/2$ and $\sqrt{3}/2$ should be applied to obtain a plane that touches the voxel vertices:

$$h_j = h_j + h_{\text{corr}}. \quad (3.21)$$

The exact amount depends on the orientation of the plane, as illustrated in Figure 3.7. However, a polytope based on the plane that touches the out-most voxel vertices would for certain orientations overestimate the crystal volume as it would incorporate some space that is not covered by the voxels. This is illustrated in the right part of the figure for a 2D case. The discussion is furthermore complicated by the binarization procedure. In case of μCT imaging, the gray value of a voxel can be considered to give information about the amount of solid material present in the voxel. However, there is a large spread of the gray values and a binarization procedure is performed, truncating those to either foreground or background. The best correction factor h_{corr} would have to be determined from the simulation of the

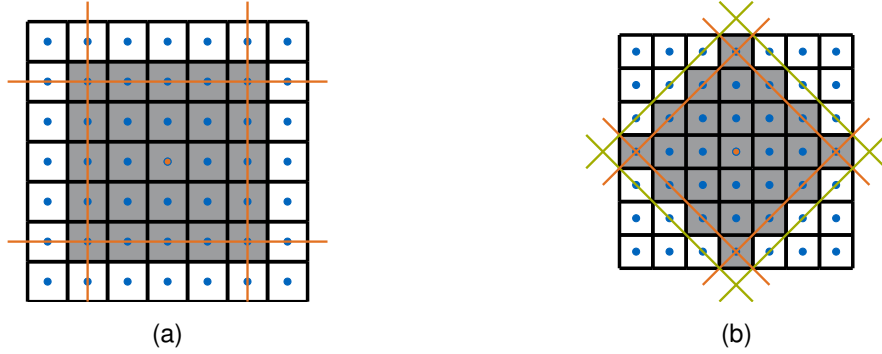


Figure 3.7: Illustration of the distance correction issue in 2D. The object is represented in gray, the pixels in white and pixel centers as blue points. In the left figure, an infinitely discretized Hough transform procedure will identify lines through pixel centers, so that the object bounded by orange lines is finally obtained. The distance of each line from the orange middle point is, therefore, half a pixel smaller than the true facet distance. In the right figure, the object bounded by orange lines would be too small, as well. However, if the distance is increased so that the lines touch the outmost pixel boundary points, the obtained object would be too large, as shown by green lines.

μ CT imaging and calibration procedures. For the sake of simplicity, no distance correction is performed in this work.

3.1.3.2 Symmetrical Crystals

The following equations for obtaining a symmetrical crystal are based on the discussion of fixed position by Reinhold and Briesen [53]. The current crystal is considered to be centered at $(0, 0, 0)$, where $(0, 0, 0)$ coincides with the arithmetic mean of the coordinates $\tilde{\mathbf{p}}$ of translated points on the crystal surface. If the crystal is symmetrical, then there exists some point \mathbf{x}_0 in space so that face distances within a face group are the same when measured from this point. The face distances computed from this point amount to

$$\mathbf{h}' = \mathbf{h} + \mathbf{A}_{\text{rot}}\mathbf{x}_0, \quad (3.22)$$

where $\mathbf{A}_{\text{rot}} = \mathbf{A}\mathbf{R}^T$ is the rotated matrix of model face normals. Since the crystal with face distances \mathbf{h}' is symmetrical, it can be converted into the symmetrical H_C -representation and back to H -representation without any loss of information

$$\mathbf{h} + \mathbf{A}_{\text{rot}}\mathbf{x}_0 = \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}} \mathbf{M}_{\mathbf{h} \rightarrow \mathbf{h}_C} (\mathbf{h} + \mathbf{A}_{\text{rot}}\mathbf{x}_0). \quad (3.23)$$

The matrix $\mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}$ is an $N_H \times N_C$ matrix of zeros and ones, stating which values of the vector \mathbf{h}_C should be copied onto which positions into the vector \mathbf{h} . The matrix $\mathbf{M}_{\mathbf{h} \rightarrow \mathbf{h}_C}$ computes the opposite operation and is obtained as the pseudo-inverse

$$\mathbf{M}_{\mathbf{h} \rightarrow \mathbf{h}_C} = \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}^+. \quad (3.24)$$

The equation 3.23 therefore simplifies to

$$\mathbf{M}_C (\mathbf{h} + \mathbf{A}_{\text{rot}} \mathbf{x}_0) = 0, \quad (3.25)$$

where

$$\mathbf{M}_C = \left(\mathbf{I} - \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}} \cdot \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}^+ \right). \quad (3.26)$$

In a general case, the crystal given by face distances \mathbf{h} is not symmetrical. The equation 3.25 cannot be exactly fulfilled and a vector \mathbf{x}_0 that minimizes

$$\|\mathbf{M}_C (\mathbf{h} + \mathbf{A}_{\text{rot}} \mathbf{x}_0)\|, \quad (3.27)$$

is sought. It is obtained as ⁴

$$\mathbf{x}_0 = - \left(\mathbf{M}_C \mathbf{A} \mathbf{R}^T \right)^+ \mathbf{M}_C \mathbf{h}. \quad (3.28)$$

due to the properties of pseudo-inverse matrices. The symmetrical crystal is obtained by

$$\mathbf{h}_C = \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}^+ \mathbf{h}', \quad (3.29)$$

which minimizes

$$\|\mathbf{h}' - \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}} \mathbf{h}_C\|. \quad (3.30)$$

The obtained polytope can be converted to the V -representation using the appropriate functionality of the Reinhold framework [52]. In order to ensure that the polytope remains valid after the applied symmetry conditions, invalid face groups are identified and their face distances scaled to touch the polytope using the approach defined in equation 3.20 for the first face in the group. A mapping from the H_C to H representation and the subsequent conversion to V -representation to get the matrix of vertices $\tilde{\mathbf{V}}$ completes the shape identification procedure. Note that this symmetrical crystal is inherently centered at (0,0,0) so that it can be used for population balance modeling. In order to get a polytope that coincides with the true surface points \mathbf{p}_i , the translation of both the surface points and the polytope must be negated. Thus, the vertices \mathbf{v}_i of the final polytope are obtained using

$$\mathbf{v}_i = \tilde{\mathbf{v}}_i + \mathbf{x}_{\text{mean}} - \mathbf{x}_0. \quad (3.31)$$

The crystal is thus centered in the point

$$\mathbf{x}_{\text{orig}} = \mathbf{x}_{\text{mean}} - \mathbf{x}_0. \quad (3.32)$$

⁴Note that the minus sign is missing in the corresponding publication [1]

3.1.4 Fit Quality Measures

Before using the shapes delivered by the shape identification procedure, the procedure itself as well as the obtained shapes must be evaluated. The accuracy of the method can be evaluated by simulating binary 3D images of crystals given by a polytope with a specified size and shape. The polytopes obtained from the shape identification procedure are then compared to the simulated ones. In case of experimental data, no such comparison is possible. However, the obtained polytope can be compared with the voxelized volume to obtain an estimation of the fit quality. Therefore, two sets of fit quality measures are defined.

3.1.4.1 Simulated Data

In this case, the simulated and the identified polytope are compared, where the details regarding simulation studies are given below. The resulting quality measures are therefore denoted by "s". The error in volume is computed as

$$q_{v,s} = \frac{V_s - V_f}{V_s}, \quad (3.33)$$

where V_s and V_f are the volumes of the simulated and fitted polytope, respectively.

For the i -th face group of the crystal, the face distance error can be computed as

$$q_{h,s,i} = \frac{h_{s,i} - h_{f,i}}{h_{s,i}}, \quad (3.34)$$

where $h_{s,i}$ and $h_{f,i}$ are the face distances of the simulated and fitted polytope, respectively. In this case, both crystals are centered at the coordinate system origin. If either of the considered crystals does not contain some face groups, they are added and it is ensured they touch the polytope exactly, so that they are in fact invisible.

Finally, if all possible symmetry operations of the given crystal model are known, it is possible to compute the disorientation angle. This is the smallest angle necessary to rotate one of the polytopes around some axis so that it coincides with the other. A procedure for computing this angle for octahedral crystals is given in chapter 5 on crystal orientations. In case of crystals with 26 faces, octahedral faces are used to compute the disorientation angle.

3.1.4.2 Experimental Data

The fit quality evaluation in case of experimental data can only be performed with respect to the voxelized volume. These measures are denoted by "r". The number of crystal voxels N_{voxels} approximates the volume of the crystal. A volume-mismatch quality measure is defined as:

$$q_{v,r} = \frac{N_{\text{outside}}}{N_{\text{voxels}}} + \frac{V_f - N_{\text{inside}}}{V_{\text{voxels}}}. \quad (3.35)$$

This value sums the approximated amount of the crystal which is outside the fitted polytope V_f , computed using the number of such voxels N_{outside} , with the empty amount left inside the crystal, approximated using the number of voxels inside the polytope, N_{inside} .

A second quality measure is obtained by projecting each of the N_{surf} surface points \mathbf{p}_j onto the fitted polytope to obtain the point $\mathbf{p}_{j,\text{proj}}$:

$$q_{\text{surf,r}} = \frac{\sqrt{\frac{1}{N_{\text{surf}}} \sum_{j=1}^{N_{\text{surf}}} \|\mathbf{p}_j - \mathbf{p}_{j,\text{proj}}\|^2}}{h_{\text{princ}}}. \quad (3.36)$$

Here, h_{princ} is the face distance of the main face group, which, in case of potash alum crystals, is the $\{111\}$ face group.

Projecting a point onto a polytope is a non-trivial problem and is computed using the steps described in appendix A.

3.2 Validation

The procedure for identifying the shape of single crystals is validated using both simulation and experiments. We chose potash alum as a model compound. Potash alum can be modeled using three face groups, the octahedral $\{111\}$ group, the cubic $\{100\}$ and the dodecahedral $\{110\}$ group, presented in Figure 3.4a. However, the later two faces grow out quickly, so that the experimentally observed crystals are often pure octahedra. Potash alum crystal model is generated using the functionality available in the framework of Alexander Reinhold and further information can be found in the corresponding thesis [52].

3.2.1 Simulated Data

3.2.1.1 Simulation Procedure

The simulation consists of two steps. First, 3D crystal polytopes are generated. Here, one reference H -representation, regarding the size and shape, is chosen. A population of N such crystals is obtained by applying N different rotation matrices to the chosen H -representation. A uniform sampling of rotation matrices is performed by choosing $Z - Y - Z$ Euler angles (α, β, γ) so that α and γ are uniformly sampled between $[-\pi, \pi]$, whereas $\beta = \arccos(t)$, where t is uniformly sampled between $[-1, 1]$ [138]. The final rotation matrix is obtained as

$$\mathbf{R} = \mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_z(\alpha), \quad (3.37)$$

where the rotations around the z and y axes are given by:

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}. \quad (3.38)$$

The second step concerns simulating binary 3D images of such particles with a chosen voxel length r . First, an empty 3D image with the length

$$l_i = \left\lceil \frac{(\lceil \mathbf{v}_{i,\max} \rceil - \lfloor \mathbf{v}_{i,\min} \rfloor)}{r} \right\rceil \quad (3.39)$$

in dimension i is created. Here $\mathbf{v}_{i,\max}$ and $\mathbf{v}_{i,\min}$ are the minimal and maximal coordinates of the polytope vertices in the i -th dimension. Each voxel, given by its indices (x, y, z) in the empty 3D image matrix is now set to foreground if the corresponding point $\mathbf{p}_{x,y,z}$ is within the polytope. Here, the voxel with indices (x, y, z) first must be translated into the coordinate system of the crystal centered at $(0, 0, 0)$ to obtain the point $\mathbf{p}_{x,y,z}$ [2]:

$$\mathbf{p}_{x,y,z} = ((x, y, z) - 0.5) \cdot r + \lfloor (\mathbf{v}_{x,\min}, \mathbf{v}_{y,\min}, \mathbf{v}_{z,\min}) \rfloor. \quad (3.40)$$

Each crystal image is stored in a separate file.

3.2.1.2 Simulated Data Sets

The goal of the simulation study is to evaluate the scheme quality for different crystal shapes, as well as the influence of algorithm parameters on the scheme quality. We consider three crystal shapes of potash alum, given in Figure 3.8. For each crystal shape, five data sets were generated. Each data set contained 100 crystals with different orientations and a specified volume-equivalent sphere diameter, where the analyzed values were $200 \mu\text{m}$, $300 \mu\text{m}$, $400 \mu\text{m}$, $500 \mu\text{m}$ and $600 \mu\text{m}$. All simulations were conducted with the same μCT resolution of $r = 7.85 \mu\text{m}$ corresponding to the one used in experiments for crystal orientation measurement presented in Chapter 5. The study can equivalently be understood as varying the resolution while keeping the size constant.

Parameters e_{tol} , N_{side} and α_{tol} were kept constant while other parameters were varied. Simulation cases given in Table 3.1 were investigated, where each case consisted of 500 crystals, as 100 crystals of each size were simulated. For cases using shapes 1 and 3 simulations were conducted both for a crystal model with 8 faces and a model with 26 faces. Shape 2 was investigated using a crystal model containing 26 faces.

3.2.1.3 Results

A simulation and a corresponding shape identification procedure was performed for each case given in Table 3.1. In each case, crystal symmetry is used, as explained in section 3.1.3.2. The first case,

Case	No. model faces	Shape	$N_{\text{rho}} / \Delta\rho$	c_{min}	c_{max}	Voxel length [μm]	N_{side}	e_{tol}	$\alpha_{\text{tol}} [^\circ]$
S1-1	8	1	$N_{\text{rho}} = 40$	1	1	7.85	20	0.02	10
S1-5	8	1	$\Delta\rho = 1$	1	1	7.85	20	0.02	10
S1f-5	26	1	$\Delta\rho = 1$	1	1	7.85	20	0.02	10
S2-1	26	2	$N_{\text{rho}} = 40$	1	1	7.85	20	0.02	10
S2-5	26	2	$\Delta\rho = 1$	1	1	7.85	20	0.02	10
S2-8	26	2	$\Delta\rho = 1$	0.95	1.05	7.85	20	0.02	10
S3-1	26	3	$N_{\text{rho}} = 40$	1	1	7.85	20	0.02	10
S3-5	26	3	$\Delta\rho = 1$	1	1	7.85	20	0.02	10
S3e-1	8	3	$N_{\text{rho}} = 40$	1	1	7.85	20	0.02	10
S3e-5	8	3	$\Delta\rho = 1$	1	1	7.85	20	0.02	10

Table 3.1: Parameters for simulation and shape identification of single crystal data sets where the shapes are illustrated in Figure 3.8.

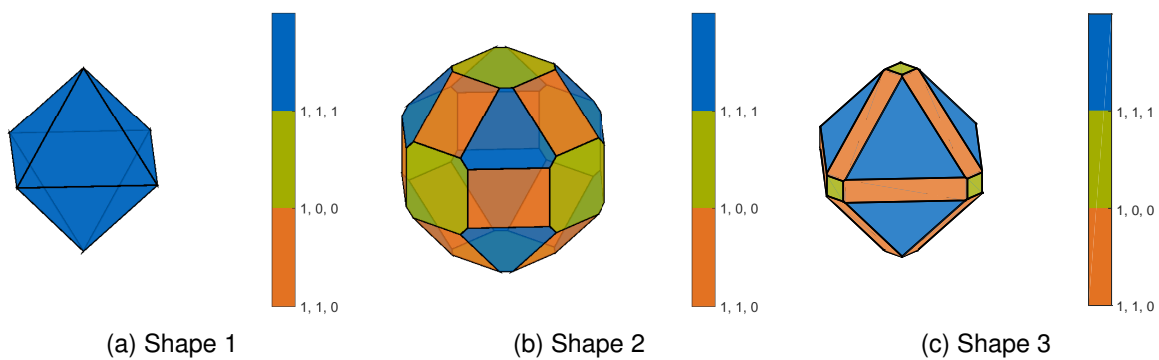


Figure 3.8: Potash alum crystal shapes used in the validation of the shape identification procedure.

S1-1, corresponds to octahedral crystals, where the shape identification procedure uses an octahedral crystal model. Mean error values regarding the case S1-1 are given in Figure 3.9. Identified shapes for one crystal orientation and different sizes are given in Figure 3.10. Here, the simulated crystal is rendered in white based on the simulated 3D image, while the identified shape is presented as a black polytope. The orientation of the crystal is identified correctly, which can be seen by the small values of disorientation angle given in Figure 3.9a even for the smallest volume-equivalent sphere diameter of $200\ \mu\text{m}$. The errors in volume and face distance are large for small crystals and improve as crystal size increases. The improvement decreases at an equivalent diameter of about 400. However, even for the largest crystals, there is still a mean error of about 5% in volume and a mean error of about 2% in face distance. This is caused by setting a crystal face distance based on Hough transform data containing points that correspond to voxel center coordinates. The resulting crystal will always be slightly too small, leading to a positive value of $q_{v,s}$ and $q_{h,s,1}$. This issue was discussed in section 3.1.3.1.

The case S1-5 is obtained by using the fixed value $\Delta\rho = 1$ instead of fixing the number of Hough bins, as in case S1-1. The results are shown in Figure 3.11. An improvement can be seen in all error measures.

In order to validate the measurement of faces that have grown-out, the shape identification procedure for octahedral crystals is repeated for a shape model consisting of 26 faces. This corresponds to the case S1f-5 and the error measures are given in Figure 3.12. In the case of simulated crystals, distance of the $\{100\}$ and $\{110\}$ faces, which were in fact not simulated, was set such that the faces touch the crystal polytope at its vertices. Mean error in orientation, measured by the disorientation angle, as well as mean volume error, remained similar as in case S1-5 using the crystal model with 8 faces. The mean face-distance error shows three curves, each corresponding to one face group. The error regarding face group $\{111\}$ is similar to the case S1-5, whereas the mean error of the face group $\{110\}$ behaves similarly. However, the mean error is significantly larger for the face group $\{100\}$. The issue is further illustrated in one example crystal, shown in Figure 3.13a, where the faces $\{100\}$ and $\{110\}$ are visible in the identified shape, even though the simulated crystal does not contain such

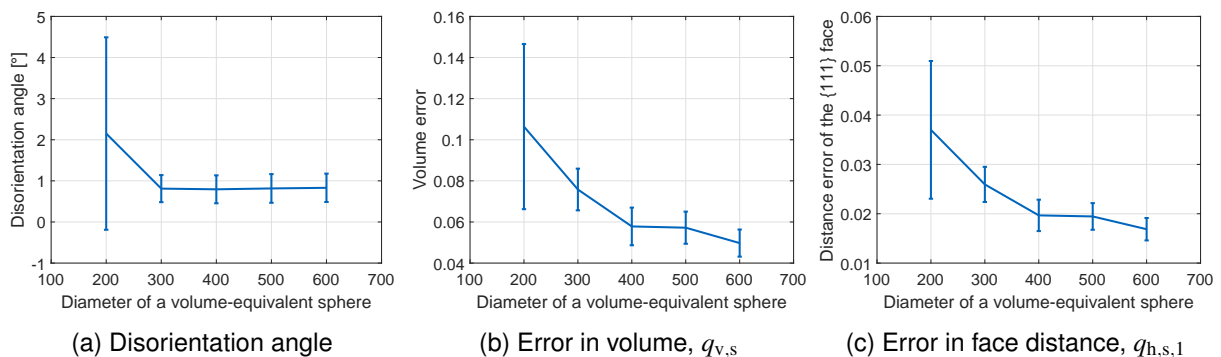


Figure 3.9: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1-1, given in Table 3.1.

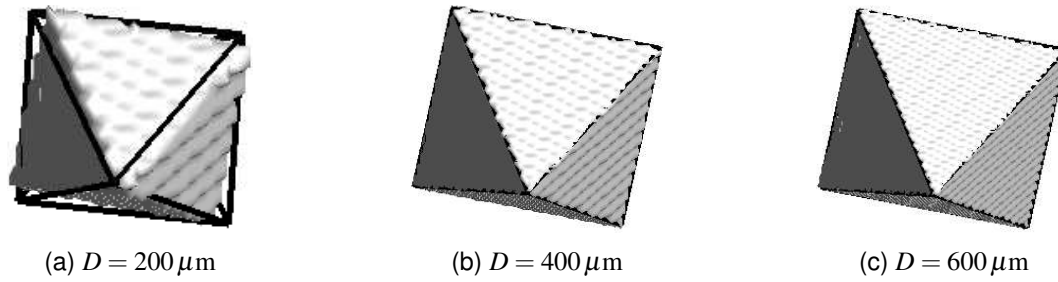


Figure 3.10: Examples of shape identification results for one crystal orientation and different volume-equivalent sphere diameters D . Simulation and shape identification parameters correspond to case S1-1, given in Table 3.1. The simulated crystals are rendered in white, while the identified shapes are presented as black polytopes.

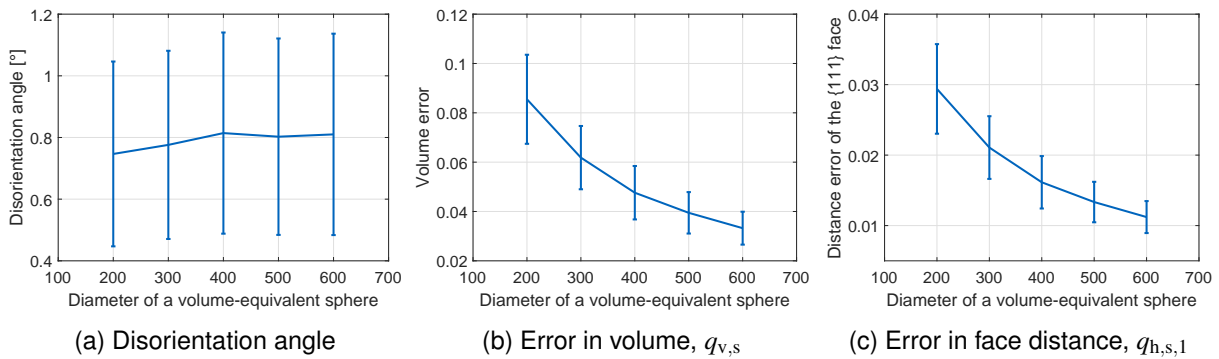


Figure 3.11: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1-5, given in Table 3.1.

faces. This is likely caused by the approach for identifying the face distance of a face which was not matched, defined by equation 3.18. As at least three points are necessary to form the convex hull, this will necessarily fit a plane slightly under the correct position. An alternative approach would be to fit the plane at the position corresponding to the last visible point in the given direction. However, this would be very sensitive to surface roughness in case of experimentally measured crystals. This error decreases significantly with the increase in crystal size, as illustrated on one example in Figure 3.13d, where the falsely detected $\{100\}$ faces are practically invisible.

It is important to note that the change of crystal model also changes the parameter α_{cap} , defined in equation 3.6, as this value depends on the smallest angle between face normals in the crystal model. However, the comparison of results for cases S1-5 and S1f-5 shows that the scheme is quite robust regarding this parameter. Furthermore, the crystal model containing 26 faces was such that the first eight faces are the $\{111\}$ faces corresponding to the octahedral crystal model. Therefore, the matching procedure will first iterate through these faces, leading both to a fast solution and avoiding potential errors if a match with other face groups is found. It is thus recommended to order faces in the crystal model such that faces that are most likely to appear in the crystal are the first ones to consider.

Results regarding shape 2 given in Figures 3.14, 3.15 and 3.16. The Figure 3.14 corresponds to the case S2-1. The mean error in orientation, measured by the disorientation angle, is significantly higher than in case of octahedral crystals for volume-equivalent sphere diameter lower than $D = 500 \mu\text{m}$. Figure 3.13b illustrates the identified shape for a crystal with $D = 200 \mu\text{m}$. It can be seen that faces are small and difficult to identify, resulting in a wrongly oriented identified polytope, while the overall identified shape is correct. In Figure 3.13e, an example of a large crystal with a correctly identified orientation is presented, further illustrating that the faces are more easily identifiable with the increase in crystal size. It is interesting to note that the mean error in volume and face distance is small for all crystal sizes, where even the mean error regarding the $\{100\}$ face group is below 5%. Thus, it is not sufficient to consider only the size-based error measures in order to quantify the fit quality.

The effect of setting a fixed $\Delta\rho = 1$ instead of using 40 Hough bins is shown in Figure 3.15, corre-

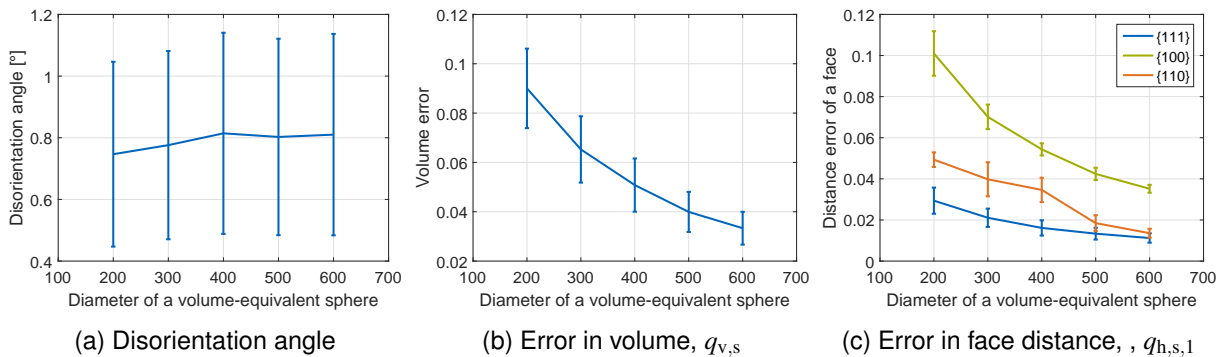


Figure 3.12: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1f-5, given in Table 3.1.

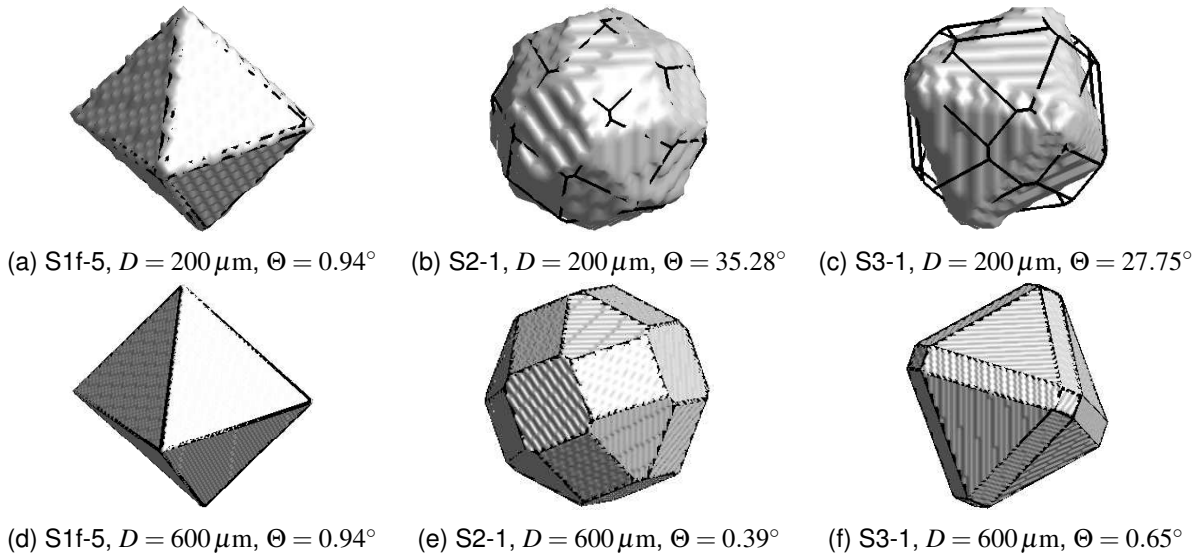


Figure 3.13: Examples of shape identification results for one crystal orientation and different volume-equivalent sphere diameters D . Simulation parameters for the corresponding cases are given in Table 3.1. The simulated crystals are rendered in white, while the identified shapes are presented as black polytopes.

sponding to the case S2-5. A significant improvement regarding the disorientation angle can be seen. Additionally changing the values of c_{\max} and c_{\min} results in the case S2-8, shown in Figure 3.16. The disorientation angle worsens with respect to S2-5. Therefore, it can be concluded that the Hough transform parameters can have a significant influence on the performance of the scheme in case of small crystals, or alternatively, low resolution realized by a large voxel length.

Similarly like shape 2, shape 3 also contains all 26 faces, with the difference that the $\{100\}$ and $\{110\}$ faces are less prominent than the $\{111\}$ face, as seen in Figure 3.8c. Results for the case S3-1 are given in Figure 3.17. The mean disorientation angle is large for the volume-equivalent sphere diameter of $D = 200 \mu\text{m}$, but it decreases to under 5° for $D = 300 \mu\text{m}$ already. An example for a small

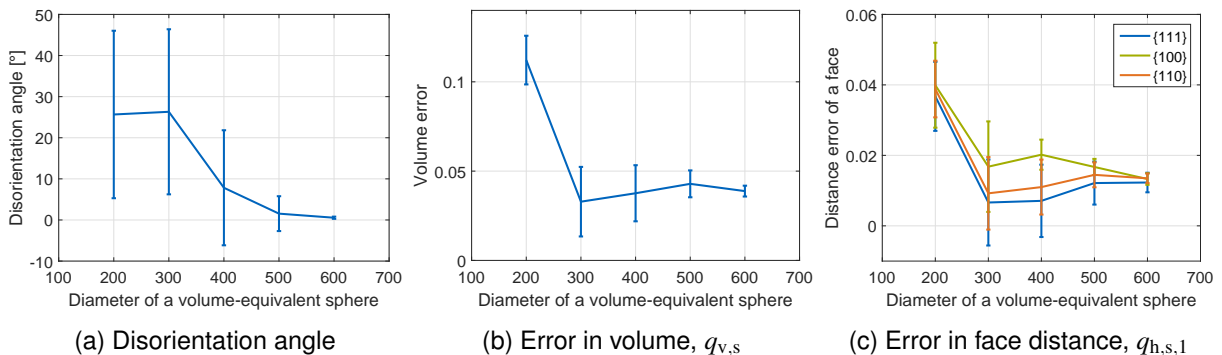


Figure 3.14: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S2-1, given in Table 3.1.

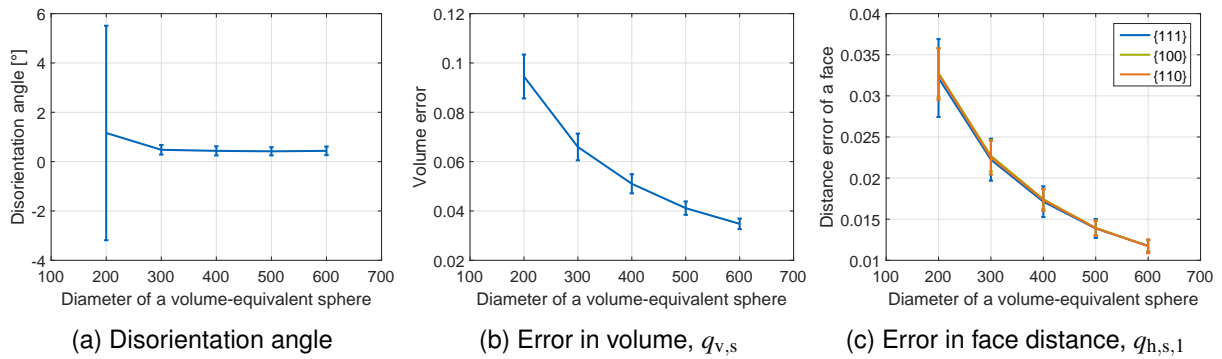


Figure 3.15: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S2-5, given in Table 3.1.

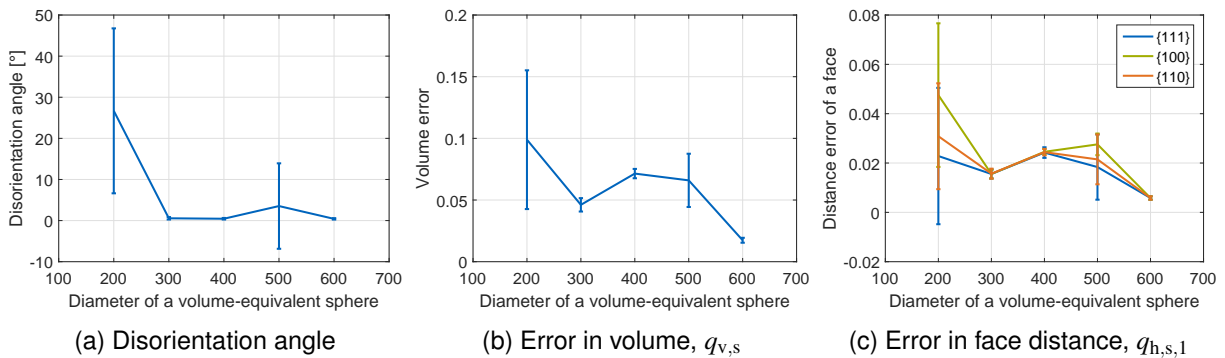


Figure 3.16: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S2-8, given in Table 3.1.

crystal is illustrated in Figure 3.13c. Apart from a wrongly identified orientation, $\{100\}$ and $\{110\}$ faces are identified to be larger than simulated, so that the identified shape resembles shape 2 more than shape 3. The accuracy improves with the increasing crystal size, so that in case of a large crystal in Figure 3.13f, both size and orientation are identified correctly. Similarly as in case of shape 2, using a fixed discretization step of $\Delta\rho = 1$ leads to an improvement in the disorientation angle, as seen in Figure 3.18.

Finally, the simulated crystals of shape 3 were identified using a shape model with 8 faces in order to evaluate the error obtained when using a simpler crystal model. The results regarding the case S3e-1 from the Table 3.1 are presented in Figure 3.19. The values are reported for cases where shape identification was possible. This was the case for only 40 out of 100 crystals for $D = 200\mu\text{m}$, whereas the value increased to 99 in case of $D = 500\mu\text{m}$, and 100 for the largest crystals with $D = 600\mu\text{m}$. The disorientation angle reaches a mean value below 5° already for a $D = 300\mu\text{m}$, meaning that the orientation can be identified with an acceptable accuracy for all but the smallest particles. The volume error indicates that the identified polytope is too large. The face distance of the $\{111\}$ face is somewhat smaller than simulated, similarly as in the other cases. The face distances in case of $\{100\}$ and $\{110\}$ faces are too large, which is to be expected as these faces are not visible in the identified polytope that contains only the eight $\{111\}$ faces. As in the previous cases, an improvement in the disorientation angle is obtained by using the fixed distance $\Delta\rho = 1$, corresponding to the case S3-5 and seen in Figure 3.18. Furthermore, unlike in case S3e-1 where $N_\rho = 40$, shape identification was possible for all particles of all sizes for the case S3e-5.

Overall, the presented study has shown that the algorithm is capable of identifying the shape of the crystals in the simulated binarized 3D images. However, the results are sensitive to the algorithm parameters and the ratio of the crystal size towards the imaging resolution. A poor choice of algorithm parameters can lead to poorly identified crystal orientation in cases when many small faces are visible on the crystal surface. However, in case of octahedral crystals that contain eight well-visible faces, all investigated parameters lead to sufficient accuracy. Therefore, simulation studies should be conducted

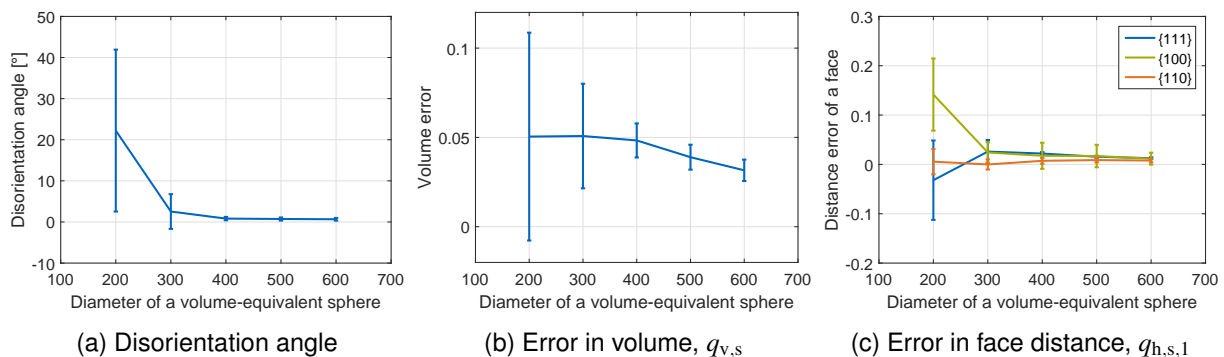


Figure 3.17: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3-1, given in Table 3.1.

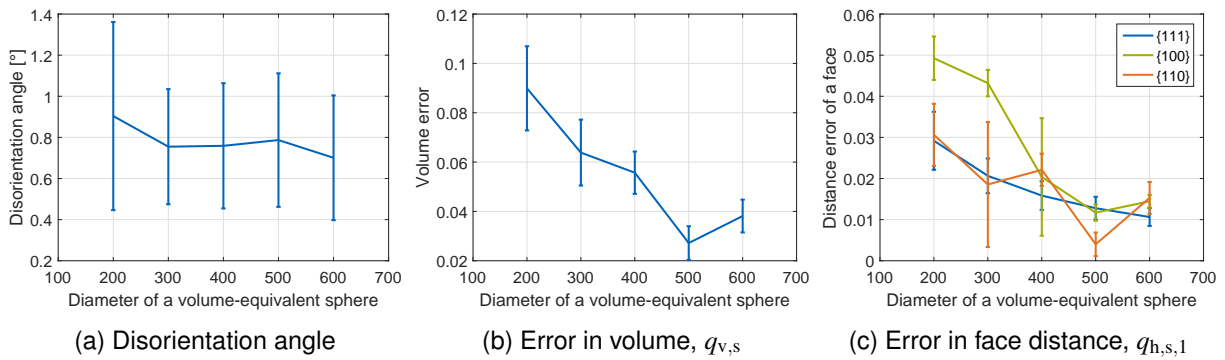


Figure 3.18: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3-5, given in Table 3.1.

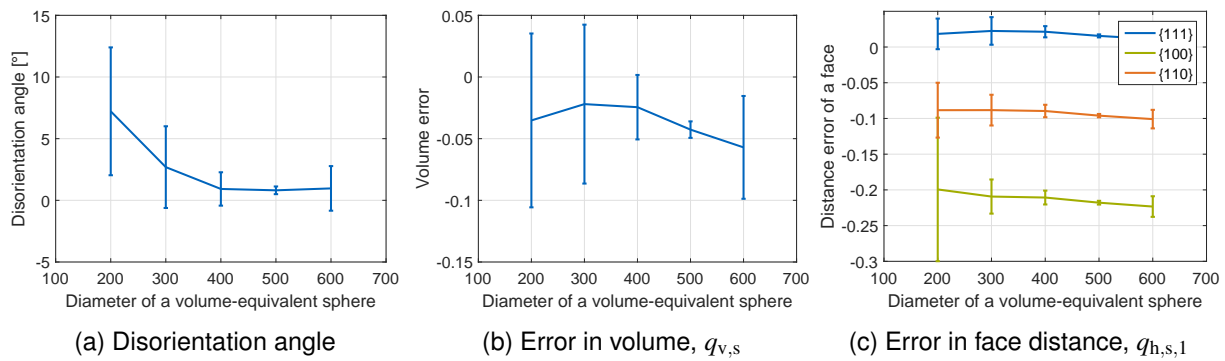


Figure 3.19: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3e-1, given in Table 3.1.

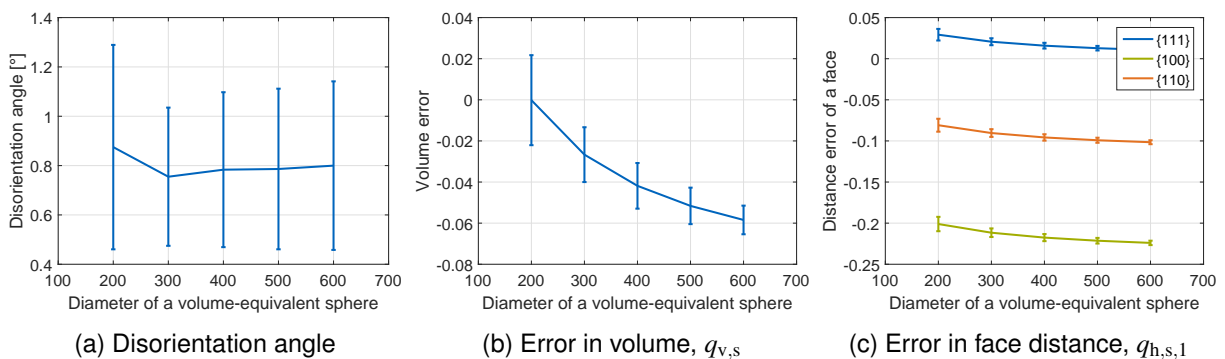


Figure 3.20: Mean error measures for simulated 100 potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S3e-5, given in Table 3.1.

before imaging a new type of particle in order to determine the imaging resolution sufficient to eliminate the parameter sensitivity for the given particle shape and size.

3.2.2 Experimental Data

3.2.2.1 Preparation and Measurement

Potash alum crystals used for the validation study were kindly provided by Stefan Schorsch (group of Prof. Mazzotti, ETH Zürich) and Alexander Reinhold (TUM, SVT). A 0.65 mL test tube was filled with styrofoam upon which potash alum particles were added in order to enhance particle separation and reduce movement. A portion of the test tube was imaged in a custom-made μ CT system available at the department (type: XCT-1600HR, supplier: Matrix Technologies, Feldkirchen, Germany). The relevant measurement parameters are given in Table 3.2, whereas the parameters for the shape identification procedure are presented in Table 3.3. The measurement lasted about 1.5 hours. The obtained crystals are visualized in Figure 3.21.

3.2.2.2 Pre-processing

Crystals were binarized using the Otsu threshold using the software MAVI [96]. As it can be seen in Figure 3.21, particles touch and a simple labeling procedure would result in more than one particle per label. MAVI tool "Particle separation by complex morphology" was used to segment individual particles. According to the MAVI manual [96], this tool uses the H-maxima transform to prevent over-segmentation in the watershed transform. The algorithm uses the parameter called "dynamics", as explained in section 2.3.2.2. The value of this parameter was set to 5. Segmented crystals were stored in separate files using MATLAB. Those whose volume contained fewer than 40000 foreground voxels, as well as non-convex particles, were ignored. Particle convexity was estimated by

$$r_{\text{conv}} = \frac{N_{\text{voxels}}}{V_{\text{convex hull}}}, \quad (3.41)$$

where N_{voxels} is the number of foreground voxels, approximating the volume, and $V_{\text{convex hull}}$ is the volume of the convex hull computed using voxel centers as point coordinates. A particle was considered convex

Parameter	Value
Tube Voltage [kV]	80
Power [W]	3.6
Exposure Time [ms]	2132
Source-to-object distance [mm]	115.36
Source-to-image distance [mm]	415.29
No. projections	2500
Voxel length [μm]	5

Table 3.2: Parameters of the μ CT measurement.

Case	No. model faces	$N_{\text{rho}} / \Delta\rho$	c_{min}	c_{max}	Voxel length [μm]	N_{side}	e_{tol}	$\alpha_{\text{tol}} [^\circ]$
Exp. data single	26	$N_{\text{rho}} = 40$	1	1	5	20	0.02	10

Table 3.3: Parameters for shape identification of experimental data: validation of the single crystal procedure.

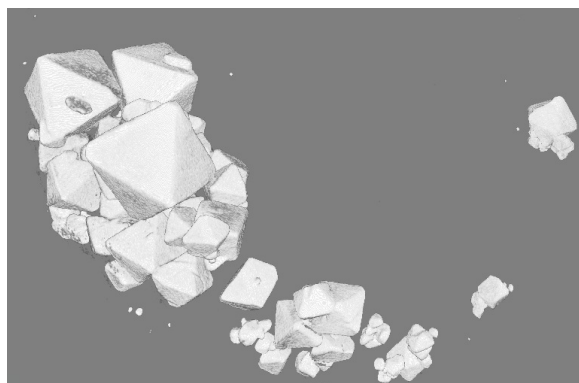


Figure 3.21: Experimentally measured potash alum particles used for validating the method for identifying the shape of single crystals. Visualization done by MyVGL [139]. Reprinted with permission from [1]. Copyright (2014) American Chemical Society.

if $r_{\text{conv}} > 0.85$. This finally resulted in 22 crystals. Time requirements for the pre-processing procedure are in order of a few minutes.

Note that the separation using H-maxima procedure leads to a loss of a layer of voxels around particles. These are watershed voxels that are equally likely to belong to all labels surrounding the voxels, where these labels may represent particles or the background.

3.2.2.3 Results

Shape identification was performed using the parameters from Table 3.3⁵. Figure 3.22 exemplary shows highly regular crystals where the fitting procedure performed well. Quadratic distance deviation values were up to $q_{\text{surf},r} = 0.06$, whereas the amount of mismatched volume was up to $q_{v,r} = 0.15$. The scheme is furthermore capable of identifying a crystal shape also in case where small particles are attached to the large ones, as shown in Figure 3.23. Here, quadratic distance deviations were up to 0.13, whereas the mismatched volume error was up to 19%.

Examples where the scheme did not perform as well are presented in Figure 3.24. Crystals appear to be broken and cause a quadratic distance deviation up to 0.19 along with a high volume mismatch error of up to 47%. This could be a result of over-segmentation of the particle separation procedure or could have occurred during transport from Zürich. The obtained shape can be understood as an

⁵Note that the parameter N_p was different in the publication [1]. There is no significant difference when selecting $N_p = 50$ as used previously, but the value $N_p = 40$ was selected to ensure faster processing. Error measure regarding volume was re-defined to be scaled by the number of voxels instead of the volume of the fitted shape. The new results include bug fixes and a correct display of the crystals, which were previously mirrored as the x and y coordinates were inverted.

approximation of the original particle shape before breakage. Figure 3.25a shows the result of applying this scheme on a crystal aggregate. As the particle does not fulfill the assumptions of the scheme, no reasonable shape fit can be expected. Such particles are investigated in the next section and will lead to the result shown in 3.25b. Note that an algorithm that assumes the crystal may represent an aggregate is likely to treat particles in Figure 3.23 as aggregates as well. As the two schemes were integrated, the user can choose to "force" the single crystal procedure if prior knowledge about the particles is available.

3.2.2.4 Execution Time

The sequential execution time on a desktop computer with 16 GB RAM and Intel Core(TM)i5-3470 CPU @ 3.20 GHz running MATLAB 2015b on a 64-bit Windows 8.1 Enterprise was about eight minutes for the 22 considered crystals. There is an improvement in the execution time with respect to the old program version using $N_p = 50$ reported in [1], where an older MATLAB version was used.

3.2.3 Conclusion

The results presented in this chapter show that the developed scheme is capable of identifying the shape of single crystals, even when such crystals contain small particles attached to their surface. The result depends strongly on the quality of the underlying images, which should ideally show symmetrical, highly regular single crystals. Whether this requirement is met depends, apart from the quality of the crystals themselves, on the preparation procedure and the pre-processing steps used to segment the crystals. Therefore, further chapters will also aim at improving these aspects.

Furthermore, the scheme accuracy depends on the shape of the underlying particles. Particles containing many crystal faces should be imaged using a higher resolution than particles containing few faces, as shown in section 3.2.1.3 where the results of the study using simulated particles are presented.

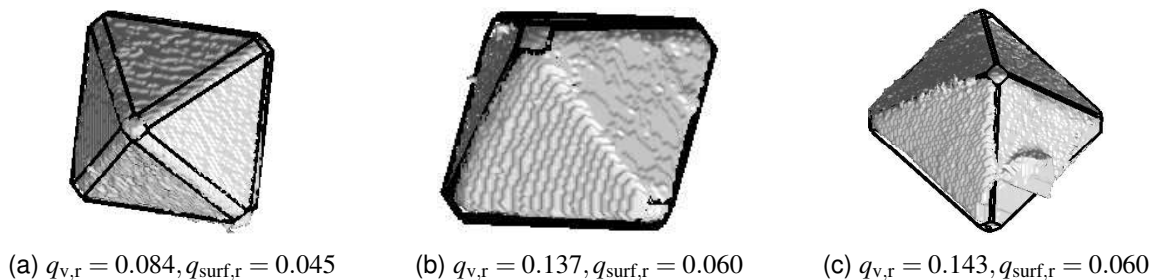


Figure 3.22: Shape identification in case of single crystals: examples where the procedure performed well and the crystals were highly regular.

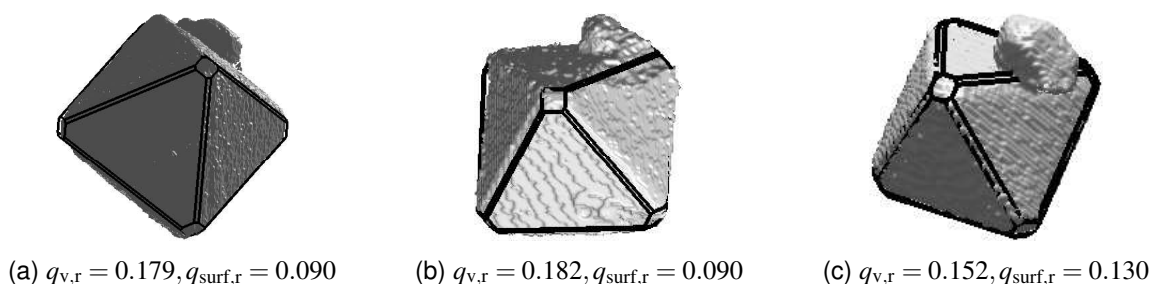


Figure 3.23: Shape identification in case of single crystals: examples where the procedure performed well and crystals contained small particles at the surface.

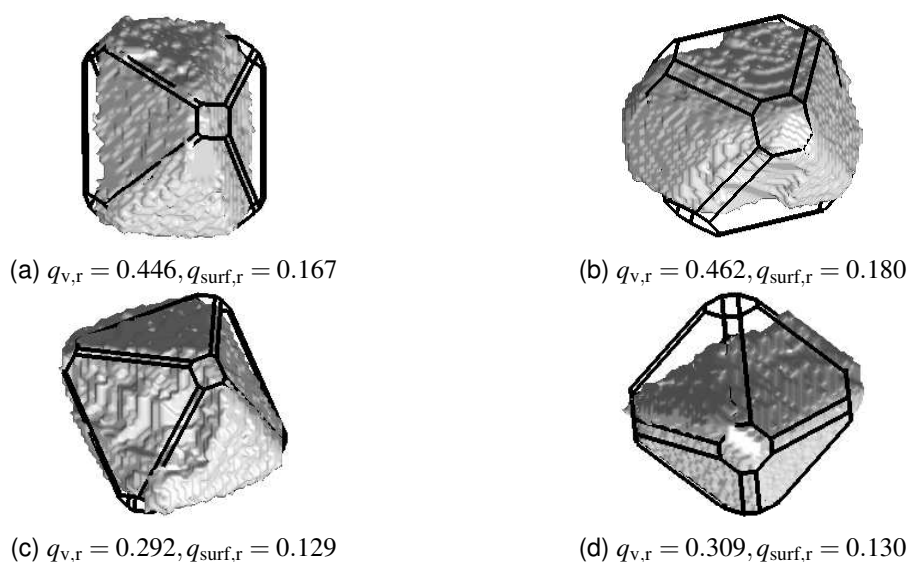


Figure 3.24: Shape identification in case of single crystals: examples where the underlying crystal did not correspond to a well-defined single particle.



Figure 3.25: Shape identification in case of an aggregate, obtained with the algorithm for single crystals (left) and with the algorithm for aggregates(right). Error measures regarding the right crystals are defined in chapter 4 on crystal aggregates.

Chapter 4

Shape Identification of Primary Particles in Crystal Aggregates

4.1 Algorithm

The previous chapter introduced a method for identifying the size and shape of single crystals from 3D images. Here, the method is extended towards analyzing crystal aggregates and is based on the corresponding publication [2]¹, while the results of the preliminary version of the method were presented at the ISIC19 conference [140]. The method contains two basic steps. First, an aggregate must be segmented into primary particles. This is described in section 4.1.2. The second step involves shape identification of each primary particle and is presented in section 4.1.2.1. The presented method is fully automatic, similarly as in the case of single crystals, presented in Chapter 3. Due to certain drawbacks of the obtained automatic result, the method can be further extended by allowing user interaction. This is presented in the next Chapter, along with the application for measuring the orientation between primary particles. The here presented, automatic method assumes that each particle is stored in a separate file. Particles may also represent single crystals, in which case they should result in only one fitted primary particle.

4.1.1 Pre-processing

Before proceeding with segmentation, the image containing an isolated particle is filtered. The image may represent a simulated or an experimentally measured particle. The particle is eroded using a structuring element defined by a matrix of length 3 voxels, with a foreground voxel in the middle and on positions corresponding to 6-neighbors of the middle voxel. Connected components in the eroded image are then identified and all but the largest one are deleted. Finally, a dilation with the same structuring element is performed. This procedure is similar to opening and ensures that surface irregularities are deleted, as shown in Figure 4.1.

¹In the previous publication, x - and y -axis of the crystals were permuted, which is corrected here. Further small bugs were fixed.



Figure 4.1: A crystal before (a) and after (b) the filtering procedure

Upon filtering, the method continues by isolating a volume point cloud, containing the coordinates of foreground voxels, as well as the surface point cloud, consisting of coordinates of foreground voxels that have at least one background voxel as a 6-neighbor.

4.1.2 Segmentation into Primary Particles

Segmenting a non-convex object into convex constituting parts is a problem statement often seen when working with images of biological cells [141, 142] and nuclei [95]. Cells and nuclei tend to cluster and overlap and often need to be separated before allowing their counting or further processing. Methods seen in the literature often rely on two basic ideas. The first group of methods is based on identifying concavity points, representing the points where objects are joined together [78, 142, 143, 144, 145]. In case of 2D images, objects can then be separated by connecting concavity points, while the approach can also be generalized to 3D confocal microscopy images by combining processed 2D slices [78]. The second group of methods relies on watershed transform, with a suitable set of pre- or post-processing steps [91, 92, 93]. These methods are directly applicable to 3D images. The basics of the watershed transform have been introduced in section 2.3.2.2. Hybrid approaches can also be seen in the literature, involving the use of concavity points to extract watershed markers [94, 141]. Separating touching particles is furthermore necessary when investigating agglomerates of spherical particles and can also be based on watershed transform [90].

The segmentation method presented in this section uses distance-transform based watershed transform, while the concavity points are used as indicators of correct segmentation in the subsequent merging procedure. It is summarized in Figure 4.2a and illustrated in Figure 4.3 on one example crystal.

4.1.2.1 Identifying Concavity Points

The concavity point identification procedure is based on the idea that a mask centered at some surface point will contain more foreground than background voxels if the point is concave. Similar approaches for the 2D images are seen in the works of Fernandez et al. [141] and Indhumathi et al. [78].

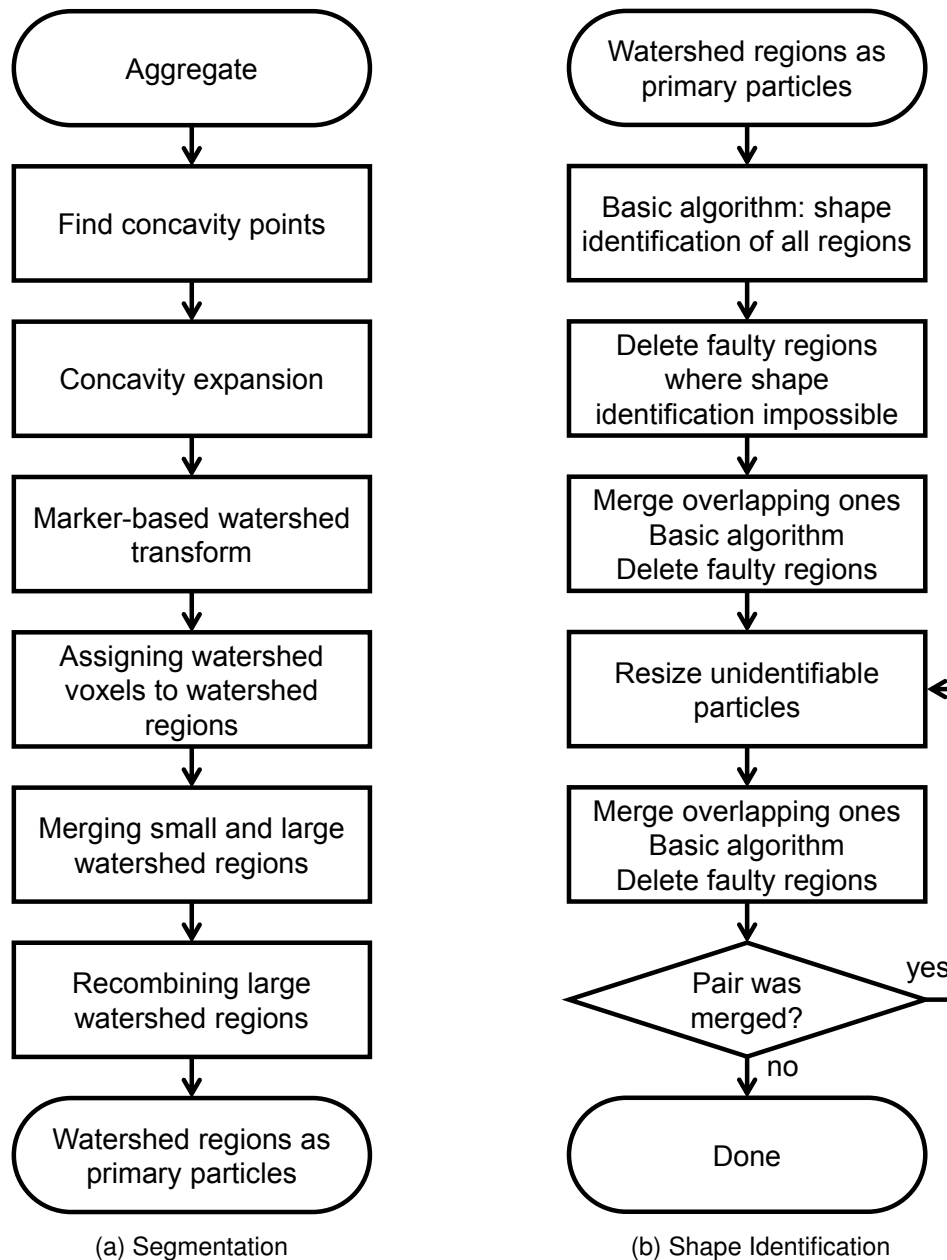


Figure 4.2: Identifying the shapes of primary particles in crystal aggregates [2].

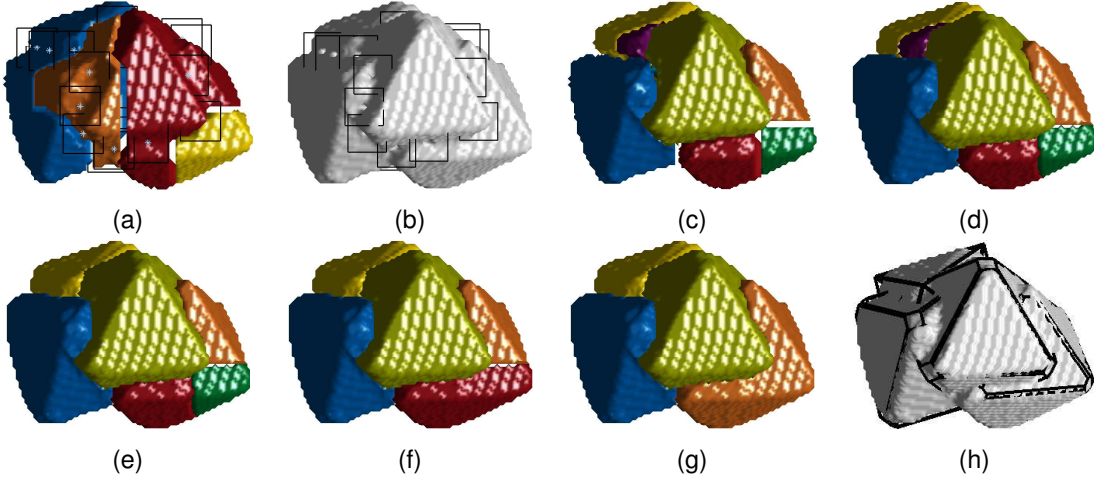


Figure 4.3: Segmentation of an aggregate into primary particles [2]. (a) Watershed segmentation before concavity expansion. (b) Concavity expansion. (c) Watershed segmentation after concavity expansion. (c) Watershed voxels are re-assigned. (d) Small and large watershed regions are merged. (e),(f) Two iterations of the procedure that recombines large watershed regions. (g) Shape identification. Concavity points are shown as blue crosses in Figure 4.3a, while the corresponding concavity search masks are presented as black boxes.

A cubic mask of length $2a + 1$ is centered at each surface point \mathbf{p} . A concavity of the point is computed as [2]:

$$c_{\mathbf{p}} = \frac{N_{\text{foreground mask}}}{N_{\text{mask}}}, \quad (4.1)$$

where $N_{\text{mask}} = (2a + 1)^3$ is the number of voxels in the mask and $N_{\text{foreground mask}}$ the number of foreground voxels within the mask. Concavity points are obtained by a peak search over these values. The peak search uses the sliding-window non-maximum suppression approach, similar to that from section 3.1.1. The idea of the sliding-window approach is that a point remains a maximum only if it is a maximum inside of each window that contains it. Here, we use a cubic mask of length $2b + 1$, where $b = \lfloor \frac{a}{2} \rfloor$. The sliding window checks only the surface points within the mask. Finally, the concavity points obtained from the sliding-window search are deleted if their concavity value $c_{\mathbf{p}}$ is lower than the threshold $1.2c_t$. The value c_t is obtained for a cubic object perfectly subdivided into voxels so that voxel sides are parallel to the object sides. A point on this surface, far away from the cube boundary, would have a concavity value equal to [2]:

$$c_t = \frac{(a + 1)(2a + 1)^2}{(2a + 1)^3} = \frac{a + 1}{2a + 1}. \quad (4.2)$$

The factor 1.2 imposes a stricter criterion to compensate for boundary roughness. The length of the mask is determined according to the aggregate size using [2]:

$$a = \lceil \sqrt[3]{0.0013N_{\text{voxels}}} \rceil, \quad (4.3)$$

where N_{voxels} is the total number of foreground voxels describing the aggregate. Obtained concavity

points are illustrated in Figure 4.3a by blue crosses, while the corresponding search masks are shown as black cubes.

4.1.2.2 Concavity Expansion

In section 2.3.2.2, the concept of distance-transform based watershed transform was introduced. This idea can be applied to split overlapping objects. For each foreground voxel, its distance to the background is computed and the sign of this value is inverted. A non-convex object would thus often have several local minima, corresponding to overlapping particles. However, this approach is not always successful and may lead to both under- and over-segmentation.

In order to prevent under-segmentation, concavity expansion is performed on the original image I . The procedure is illustrated in Figures 4.3b and 4.4. It is a simplified, 3D approach to the idea presented by Zhang et al. [94] for splitting clustered nuclei. The goal is to ensure that watershed segmentation results in a separation near concavity points. The concavity expansion procedure therefore removes foreground voxels near concavity points, in the direction of the concavity. The direction \mathbf{d} of the concavity is defined using the cubic mask of length $2a + 1$, as for finding concavity points. It is the direction of a vector which ends at the concavity point \mathbf{p}_c and starts at the point \mathbf{p}_b , representing the arithmetic mean of the coordinates of background points in the mask [2]:

$$\mathbf{d} = \frac{\mathbf{p}_c - \mathbf{p}_b}{\|\mathbf{p}_c - \mathbf{p}_b\|}. \quad (4.4)$$

The procedure now finds all points \mathbf{p} of the volume point cloud which fulfill [2]:

$$\|\mathbf{p} - \mathbf{p}_b\| \tan \alpha \leq 1, \quad (4.5)$$

where α is the angle between $\mathbf{p} - \mathbf{p}_b$ and \mathbf{d} . Voxels corresponding to these points are assigned to the background if they additionally fulfill [2]:

$$\|\mathbf{p} - \mathbf{p}_b\| \leq 0.25 \cdot \max_{\|\mathbf{p} - \mathbf{p}_b\| \tan \alpha \leq 1} \|\mathbf{p} - \mathbf{p}_b\|. \quad (4.6)$$

This ensures that the concavity expansion procedure does not split the object. The procedure results in image I_{exp} with thin holes through the object near concavity points, as illustrated in Figure 4.4b.

4.1.2.3 Marker-based Watershed Transform

A distance-transform based watershed segmentation is performed on the image I_{exp} resulting from concavity expansion. This procedure was discussed and illustrated in section 2.3.2.2 and is briefly repeated here. First, I_{exp} is inverted and the distance transform I_{distexp} of the resulting image is computed. The sign of the values in this image are changed to enforce minima instead of maxima in the gray value landscape. Furthermore, voxels that corresponded to background in I_{exp} are given the value $-\infty$ in the distance transform image to ensure that background is another watershed basin. This results in the



Figure 4.4: (a) Concavity expansion procedure deletes foreground voxels in the concavity direction, defined between the concavity point p_c (orange) and the middle point of the background part of the mask p_b (blue). The mask is shown in black. The procedure results in a thin hole through the object near each concavity point (b). Reprinted (adapted) with permission from [2]. Copyright (2016) American Chemical Society.

image $\tilde{I}_{\text{distexp}}$. Finally, watershed transform performed on $\tilde{I}_{\text{distexp}}$ results in the image I_{wsexp} and ends the distance-transform based watershed segmentation.

The image I_{wsexp} contains a set of differently-colored labels, each representing a separate region, as well as a set of watershed voxels that split the regions. However, voxels that were deleted in the concavity expansion procedure belong to the background region in this image. Thus, the image I_{wsexp} is used to define a mask I_{mask} for the distance-transform based watershed segmentation of the original image I . The mask contains a value 1 at the locations of the obtained foreground regions and a value 0 at all other locations. MATLAB function `imimposemin` is then used to impose minima in \hat{I}_{dist} at locations given by I_{mask} . The image \hat{I}_{dist} is obtained by distance-transforming I and inverting the sign of the result. The obtained image I_{distmin} is similar to the distance-transform image but contains minima only at the locations defined by I_{mask} . In order to ensure that the background is a separate region, voxels representing background in the original image I are set to $-\infty$ in I_{distmin} to get \tilde{I}_{dist} . Finally, a watershed transform of \tilde{I}_{dist} is performed, resulting in the image I_{ws} . In order to ensure no voxels are lost, all foreground voxels of I that were assigned to background in I_{ws} are set to watershed voxels, and will be assigned to the nearest region in the following step.

The result can be seen in Figure 4.3c on one example. The image clearly shows over-segmentation as there are more watershed regions, represented by different colors, than there are primary particles. However, this image represents an improvement with respect to the distance-transform based watershed procedure without using the concavity expansion, shown in Figure 4.3a. In that image, regions contain parts of several primary particles. Thus, it is not possible to correctly segment a primary particle by concatenating several regions.

4.1.2.4 Assigning Watershed Voxels to Watershed Regions

Watershed voxels are voxels that belong to neither region and therefore result in holes when only regions are visualized, as seen in Figure 4.3c. In order to avoid this, they are assigned to the nearest foreground region in an iterative procedure. The nearest region is simply the first found region that

contains a voxel on the distance 1 voxel from the watershed voxel. If no such region exists, a region with the distance $\sqrt{2}$ voxels is searched for, followed by the distance $\sqrt{3}$ voxels if this fails. Finally, if no neighbor can be found for the current watershed region, it remains to be assigned in one of the next iterations. The procedure is stopped when no watershed voxels remain and the result is illustrated in Figure 4.3d.

4.1.2.5 Merging Small and Large Watershed Regions

The concavity expansion resolved the under-segmentation problem. It remains to resolve the over-segmentation problem visible in Figure 4.3d, showing many regions instead of only four primary particles. Before proceeding with region merging, neighborhood relationships between regions are defined. Two regions are considered neighbors if they contain a pair of voxels that are 26-neighbors.

Here, small regions are treated separately and are merged with large regions, as was also done by Umesh Adiga and Chaudhuri [91]. A region is considered small if it contains fewer than 3% of the total number of voxels. It is assumed that such regions do not represent full primary particles. In each iteration, one small region i is considered. It is paired with each of the neighboring regions j and a concavity value

$$c_{i,j} = \left| 1 - \frac{N_i + N_j}{V_{i,j}} \right| \quad (4.7)$$

is computed. Here, N_i and N_j are the numbers of voxels in the regions i and j , while $V_{i,j}$ is the volume of the convex hull obtained when using voxel indices of the two regions as point coordinates. Region j leading to the smallest concavity value $c_{i,j}$ is chosen, upon which the regions are concatenated and the neighborhood relationships updated. If during this procedure several small regions are concatenated so that the resulting region is no longer small, it is deleted from the list of small regions and treated as a large region. The result of this procedure is shown in Figure 4.3e.

4.1.2.6 Recombining Large Watershed Regions

The final step in the segmentation procedure is merging of large regions based on their concavity, similarly as in [93]. Before proceeding, neighborhood relationships between regions are redefined. Two regions are now considered neighbors if they contain a pair of voxels on the aggregate surface that are 6-neighbors. This ensures that neighboring regions indeed touch on the crystal surface. Neighboring regions are then iteratively merged using the procedure described below.

Each iteration of the procedure starts by updating neighborhood relationships based on concavity points. Concavity points may nullify certain neighborhood relationships and thus inhibit a pair of regions from being merged. Each concavity point \mathbf{p} has a mask of influence C , covering all voxels inside a cube of length $2a + 1$, where a is defined in equation 4.3. If voxels in the mask belong to exactly two regions, their neighborhood relationship is nullified. In case of more than two regions in the mask, regions are

considered pairwise. For each region pair (i, j) , a concavity value inside the mask is computed by [2]:

$$c_{\mathbf{p},i,j} = \frac{N_{\text{voxels of } i \text{ in mask } C} + N_{\text{voxels of } j \text{ in mask } C}}{N_{\text{mask}}}, \quad (4.8)$$

where $N_{\text{voxels of } i \text{ in mask } C}$ and $N_{\text{voxels of } j \text{ in mask } C}$ are the number of voxels of regions i and j inside the mask C , containing the total of $N_{\text{mask}} = (2a + 1)^3$ voxels. The neighborhood relationship between i and j is deleted if $c_{\mathbf{p},i,j} < c_t$. Note that in case where only two voxels are present in the concavity mask, the threshold of $1.2c_t$ is implicitly used, as this is the threshold for concavity point detection. Furthermore, each pair of regions is considered several times, as there may exist several concavity points in their vicinity. If the nullification criterion is fulfilled for at least one concavity point, these regions will no longer be marked as neighbors and their merging will no longer be possible.

Upon establishing neighborhood relationships, all pairs of neighboring regions are investigated and their concavity is computed according to equation 4.7. The pair leading to the smallest concavity is merged and the neighborhood relationships are updated so that all neighbors of either region are now neighbors of the merged region.

Two iterations of the merging procedure are shown in Figures 4.3f and 4.3g. The procedure stops if there no longer exists a pair of neighbors which can be merged. In the case shown in Figure 4.3, the second iteration completes the segmentation.

4.1.3 Shape Identification of Primary Particles

The segmentation procedure results in primary particle candidates. It remains to perform a shape identification for each primary particle. Due to imperfections in the segmentation, the basic shape identification algorithm is followed by a set of post-processing steps which may further discard and concatenate regions or re-size the obtained shapes. The whole procedure is summarized in Figure 4.2b, and explained below in more details.

4.1.3.1 Basic Algorithm

The difference between a primary particle in an aggregate and a single crystal is that the primary particle is often incomplete, as its parts overlap with other particles in the aggregate. Therefore, not all primary particle faces are visible on the aggregate surface and the algorithm for shape identification of single crystals presented in chapter 3 must be adapted accordingly.

The algorithm is based on the surface point cloud of a primary particle, containing only aggregate surface points that belong to that primary particle.² It follows the scheme in Figure 3.1, with some modifications to each step, described below.

In the first step, face normals are identified by Hough transform, using the surface point cloud of the current primary particle. Due to its incompleteness and the imperfections in segmentation, a filtering of

²Similarly as in case of single crystals, points must be translated so that their mean coordinate coincides with the origin of the coordinate system, $(0,0,0)$, where the translated points are denoted by $\tilde{\mathbf{p}}_i$. Note that this translation is done for each primary particle separately, as the translation vector is different for each primary particle.

the Hough transform peaks is performed. A peak $\tilde{b}_{pks,i}$ in the reduced Hough space $\tilde{b}(\phi_i, \theta_i)$ is deleted if

$$\tilde{b}_{pks,i} < \frac{\frac{1}{N_{dir}} \sum_{k=1}^{dir} \tilde{b}_k}{\frac{1}{N_{pks}} \sum_{k=1}^{N_{pks}} \tilde{b}_{pks,k}} \max_j \tilde{b}_{pks,j}. \quad (4.9)$$

Here, $\tilde{b}_{pks,j}$ are the N_{pks} identified peaks and N_{dir} the number of considered spatial directions, representing the size of the reduced Hough space \tilde{b} . This filtering is the only difference regarding the procedure for identifying face normals.

In the second step, a matching between identified and model face normals is established and the rotation matrix \mathbf{R} between them is found. The procedure for single crystals, presented in Figure 3.5 is briefly summarized here. It is based on choosing a set of three best identified face normals and iteratively forming approximations of the rotation matrix by assigning these as correspondents for some three model face normals. An approximation \mathbf{R}_{cand} is allowed if the error criterion

$$L(\mathbf{R}_{cand}) < e_{tol} \quad (4.10)$$

is passed, where the objective function $L(\mathbf{R}_{cand})$ is defined in equation 3.15. The rotation matrix candidate is used to rotate the remaining model face normals and match them to the remaining identified face normals. In the algorithm for single crystals, the matching procedure was stopped if all identified face normals had a match. Otherwise, the case leading to a maximal number of matched face normals was chosen and \mathbf{R} was computed based on the chosen matching. The problem when applying this procedure to primary particles in crystal aggregates lies in the stopping criterion. It may happen that a primary particle is covered by other particles to such an extent that only three face normals were identified. In this case, the matching corresponding to the first rotation matrix \mathbf{R}_{cand} that passes the error criterion is accepted as all three identified face normals were matched. Thus, the procedure is modified as follows. It no longer stops if all identified face normals were matched. Furthermore, if only three face normals were identified, the matching leading to the smallest $L(\mathbf{R}_{cand})$ is chosen to obtain the final rotation matrix. Otherwise, the matching leading to the largest number of matched faces is still used, as in the procedure for single crystals.

The third step is the computation of face distances for each rotated model face normal \mathbf{Ra}_i . It is possible that a rotated face normal is not visible on the aggregate surface. Such a face is either too small, has grown out or is covered by another primary particle. In the first two cases, the face is considered as "existing" on the aggregate surface and its face distance can be determined. Otherwise, the face distance cannot be determined and must be obtained using symmetry. Therefore, it is first determined whether a given rotated model face exists on the aggregate surface. The adopted approach is tailored for the investigated potash alum crystals which contain three face groups $\{111\}$, $\{100\}$ and $\{110\}$, as shown in Figure 2.1. Experimentally observed crystals had pronounced $\{111\}$ faces, so that the $\{111\}$ is the principal face group. In this case, three rules can be derived to determine whether a

given rotated model face \mathbf{Ra}_i exists on the aggregate surface:

- All faces that have a match among identified faces are considered "existing".
- If a face of type $\{111\}$ has no match among the identified faces, it is considered as "non-existing".
- Members of other face groups are considered "existing" if all neighboring $\{111\}$ faces had a match among identified faces; otherwise, they are considered "non-existing".

In case of matched faces, face distance is determined as in the case of single crystals, which is described in section 3.1.3. However, the "non-existing" faces have no distance value and the resulting polytope may be open. For this reason, it is also not yet possible to create a non-redundant H -representation and ensure that faces indeed touch the obtained polytope. First, the remaining distance values must be obtained from the symmetry conditions. Thus, the existence mapping matrix \mathbf{S}_E is considered. This matrix has a size of $n_E \times n_H$. It contains a row with a value 1 at place i for each "existing" face i , while all other values are 0. Symmetry conditions are applied to existing faces, similarly as in the single-crystal case. First, the new origin of the symmetrical crystal is computed as ³:

$$\mathbf{x}_0 = -(\mathbf{M}_{C,E}\mathbf{S}_E\mathbf{A}\mathbf{R}^T)^+ \mathbf{M}_{C,E}\mathbf{S}_E\mathbf{h}, \quad (4.11)$$

where [2]:

$$\mathbf{M}_{C,E} = \mathbf{I} - (\mathbf{S}_E\mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}) \cdot (\mathbf{S}_E\mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}})^+. \quad (4.12)$$

The crystal is then moved using this new origin to obtain a crystal described by the vector $\mathbf{S}_E\mathbf{h}'$ of face distances [2]:

$$\mathbf{S}_E\mathbf{h}' = \mathbf{S}_E\mathbf{h} + \mathbf{S}_E\mathbf{A}\mathbf{R}^T\mathbf{x}_0. \quad (4.13)$$

In this crystal, all faces of a face group have the same face distance, but some faces are missing and the crystal is open. Thus, the face distance of each face group can be obtained by mapping into the \mathbf{h}_C space, using only the existent faces:

$$\mathbf{h}_C = (\mathbf{S}_E\mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}})^+ \mathbf{S}_E\mathbf{h}'. \quad (4.14)$$

This will result in a value 0 for face groups where no face was identified as "existing". In this case, the face distance is set to some large value, so that the faces are grown-out, and the corresponding \mathbf{h} vector is identified:

$$\mathbf{h} = \mathbf{M}_{\mathbf{h}_C \rightarrow \mathbf{h}}\mathbf{h}_C. \quad (4.15)$$

This crystal is complete and symmetrical, but may contain grown-out faces. The subsequent step identifies all grown-out faces by forming an H -representation containing only non-redundant face normals. This reduced H -representation is converted into the V -representation, consisting of vertices \mathbf{v}_k . Face

³Note that the minus sign is missing in the corresponding publication [2]

distances for the grown-out face groups are then mapped back so they touch the polytope using:

$$h_i = \max_k \langle \mathbf{v}_k, \mathbf{R}\mathbf{a}_i \rangle, \quad (4.16)$$

where i is the first face of the group i . As a valid face distance for each face group is obtained, a mapping of this H_C -representation to the H - and V -representation, containing the vertices $\tilde{\mathbf{v}}_i$ completes the shape identification procedure. The obtained symmetrical crystal is centered at $(0,0,0)$, so that the final vertices \mathbf{v}_i of the polytope are obtained by applying the equation 3.31. This final crystal is centered at \mathbf{x}_{orig} , defined in equation 3.32.

4.1.3.2 Post-processing of Non-ideal Cases

The described shape identification procedure may fail to provide a good shape or may provide a shape of inadequate size, if only very few face normals could be identified. Furthermore, the segmentation procedure may still split a crystal into two or more particles, so that additional merging should be done. Therefore, the basic algorithm defined above is followed by a set of post-processing steps, as shown in Figure 4.2b. They are explained in more detail below.

4.1.3.3 Deleting Faulty Regions

Faulty regions represent regions where no shape identification was possible. This may occur if fewer than three adequate face normals were identified to find the rotation matrix. Furthermore, an empty crystal is returned if the largest number of identified face normals belongs to a face group other than the principal, $\{111\}$ group, or the faces of the $\{111\}$ group are found to be grown-out.

Points belonging to these faulty regions are deleted from both the surface and the volume point cloud. An example of such a region is shown in Figure 4.5.

4.1.3.4 Merging Overlapping Particles

The automatic segmentation procedure described in section 4.1.2 may still be imperfect and split a primary particle. In this case, the shape identification procedure should provide almost identical particles.



Figure 4.5: Faulty region, presented in orange in the left image is identified and deleted, resulting in the right image. Concavity points are shown in red in the left image.

Therefore, pairs of highly overlapping polytopes are identified and the corresponding watershed regions are merged. The merged regions are then processed by the basic shape identification algorithm. The pair (i, j) of identified particles is considered highly-overlapping if the degree of overlap with respect to i or j is higher than 0.9. The degree of overlap with respect to particle i is computed as

$$o_i = \frac{V_{\text{overlap},i,j}}{V_i}, \quad (4.17)$$

where $V_{\text{overlap},i,j}$ is the volume of the polytope obtained as the intersection of identified polytopes for particles i and j , while V_i is the volume of the i -th particle polytope. The degree of overlap with j is computed analogously. The result of this procedure is illustrated in Figure 4.6.

4.1.3.5 Moving and Re-sizing the Identified Shape

In certain cases, the size of the primary particle cannot be identified with certainty, although the shape can. For example, one can determine the orientation of an octahedron using normals to four faces that intersect in the common vertex. However, it is not possible to know the size of the faces in that case. A potash alum particle is declared to have an identifiable size if there exists a matched $\{111\}$ face whose adjacent $\{111\}$ faces were also matched. Faces are considered adjacent if they share an edge. If the size is not identifiable, the particle polytope is re-sized and the origin is moved, accordingly. The direction \mathbf{d}_M of origin movement is defined using the $n_{m,\text{oct}}$ matched rotated model $\{111\}$ faces, \mathbf{Ra}_i

$$\mathbf{d}_M = \frac{-\sum_{i=1}^{n_{m,\text{oct}}} \mathbf{Ra}_i}{\left\| \sum_{i=1}^{n_{m,\text{oct}}} \mathbf{Ra}_i \right\|}. \quad (4.18)$$

The sign of this vector is inverted if it is determined that an improvement is obtained when moving the crystal in the opposite direction.

The best crystal size and the best corresponding origin are found by an iterative procedure. At each iteration i , the origin of the crystal is moved for the value \mathbf{d}_M :

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{d}_M, \quad (4.19)$$

where $\mathbf{x}_0 = \mathbf{x}_{\text{orig}}$. Then, the best crystal size is computed using a bisection method. Using the best obtained crystal size, the projection error value

$$q_p = \frac{1}{N_{\text{surf}}} \left\| \mathbf{p}_i - \mathbf{p}_{i,\text{proj}} \right\| \quad (4.20)$$

is computed. Here, the N_{surf} points \mathbf{p}_i , describing the surface point cloud of the considered primary particle, are projected onto the best-sized crystal to get $\mathbf{p}_{i,\text{proj}}$. The search is stopped when q_p no longer reduces.

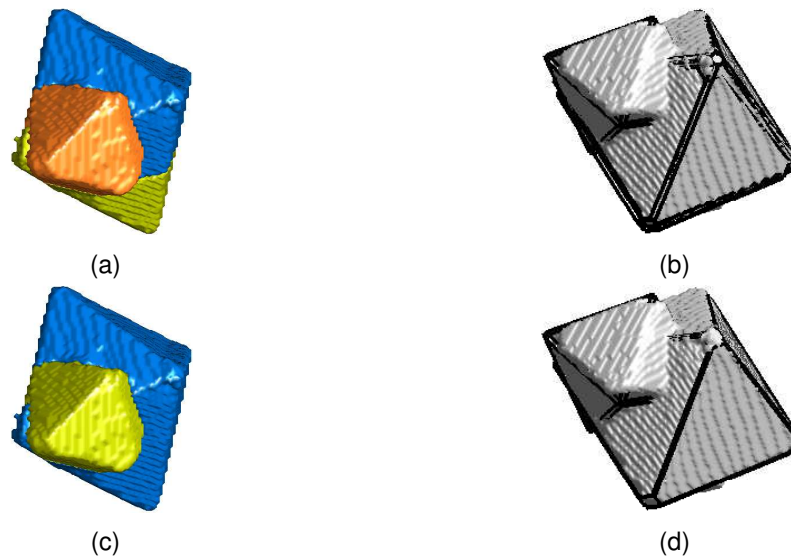


Figure 4.6: The blue and green regions (upper left) belong to the same primary particle, so that the identified polytopes overlap strongly (upper right). The algorithm detects such situations and merges the regions (lower left) to obtain only one identified polytope (lower right).

The bisection method for finding the best-sized crystal considers in each iteration a smaller \mathbf{h}_S and a larger \mathbf{h}_L version of the current crystal. For these computations, surface points \mathbf{p} and the crystal must be translated so that the crystal center is in the coordinate system origin. At the beginning, a smaller version is chosen so that no point of the surface point cloud is inside this crystal. A larger crystal is chosen so that all surface points are inside it. In each iteration, a new crystal is defined as $\mathbf{h}_{\text{new}} = 0.5(\mathbf{h}_L + \mathbf{h}_S)$. If there are more surface points outside than inside of \mathbf{h}_{new} , \mathbf{h}_{new} will be used as the "smaller" boundary in the next iteration. Otherwise, it becomes the "larger" boundary. The search is ended when the following is fulfilled:

$$|\mathbf{h}_L(1) - \mathbf{h}_S(1)| < 0.1. \quad (4.21)$$

In case where crystal size is truly not identifiable, this approach would give one of the sizes that provide the exact same fit quality. However, experimentally observed crystals often show small faces or face parts that were not matched but allow one to compute the crystal size. The quality criterion in equation 4.20 would ensure that the best fit covers these faces.

One result of the procedure for moving and re-sizing of a crystal polytope is given in Figure 4.7. This procedure can introduce highly overlapping shapes, discussed in the previous section. In such a case, post-processing steps are repeated, as illustrated in Figure 4.2b. Upon reaching a state where no primary particles are merged, the shape identification procedure is ended.



Figure 4.7: Before (a) and after (b) moving and re-sizing the smaller primary particle shape fit.

4.1.4 Fit Quality Measures

As in the case of single crystals, it is necessary to estimate the quality of fitted shapes when working with experimental data as well as to evaluate the procedure by comparing the fitted polytopes to the simulated ones.

4.1.4.1 Simulated Data

Before computing the quality measures for the simulated data, it is necessary to find a match among identified polytopes for each simulated primary particle in an aggregate. This is done by choosing that identified polytope whose centroid is closest to the one of the simulated polytope. It is furthermore ensured that no identified polytope is matched to more than one simulated polytope.

A volume error is defined with respect to the i -th primary particle, similarly as in the case of single crystals

$$q_{v,s,i} = \frac{|V_{s,i} - V_{f,i}|}{V_{s,i}}. \quad (4.22)$$

Here, $V_{s,i}$ and $V_{f,i}$ are the volumes of the i -th simulated and corresponding identified polytope, respectively. An overlap error is defined as

$$q_{o,s,i} = 1 - \frac{V_{o,i}}{V_{s,i}}, \quad (4.23)$$

where $V_{o,i}$ is the volume of the intersection of the i -th simulated polytope and the corresponding identified polytope. Finally, a disorientation angle between the simulated and the corresponding identified polytope is computed in a manner that will be explained in chapter 5. Disorientation angle represents a new error measure and was not used in the corresponding publication [2].

An aggregate was considered to have a well-identified shape if the shape identification was possible for each primary particle, the quality measures $q_{v,s,i}$ and $q_{o,s,i}$ were below 0.15 for each primary particle and the disorientation angle for each primary particle was below 10° . If there were fewer identified than simulated particles, but the given quality criteria were fulfilled for the identified particles, the aggregate is declared to contain missing primary particles. Finally, if at least one of the three criteria based on fit quality measures is not fulfilled for some primary particle, the shape identification for that aggregate is considered unsuccessful. Shape identification is also declared unsuccessful if there were more

identified than simulated polytopes. Furthermore, if a simulated primary particle is overlapped by 80 % by another, it is removed from consideration as its shape identification is assumed impossible.

4.1.4.2 Experimental Data

The volume-based quality measure for the experimental data, $q_{v,r}$ retains the same form as in case of single crystals (eq. 3.35):

$$q_{v,r} = \frac{N_{\text{outside}}}{N_{\text{voxels}}} + \frac{V_{\text{agg}} - N_{\text{inside}}}{V_{\text{voxels}}}, \quad (4.24)$$

with the difference that V_{agg} is now the volume of the whole aggregate. It is computed using:

$$V_{\text{agg}} = \sum_{i=1}^{n_p} V_i - \sum_{i_1, i_2 > i_1}^{n_p} V_{i_1, i_2} + \dots + (-1)^{(n_p-1)} \sum_{i_1, \dots, i_{n_p} > i_{n_p-1}}^{n_p} V_{i_1, \dots, i_{n_p}}, \quad (4.25)$$

where n_p is the number of identified polytopes and V_{i_1, \dots, i_k} is the intersection of polytopes i_1, i_2, \dots, i_k .

The quality measure $q_{\text{surf},r}$ is also extended to the case of n_p primary particles [2]:

$$q_{\text{surf},r,\text{all}} = \frac{\sqrt{\frac{1}{N_{\text{surf}}}} \sum_{j=1}^{N_{\text{surf}}} \min_i \|\mathbf{p}_j - \mathbf{p}_{j,\text{proj},i}\|^2}{\max_i h_{\text{princ},i}}. \quad (4.26)$$

Here, each point \mathbf{p}_j of the aggregate surface point cloud is projected to each primary particle i to get $\mathbf{p}_{j,\text{proj},i}$, where the minimal such distance is finally considered. The scaling value is the largest obtained octahedral face distance, when considering all primary particles and polytopes centered at the coordinate system origin.

Cases with faulty regions can lead to high error values. In order to distinguish them from cases containing badly fitted shapes, a quality measure that only considers the $N_{\text{surf},\text{fit}}$ surface points that remain after deleting the faulty regions, is defined similarly [2]:

$$q_{\text{surf},r,\text{fit}} = \frac{\sqrt{\frac{1}{N_{\text{surf},\text{fit}}}} \sum_{j=1}^{N_{\text{surf},\text{fit}}} \min_i \|\mathbf{p}_j - \mathbf{p}_{j,\text{proj},i}\|^2}{\max_i h_{\text{princ},i}}. \quad (4.27)$$

The shape identification of an aggregate is considered successful with respect to these quality measures if $q_{\text{surf},r,\text{all}} < 0.15$ and $q_{v,r} < 0.2$. If this is not fulfilled, but $q_{\text{surf},r,\text{fit}} < 0.15$ and $q_{\text{surf},r,\text{fit}} < q_{\text{surf},r,\text{all}}$, the aggregate is considered to contain primary particles where the shape identification was impossible. Finally, in all other cases, the shape identification is declared unsuccessful.

4.2 Validation

4.2.1 Simulated Data

4.2.1.1 Simulation Procedure

Similarly as in the case of single crystals, samples of crystal aggregates were simulated. Three samples were derived from one basic set of 100 aggregates. In the basic set, there are 58 aggregates with two primary particles, 13 with three, 21 containing four and 8 aggregates with five primary particles. The sizes of the primary particles are drawn from a normal distribution with a mean of $250\ \mu\text{m}$ and a standard deviation of $30\ \mu\text{m}$. An aggregate of two primary particles is obtained by first creating two primary particles with the desired size which are centered at the origin. One of the primary particles is then moved in a random direction and a position where it lightly touches the other particle is determined by bisection. An aggregate of more than two primary particles is obtained by adding primary particles successively so that they lightly touch the existing aggregate. Primary particle orientations are drawn according to the uniform sampling approach described in section 3.2.1.1. Three data sets are obtained by growing each primary particle for $50\ \mu\text{m}$, $150\ \mu\text{m}$ and $300\ \mu\text{m}$, respectively. The result of the growth procedure for one aggregate can be seen in Figure 4.8.

Upon obtaining a set of aggregates, 3D images are simulated. The used resolution, along with the other parameters, is given in Table 4.1. The procedure for generating 3D images is the generalized version of the one described for single crystals, using the minimal and maximal coordinates over all primary particles instead the coordinates of one particle [2]. The obtained aggregates differ from those in the corresponding publication [2] as the uniform sampling of rotation matrices, explained in section 3.2.1.1 was used here.

4.2.1.2 Surface Roughness

In addition to the perfect crystals described above, images of crystals containing surface roughness were created in order to investigate its influence on the performance of the algorithm. The amount of surface roughness is controlled by the parameter $p_{\text{roughness}}$ which can take a value of 0, 0.2, 0.4 or 0.6,

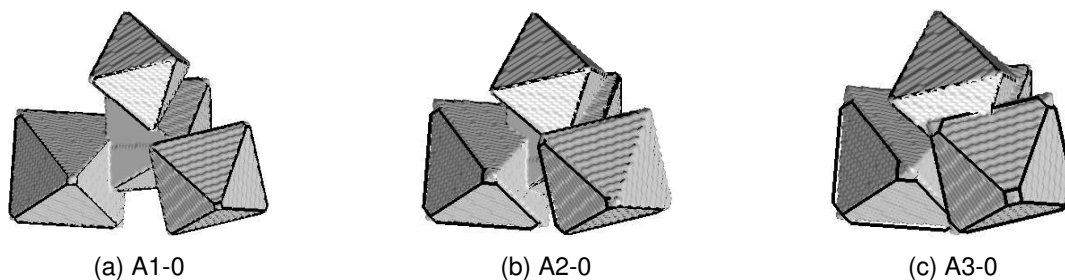


Figure 4.8: An illustration of the effect of primary particle growth on one aggregate from data sets A1-0, A2-0 and A3-0, defined in Table 4.1.

Case	No. model faces	Growth [μm]	$N_{\text{rho}} / \Delta\rho$	c_{min}	c_{max}	Voxel length [μm]	N_{side}	e_{tol}	$\alpha_{\text{tol}} [^\circ]$
A1- p	26	50	$N_{\text{rho}} = 40$	1	1	15	20	0.02	10
A2- p	26	100	$N_{\text{rho}} = 40$	1	1	20	20	0.02	10
A3- p	26	150	$N_{\text{rho}} = 40$	1	1	28	20	0.02	10

Table 4.1: Parameters for simulation and shape identification of aggregate data sets. p represents the surface roughness parameter $p_{\text{roughness}}$. There were four different surface roughness levels for each data set, so that 12 data sets are finally obtained.

where 0 corresponds to the case with no surface roughness described above. In order to add surface roughness, the face distances \mathbf{h} of each primary particle are enlarged by 10 %, resulting in a larger version of the particle, denoted by \mathbf{h}_L . They are also reduced by 10 %, resulting in a smaller version of the particle, \mathbf{h}_S .⁴ The voxels in the space between \mathbf{h}_L and \mathbf{h}_S can change their value. Voxels in the "upper" space, covering the space between \mathbf{h} and \mathbf{h}_L , were previously set to background as they are not inside the original particle \mathbf{h} . A voxel in this space, if connected to more than 9 foreground voxels, can change its value to foreground with the probability given by $p_{\text{roughness}}$. Similarly, voxels in the space between the original particle \mathbf{h} and the smaller version \mathbf{h}_S were originally assigned to the foreground as they are smaller than the original particle. A voxel in this space, if connected to fewer than 9 foreground voxels, can change its value to background with the probability $p_{\text{roughness}}$. Finally, an opening with an 18-connected structuring element is performed to obtain the final 3D image of the crystal with surface roughness.

The effect of surface roughness can be seen in Figure 4.9 on the case of one aggregate from data sets of type A2.

⁴Note that increasing and decreasing face distances \mathbf{h} is performed for the polytope centered at the origin of the coordinate system. The actual polytope, whose center is elsewhere, must be translated to perform these computations and then translated back to its actual position to obtain \mathbf{h}_L and \mathbf{h}_S .

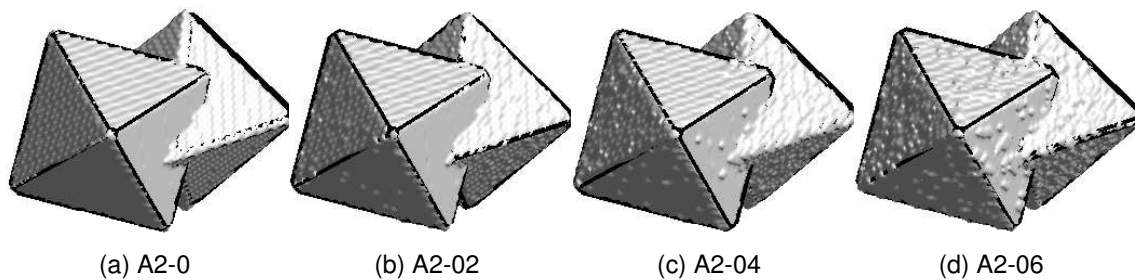


Figure 4.9: An illustration of the effect of surface roughness on one crystal from different data sets of type A2, defined in Table 4.1.

4.2.1.3 Results

Each identified aggregate was evaluated based on both the fit quality measured for simulated data and the fit quality measures for experimental data. Both sets of fit quality measures declare the shape identification of an aggregate either as successful, unsuccessful, or containing particles where the shape could not be identified, as explained in section 4.1.4. The results regarding both sets of quality measures are given in Table 4.2. Regarding the comparison with the simulated polytopes, it can be seen that the shape identification was successful for between 84 % and 95% of aggregates. This is similar to the result presented in [2], where the primary particles had different orientations. Furthermore, similar results are obtained when changing the parameter for the discretization of distances in Hough transform from $N_\rho = 40$ to a fixed $\Delta\rho = 1$.

Examples of successful shape identification are given in Figures 4.8 and 4.9. It can be seen that small $\{100\}$ and $\{110\}$ faces were identified, even though the simulated crystals were perfect octahedra. This is an artifact of the algorithm for shape identification and was investigated by simulation of single crystals in section 3.2.1.3. Similarly as in the previous study [2], the number of successfully fitted aggregates decreases with the amount of primary particles in an aggregate, as presented in Table 4.3. The number of successful shape identifications also decreases with the increasing particle overlap, controlled by the amount of growth of the h -distance. High overlap between particles means that a smaller portion of the particle is available for shape identification. Contrary to these parameters, surface roughness does not have a large influence on the success of the procedure, while it may influence the success of the scheme for individual aggregates, as seen in Table 4.2.

Unsuccessful shape identification is often caused by the fact that two regions were wrongly concatenated. An example can be seen in Figure 4.10. This may happen if more than two regions are in vicinity of a concavity point so that some regions may be concatenated. Furthermore, it can occur if a concavity point is missing. Concavity points are sometimes not identified if two crystals touch lightly as they will not produce enough concavity voxels in the corresponding search mask, used to define the concavity value by equation 4.1. In case of experimental data, a procedure for splitting an aggregate into agglomerates with truly overlapping particles will be introduced in order to resolve this issue. However, the particle may still overlap to a high degree with other particles, so that the problem persists. Concavity points may furthermore be absent if highly grown-in particles have a similar orientation, as seen in Figure 4.11.

A wrong segmentation can also split a primary particle in two regions, as illustrated above in Figure 4.6. An unsuccessful shape identification may result if this problem is not identified, unlike in Figure 4.6. Finally, an unsuccessful shape identification may be achieved even if the segmentation is correct or almost correct. This is illustrated in Figure 4.12. It is more likely to occur if particles strongly overlap, as a smaller part of the particle is available for shape identification.

Wrongly concatenating a pair of regions may also cause an aggregate to be declared as containing a primary particle where no shape identification is possible. In Figure 4.11, all identified shapes agree well with the simulated polytopes in spite of the wrong segmentation.

Simulation param.			Quality measures - sim. data			Quality measures - exp. data		
Sim. case tab 4.1	Growth h dist.	Surface rough. $p_{\text{roughness}}$	[%] Agg. all well identified	[%] Agg. with badly identified	[%] Agg. with missing	[%] Agg. all well identified	[%] Agg. with badly identified	[%] Agg. with missing
A1-0	50	0	91	7	2	93	5	2
A1-0.2	50	0.2	93	6	1	94	6	0
A1-0.4	50	0.4	94	6	0	94	6	0
A1-0.6	50	0.6	95	4	1	95	4	1
A2-0	150	0	92	8	0	95	5	0
A2-0.2	150	0.2	92	5	3	95	5	0
A2-0.4	150	0.4	91	8	1	93	7	0
A2-0.6	150	0.6	92	7	1	96	4	0
A3-0	300	0	86	12	2	94	5	1
A3-0.2	300	0.2	84	12	4	96	4	0
A3-0.4	300	0.4	84	12	4	97	3	0
A3-0.6	300	0.6	85	11	4	95	4	1

Table 4.2: Shape-identification success evaluated for the 12 simulated data sets using the quality measures for simulated data, defined in Section 4.1.4.1, and the quality measures for experimental data, defined in Section 4.1.4.2. First three columns show the simulated case and the parameters used in sample generation. The fourth and seventh columns represent the percentage of aggregates where the shape of all primary particles was correctly identified. The fifth and eighth columns represent the percentage of aggregates containing at least one badly identified primary particle. The sixth and ninth columns show the percentage of aggregates where the shape of some primary particles could not be identified, while it was identified correctly for the remaining particles.

Data sets	Number of primary particles			
	2	3	4	5
A1	99.57	92.31	86.90	65.63
A2	99.57	94.23	80.95	59.38
A3	99.57	96.15	61.90	18.75

Table 4.3: Percentage of successful shape identifications per aggregate class consisting of two to five primary particles. The values are computed for each group of simulated data sets by considering all four data sets obtained by varying the surface roughness parameter.

As seen when comparing the fifth and eighth column of Table 4.2, fit quality measures based on real data underestimate the number of unsuccessful shape identifications, especially for high particle overlap. The reason is that these measures are defined for the aggregate as a whole instead of for each primary particle. Thus, disagreements within one primary particle do not increase the overall error enough, as seen in the quality measures for the aggregate in Figure 4.12. Also, these quality measures are less likely to identify errors regarding primary particles that contribute little to the overall aggregate volume. Comparing the sixth and ninth column of Table 4.2 reveals that the number of aggregates where no shape identification was possible for some primary particles is also underestimated by the quality measures regarding real data. In cases such as the one illustrated in Figure 4.11, no region where shape identification failed exists. Therefore, the aggregate will be classified as having either a successful or unsuccessful shape identification by the quality measures developed for real data. Contrary, quality measures for simulated data will acknowledge that too few polytopes were identified. Depending on the agreement of the identified polytopes with the simulated ones, the aggregate will be considered to have either an unsuccessful shape identification or contain primary particles where no shape identification was possible.

4.2.2 Experimental Data

4.2.2.1 Preparation and Measurement

Potash alum crystals used for the validation of the aggregate shape identification procedure were produced by seeded batch cooling crystallization. A set of aggregates was chosen by visual inspection of the crystals. Particles were manually dispersed over a microscope slide, ensuring no touching between the particles. The slide was then covered by adhesive tape to obtain the dispersed crystals. The tape was rolled in and placed in an 0.65 ml micro test-tube. The middle part of the tube was imaged by a ZEISS X-Radia Versa XRM-500 μ CT machine with a tungsten target by Jonathan Schock, a PhD Candidate at the Chair of Biomedical Physics, TUM. Parameters of the measurement and the shape identification procedure are given in Tables 4.4 and 4.5. CT reconstruction was performed by a proprietary software XRM Reconstructor, v. 10.7.3245, at the Chair of Biomedical Physics, TUM. The



Figure 4.10: Segmentation (left) and shape identification (right) for one aggregate from sample A1-0. Two green primary particles were wrongly concatenated. The aggregate is considered to have an unsuccessful shape identification by both sets of quality measures.



Figure 4.11: Segmentation (left) and shape identification (right) for one aggregate from sample A3-0.2. The orange primary particle was wrongly concatenated with the part of another particle. However, the shape of the orange particle was identified correctly in spite of this segmentation error. The aggregate is considered to contain primary particles where no shape identification was possible by the quality measures developed for simulated data. It is considered to have a successful shape identification by the quality measures for real data.



Figure 4.12: Segmentation (left) and shape identification (right) for one aggregate from sample A3-0.4. The segmentation is almost correct, but a part of the orange particle was assigned to the blue one. However, the shape of the blue particle was identified as too small. The aggregate is considered to have an unsuccessful shape identification by the quality measures for simulated data. The shape identification is considered successful by the quality measures for real data, as $q_{v,r} = 0.088$ and $q_{surf,r,all} = 0.05$.

obtained 3D volume, as well as one vertical slice through it, are shown in Figure 4.13.

Time needed for sample preparation and measurement is in the order of a few hours for this data set.

4.2.2.2 Isolating Aggregates

As visible in Figure 4.13b, crystals have significantly higher gray values than the adhesive tape structures. Therefore, a simple thresholding procedure was performed to obtain the foreground crystal voxels. The threshold was determined manually by observing its influence on different slices using the software MAVI [96]. Upon binarization, aggregates were isolated by labeling connected components in MAVI. Aggregates may still contain lightly touching particles, which are assumed to have been created during transport and the simple preparation procedure. As the identification of concavity points may encounter problems at the presence of lightly touching aggregates, aggregates must be split into sub-aggregates that contain highly overlapping particles, or may represent single crystals. This procedure was performed in MATLAB and is described below.

An image I of each aggregate is first eroded with a cube structuring element whose length is 15 voxels. This will ensure the separation, but also highly shrink the particles. A labeling of connected components in the eroded image I_E of the aggregate is then performed to get the image I_L . If I_L consists of exactly one connected component, the aggregate does not contain any loosely-bound particles and is stored for further processing. Otherwise, foreground voxels of I_L are used to create a mask image and impose minima for the distance-transform-based watershed-transform of the aggregate image. This procedure is the same as that described in section 4.1.2.2 regarding concavity expansion. Watershed procedure will result in a set of regions describing sub-aggregates with overlapping particles, the background and a set of watershed voxels that are equally likely to belong to either region. Any voxels that belonged to the crystal foreground but were assigned to background by the watershed transform are also considered as watershed voxels. Watershed voxels are then re-assigned to the closest foreground region using the procedure from section 4.1.2.4. Finally, each so obtained sub-aggregate is written

Parameter	Value
Tube Voltage [kV]	60
Power [W]	5
Exposure Time [s]	6
Magnification	0.39x
Source-to-sample distance [mm]	50
Detector-to-sample distance [mm]	200
No. projections	1601
Voxel length [μm]	13.6
Detector pixel binning	2 x 2

Table 4.4: Parameters of the μCT measurement for validation of the aggregate shape identification procedure.

Case	No. model faces	$N_{\text{rho}} / \Delta\rho$	c_{min}	c_{max}	Voxel length [μm]	N_{side}	e_{tol}	$\alpha_{\text{tol}} [^\circ]$
Val-A	26	$N_{\text{rho}} = 40$	1	1	13.6	20	0.02	10

Table 4.5: Parameters for shape identification of experimental data: validation of the aggregate procedure.

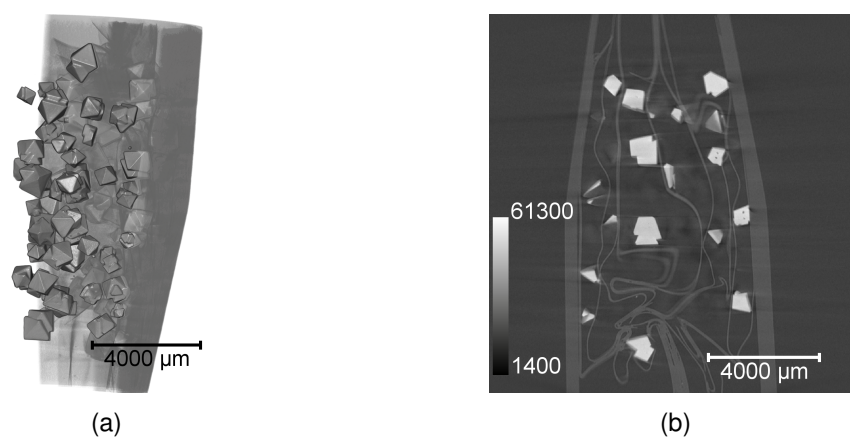


Figure 4.13: 3D volume of the particles used for validating the aggregate shape identification procedure (a), as well as one slice through it (b). Rendering was performed by Jonathan Schock, at the Chair of Biomedical Physics, TUM, using AVIZO Fire [146]. Reprinted with permission from [2]. Copyright (2016) American Chemical Society.

into a separate file and processed with the procedure presented in this chapter. Crystals containing fewer than 10000 foreground voxels were ignored. The obtained data set consisted of 85 crystals, representing highly-overlapping aggregates or single crystals.

Time requirements regarding aggregate isolation are in order of a few minutes for this data set.

4.2.2.3 Results

As no information about the true size and shape of these crystals is available, the quality of the shape identification procedure was evaluated using the measures and criteria for experimental data, presented in section 4.1.4.2. There were 65 aggregates with well-identified shape. Some examples are shown in Figure 4.14. It can be seen that the primary particle pairs can have both the same as well as different orientations. Furthermore, primary particles may differ in size. The last row shows examples of aggregates that were judged to have a successful shape identification, even though some issues are present. The large particle in Figure 4.14j was erroneously split into two primary particles. The algorithm failed to detect this and concatenate them back into one particle, so that two polytopes for the same particle were identified. Figure 4.14k shows an undetected small particle on the surface of the large particle. Even if this particle were to be segmented as such, it would probably not be possible to identify its shape. Finally, Figure 4.14l contains a primary particle where no shape identification was possible. However, as this particle is highly grown into one of the other particles, the error measure $q_{\text{surf},r,\text{all}}$ is small enough to allow the shape identification for the whole aggregate as successful. Note that in the previous program version, corresponding to the publication [2], the third primary particle was identified, but its size was over-estimated.

Figure 4.15 shows 6 out of 10 particles where the aggregate is considered to contain particles where no shape identification was possible. In Figure 4.15e, apart from failing to identify the shape of some particles, the size of the smaller particle was over-estimated. In Figure 4.15f, the larger particle was erroneously split into two parts, one of which was small and did not contain enough information to identify the shape. This, combined with the value $q_{v,r} > 0.2$, which is assumed to be due to surface roughness and asymmetry, lead to this aggregate erroneously being classified as containing primary particles where no shape identification was possible.

Finally, Figure 4.16 shows 6 out of 10 particles where the aggregate shape identification is considered unsuccessful. In Figure 4.16a it is difficult to decide whether the large particle is an aggregate or simply asymmetrical. A fit to the symmetrical crystal model in case of an asymmetrical particle would cause large error measures, thus classifying the shape identification as unsuccessful, which is further assumed to occur in Figure 4.16c. The aggregate in Figure 4.16b is also difficult to segment into primary particles so that the large particle is split into two parts, one of which is further concatenated with one of the smaller particles. The particle in Figure 4.16d was split into several parts due to surface roughness. It was not possible to identify the shape of the smaller parts. Furthermore, the surface roughness and the attached segments caused the error measures to exceed the thresholds. The particles in Figure 4.16e were re-concatenated into one particle, thus failing to report a missing shape identification for the

smaller particle. Errors in segmentation also caused the cubic particle in Figure 4.16f to be identified.

As already noted above, some changes with respect to the previous program version, leading to results published in [2] were observed, causing some particles to change their final class. However, the overall results remain the same.

4.2.2.4 Execution Time

The sequential execution time on a desktop computer with 16 GB RAM and Intel Core(TM)i5-3470 CPU @ 3.20 GHz running MATLAB 2015b on a 64-bit Windows 8.1 Enterprise was about 1.5 hours, leading to an average execution time of somewhat over one minute per aggregate. It is furthermore possible to run the computations in parallel using MATLAB parallelization toolbox. In the corresponding publication [2] using an older version of the program, a total execution time of 33 minutes for a parallel computation using four cores was reported.

4.2.3 Conclusion

It can be concluded that the presented automatic shape identification procedure for primary particles of crystal aggregates provides a reasonable shape in most cases. Problems are encountered if there is a large number of primary particles, primary particles have a very high level of overlap or the crystal is not symmetrical. Furthermore, lightly touching aggregates were separated into sub-aggregates with highly overlapping primary particles in order to avoid segmentation issues. Therefore, further improvements are necessary before the procedure can be applied to measuring the relative orientation between primary particles. This should involve a better sampling and preparation procedure, ensuring statistical relevance and a better crystal separation. It is presented in the following chapter.

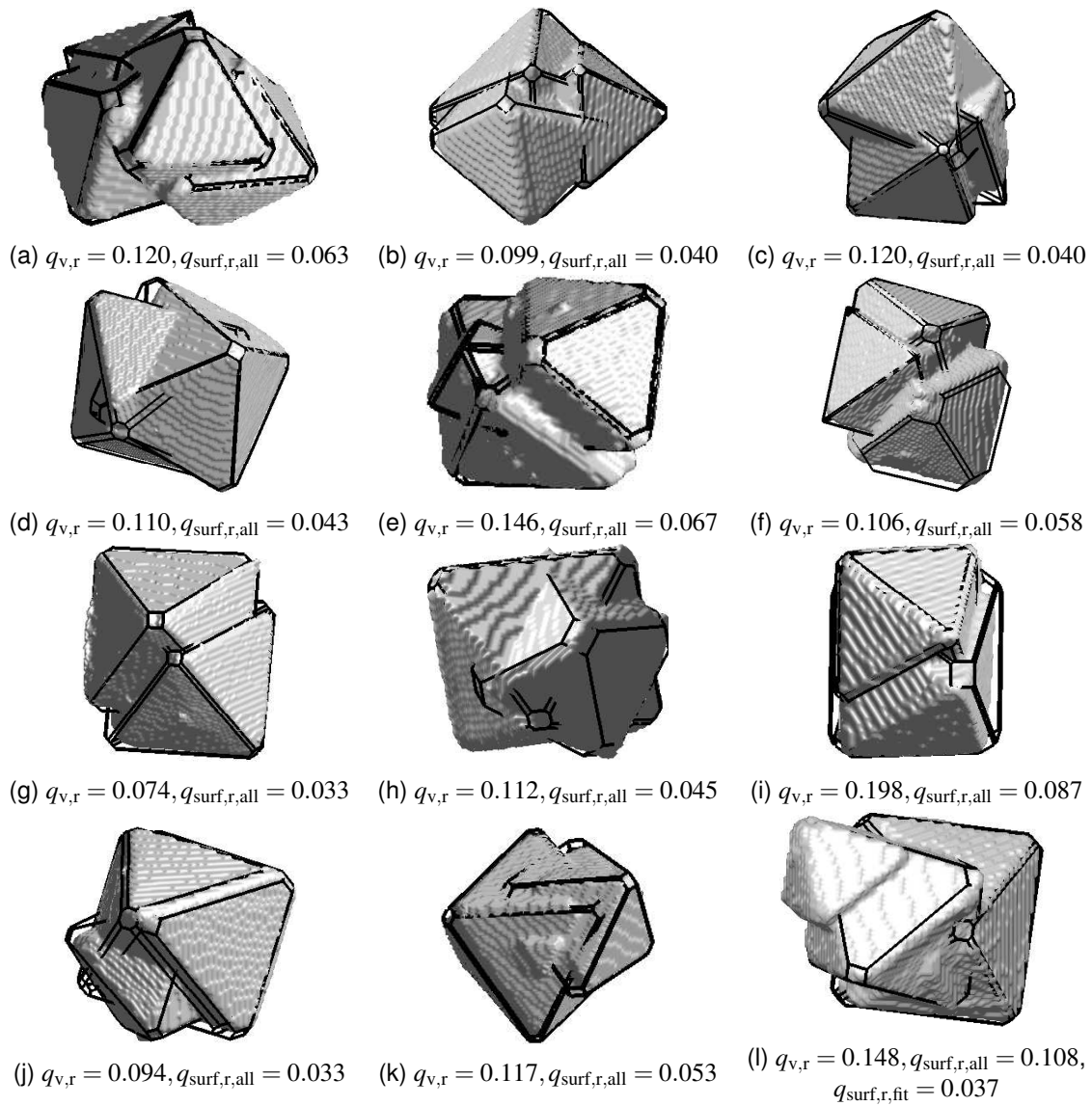


Figure 4.14: Shape identification in case of crystal aggregates: examples where the procedure performed well. Error measure $q_{surf,r,fit}$ was left out in cases where it was equal to $q_{surf,r,all}$.

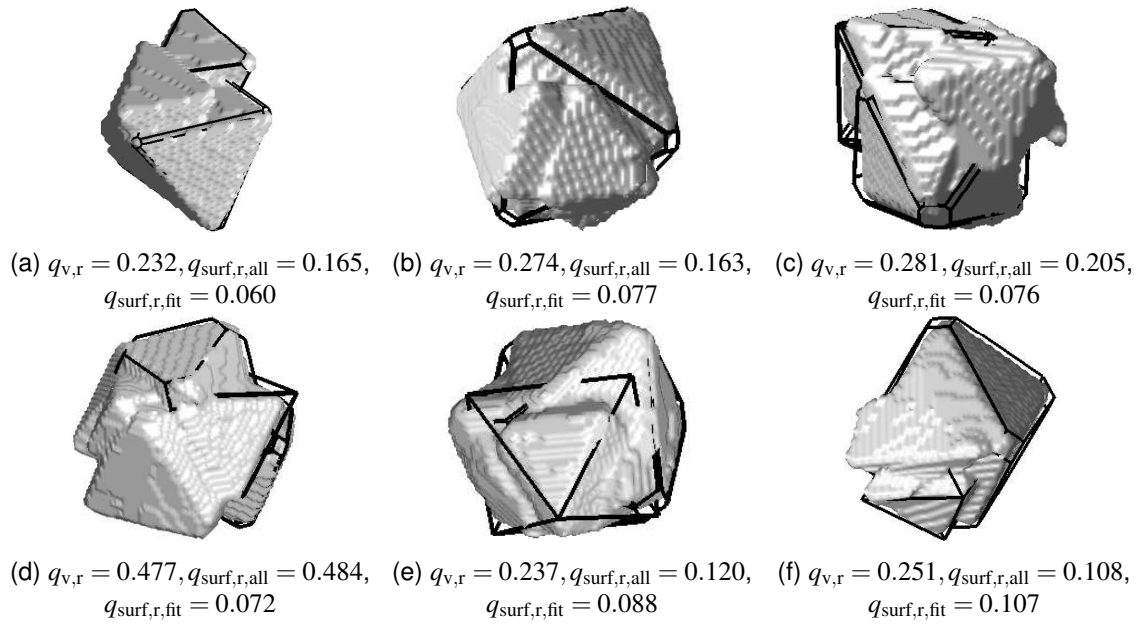


Figure 4.15: Shape identification in case of crystal aggregates: examples where the aggregate is considered to contain primary particles where no shape identification is possible.

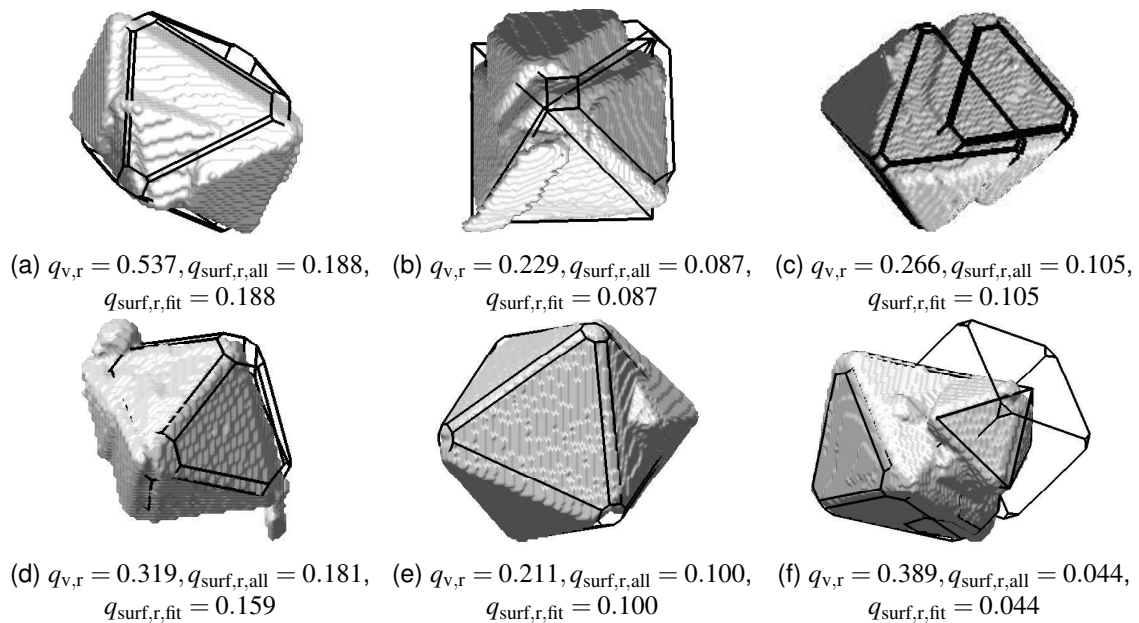


Figure 4.16: Shape identification in case of crystal aggregates: examples where the procedure performed badly.

Chapter 5

Orientation of Primary Particles in Crystal Aggregates

In the previous chapter, it was shown how μ CT imaging can be used to obtain the size, shape, position and orientation of each primary particle in a crystal aggregate. This information can then be used to measure the mutual orientation between pairs of primary particles and study the underlying aggregation mechanism. So far, such investigations were only possible using diffraction pattern studies, such as in the work of Collier et al. [34, 42]. They have shown that calcite aggregates tend to have aligned primary particles when grown at low ionic strength. Visual observation of experiments conducted at high ionic strength did not show such behavior. Therefore, it was concluded that particles grown at low ionic strength re-orient themselves upon collision into a more favorable position under the influence of a thicker electrical double-layer [34].

The aim of this chapter is to show the applicability of the developed method based on μ CT imaging to studying the orientation between primary particles in aggregates. Potash alum is chosen as a model compound, given that it easily crystallizes into octahedra. A regular octahedron is a dual body of the cube and has the same symmetry operations. In case of cubes, a theoretical distribution of the so-called disorientation angle, describing mutual orientations between crystallites, is known from the literature [131, 132, 147]. This theoretical distribution also holds for octahedral crystals, serving as a reference expected in the case that all orientations of each primary particle are equally probable. It can be assumed that perfectly aligned primary particles have a stronger bond as there is no mis-alignment of the respective lattices. It is thus possible that aggregates with aligned primary particles would be more likely to withstand the disruptive hydrodynamic forces. In case of potash alum, Pratola et al. [37] have shown that agglomerate strength increases with increasing supersaturation. At high supersaturation, aggregates are more likely to be formed as there is more material available, enabling the pair of particles to be cemented together. It is possible that aggregates formed at high supersaturation would be more likely to survive the hydrodynamic forces even when the primary particles are not aligned. Therefore, in cooperation with the group of Prof. Sundmacher, Otto-von-Guericke-University Magdeburg, we investigated the orientations between primary particles grown at different supersatura-

tion profiles. The obtained results were presented in a joint publication [3]¹, while the results of the first experiment, processed with a simpler procedure, were presented at the BIWIC conference [148].

5.1 Aggregation and Growth Experiments

The experiments were conducted at the Chair of Process Systems Engineering at the Otto-von-Guericke-University Magdeburg. Each experiment was a seeded batch cooling crystallization of aluminum potassium sulfate dodecahydrate, CAS No. 7784-24-9, also known as potash alum. The parameters of all experiments are given in Table 5.1. The first experiment, denoted by R, was conducted using 8 g of seeds purchased from Roth (Art. No. CN78.2). In the second set of experiments, denoted by M, 10 g of seeds purchased from Merck were used. This set consists of three experiments performed with different cooling rates in order to create different supersaturation profiles. The experiments M were conducted by Viktoria Wiedmeyer, at the time a PhD candidate at the Chair of Process Systems Engineering at the Otto-von-Guericke-University. The experiment R was performed together by Viktoria Wiedmeyer and the author. The experiments R and M further differ in the method of preparation for the imaging in the μ CT, as it will be explained below.

The experiments were performed in a stirred double-jacketed batch vessel with a round bottom and an inner diameter of 10 cm, shown in Figure 5.1. In order to obtain a narrow seed size distribution, the Roth and Merck material was sieved for 60 min between $200\ \mu\text{m}$ and $30\ \mu\text{m}$ using a vibratory sieve shaker (AS 200 control, Retsch). A solution of potash alum in water, saturated at $30\ ^\circ\text{C}$, according to the saturation measurements made in Magdeburg, was prepared at the start of each experiment. This solution was heated above the saturation temperature to ensure dissolution. An automatic cooling with the cooling rates given in Table 5.1 was started and seeds were added at a temperature slightly lower than the saturation temperature. Before stopping the experiments at the desired final temperature of $20\ ^\circ\text{C}$, samples of crystals were taken by a pipette with an inlet diameter between 1 mm and 2 mm. Each

¹In the previous publication, x - and y -axis of the crystals were permuted, which is corrected here. Further small bugs were fixed.

Parameter	R	M10	M5	M3
Seed manufacturer	Roth	Merck	Merck	Merck
Seed amount [g]	8	10	10	10
Seeding temperature [$^\circ\text{C}$]	29.9	29.8	29.8	29.8
Cooling rate [K/h]	5	10	5	3
Amount dissolved solid [g]	161.3	161.2	161.3	161.2
Amount of water [g]	1000.4	1000.5	1000.1	1000.1
Sampling temperature [$^\circ\text{C}$]	20.4	21.3	20.8	20.5
Stirring rate [rpm]	250	250	250	250
Effective voxel length [μm]	6.80	7.85	7.85	7.85
Preparation method	Aggregates	All	All	All

Table 5.1: Parameters of the conducted seeded batch cooling crystallization experiments [3].

sample consisted of the slurry drawn at four different positions in the vessel. The drawn samples were emptied on a Büchner funnel covered by a filter paper. The excess solution was filtered by a vacuum pump, the crystals were then washed by a cold 50-50 mixture of ethanol and water and air-dried.

During the experiments, the solution concentration was monitored by an ATR-FTIR probe (Nicolet iS 10 FT-IR, Thermo Fisher Scientific). The evaluation of the concentration and supersaturation was performed by Viktoria Wiedmeyer, using the partial least-squares (PLSE) calibration model previously created at OVGU Magdeburg [3]. There was an offset in the concentration measurement c_{meas} . It was corrected by considering the known concentration at the beginning of the experiment, $c_1(t_1)$ and the concentration $c_2(t_2)$ at the end of the experiment. It was assumed that the offset changes linearly with time t , so that the final concentration is obtained as [3]:

$$c(t) = c_{\text{meas}}(t) + \Delta c_1 + \frac{\Delta c_2 - \Delta c_1}{t_2 - t_1}(t - t_1), \quad (5.1)$$

where $\Delta c_1 = c_1 - c_{\text{meas}}(t_1)$ and $\Delta c_2 = c_2 - c_{\text{meas}}(t_2)$. The concentration c_2 at the end was measured by taking samples of the solution, filtering them to separate the crystals and allowing the water to evaporate.

The values of solution concentration $c(t)$ at different times t can be used to compute the supersaturation:

$$S(t) = \frac{c(t)}{c_{\text{sat}}(T(t))}, \quad (5.2)$$

where $c_{\text{sat}}(T(t))$ is the saturation concentration at the temperature T at time t . The saturation concentration was measured in Magdeburg and the resulting equation was provided by Viktoria Wiedmeyer [3]:

$$c_{\text{sat}}(T) = 0.18T_{\text{K}}^2 - 102.726T_{\text{K}} + 14760.7. \quad (5.3)$$

Here, T_{K} is the temperature in Kelvin and the resulting concentration value is in $\frac{\text{g hydrate}}{\text{kg free water}}$. This saturation concentration agrees well with the measurements made by Mullin et al. [149]. The finally obtained concentration and supersaturation profiles, as computed by Viktoria Wiedmeyer, are presented in Figure 5.2.

The experiment was further monitored by a QICPIC (R02, Sympatec) flow-through microscope. The setup, described in more detail in [116], involves continuously withdrawing the suspension by a peristaltic pump and pumping it through a measurement cuvette where the images are taken with a rate of 10 frames/s. An image from the beginning of the experiments R and M3 can be seen in Figure 5.3. It can be seen that the seeds dissolve slightly upon being introduced into the solution, as the particles are oval instead of showing sharp edges. This indicates that the saturation concentration measurement is not completely accurate. Furthermore, small crystals are visible in Figure 5.3b corresponding to the experiment M3. As the seeds in both experiments are drawn from the same sieve fractions, it is assumed that, in experiment M3, small crystals were attached to the large seed particles during sieving and were only detached upon suspension. The same behavior was observed in experiments M10 and M5 using the seed material from Merck.

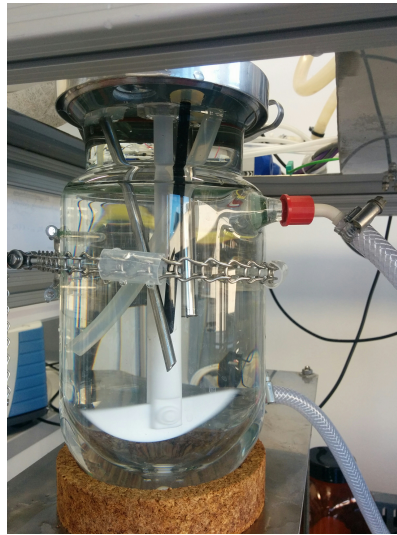


Figure 5.1: Crystallization vessel and the stirrer geometry.

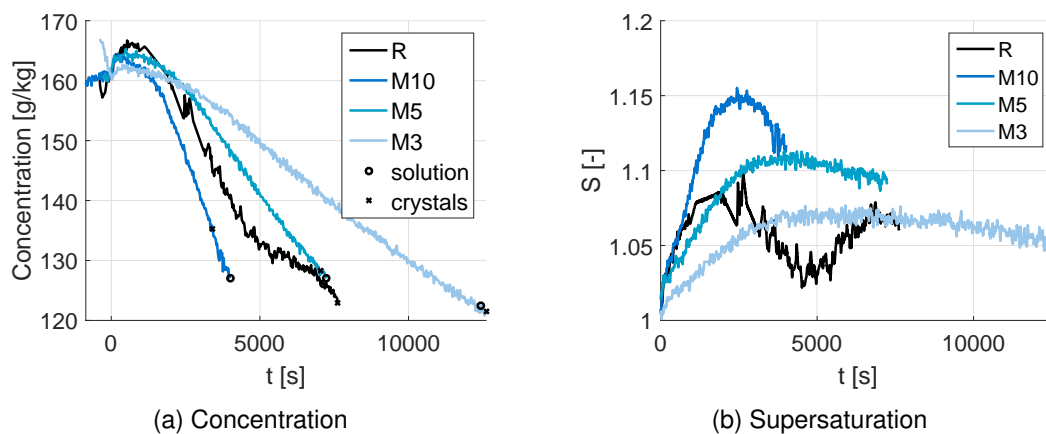


Figure 5.2: Concentration and supersaturation profiles measured in the experiments, based on the data computed by Viktoria Wiedmeyer. The time of solution and crystal sampling is marked by black circles and crosses, respectively. Seeding was performed at time $t = 0$ s for all experiments. Reprinted from [3] with permission from Elsevier.

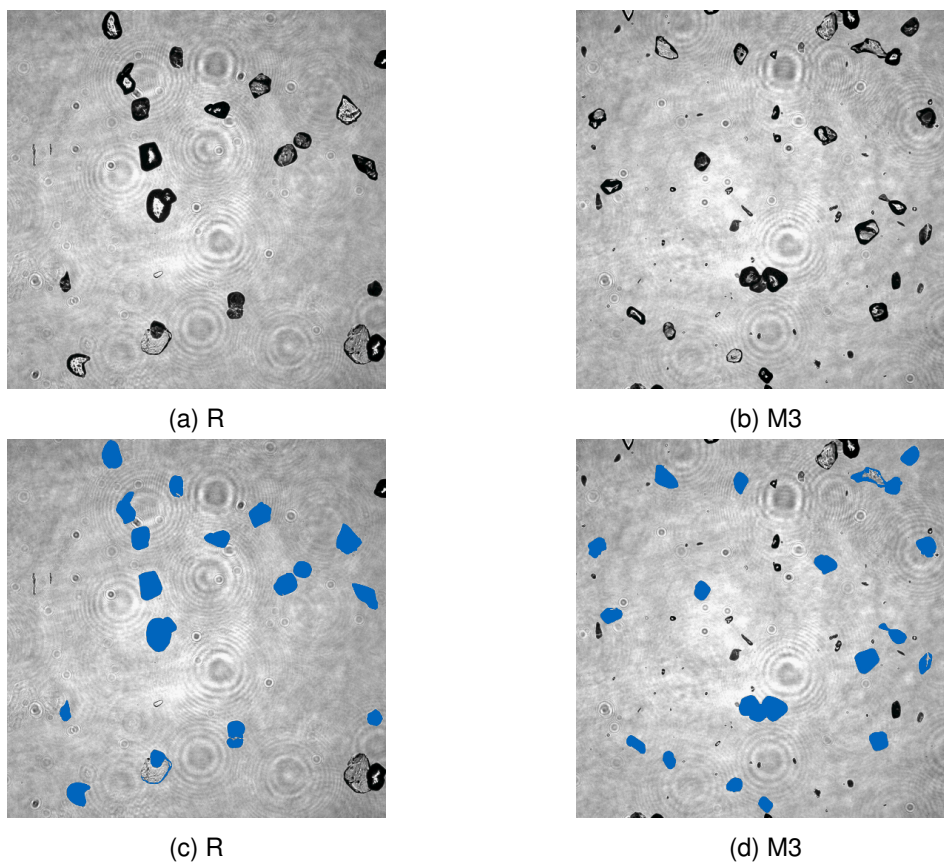


Figure 5.3: An image of the crystals taken 37s after adding seeds in the experiment R (a), and 270s after adding seeds in the experiment M3 (b). The images in the lower row contain particles where the area-equivalent circle diameter is greater than $200\mu\text{m}$ marked in blue. The images in the upper row are reprinted from [3] with permission from Elsevier.

5.2 Preparation and Imaging

The crystals obtained after sampling, washing and drying, as described above, were prepared for imaging in the μ CT using two different procedures. In case of the experiment R, the author performed the preparation procedure denoted as "Aggregates" in Table 5.1. Here, the filter paper with the sampled crystals was divided into two parts. Each crystal from one half of the filter paper was picked up by tweezers and observed under the microscope in order to determine whether it was an aggregate or a single crystal. Crystals that were deemed to potentially represent aggregates were sorted into a separate dish and were prepared for the imaging in the μ CT. Here, an adhesive tape was rolled around a toothpick and crystals were positioned onto the tape in rows of three, ensuring there is enough space between the crystals, as illustrated in Figure 5.4a. The tape with the crystals was then rolled around the toothpick, as shown in Figure 5.4b.

During the described preparation procedure, crystals can be damaged and aggregates may break into primary particles due to being picked up by tweezers several times. Thus, in case of M experiments, we chose to prepare both single crystals and agglomerates for imaging. Here, no sorting into single crystals and agglomerates was performed, so that the disintegration damage is less likely. This preparation procedure is referred to as "All" in Table 5.1 and was performed by the Magdeburg research group. About a tenth of each M sample was prepared in this manner [3].

μ CT imaging was performed by Jonathan Schock, a PhD candidate at the Chair of Biomedical Physics at the Technical University of Munich, lead by Prof. Franz Pfeiffer. A toothpick with the crystals was placed in a sample holder clamp and imaged with a ZEISS X-Radia Versa XRM 500 machine. The parameters used in the measurement are shown in Table 5.2. The CT reconstruction was performed by the proprietary software XRM Reconstructor and the final volume was cropped.

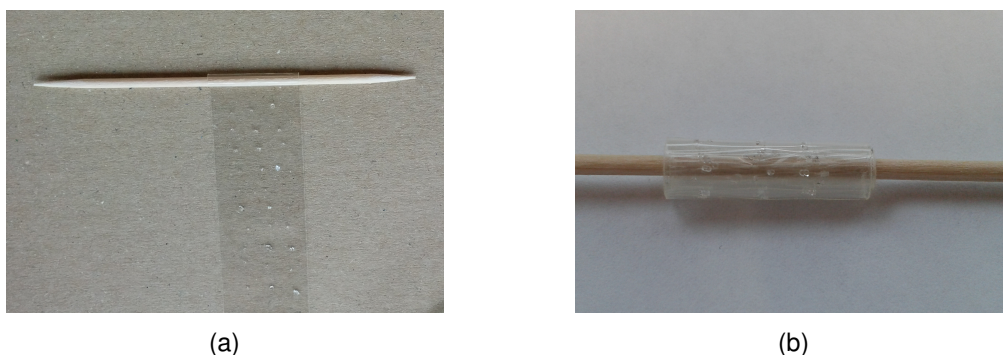


Figure 5.4: μ CT preparation procedure. Crystals are placed on an adhesive tape so that there is enough space between separate particles (a). The adhesive tape is then rolled around the toothpick, ensuring that the crystals remain separated (b). Reprinted from [3] with permission from Elsevier.

Parameter	R	M
Tube Voltage [kV]	80	80
Power [W]	6	6
Exposure Time [s]	5	5
Magnification [x]	0.39	0.39
Source-to-sample distance [mm]	25	30
Detector-to-sample distance [mm]	100	100
No. projections	1601	1601

Table 5.2: Parameters of the μ CT measurements performed by Jonathan Schock, Chair of Biomedical Physics, TUM. Reprinted (adapted) from [3] with permission from Elsevier.

5.3 Image Processing

5.3.1 Image Pre-processing

The obtained stack of images was loaded into the software MAVI [96], resulting in a 3D image of the measured volume. The 3D image was converted from 16 bit into 8 bit format in order to reduce the memory requirements. Figure 5.5 shows one horizontal slice with a cross-section through some of the crystals, layers of the adhesive tape and the toothpick. It can be seen that the difference in gray values between the crystals and the other structures is high, so that the crystals can be easily segmented. Therefore, the 3D image was binarized. The binarization threshold was chosen using the Otsu method for all experiments except M10, where the threshold was chosen manually by observing its effect on different slices. A labeling of connected components, using the 26-8 neighborhood was performed in order to separate different crystals. This image was stored for further processing with MATLAB [150].

In case of each experiment, each crystal (label) is extracted and stored into a separate file using MATLAB. Crystals are ignored if they touch the boundary of the original 3D image or if their total volume, approximated by counting foreground voxels, is smaller than the volume of a sphere of diameter $D = 200 \mu\text{m}$. This threshold is chosen to correspond to the pore size of the finer sieve used to obtain seed crystals. Thus, it ensures the exclusion of crystals that nucleate during the experiment. Note that the crystals grow during the experiment so that the final size is larger than the sizes obtained through sieving.

In experiment M10, five crystals whose shape did not correspond to either agglomerated or single crystals were observed. These particles are excluded from further consideration. Three such particles are illustrated in Figure 5.6.

5.3.2 Segmentation into Primary Particles

The procedure for segmenting images of crystal aggregates and extracting their size and shape was presented in Chapter 4 and is summarized in the flow chart given in Figure 4.2a. The validation study has shown that while a reasonable shape fit can be found in most cases, some issues are still present.

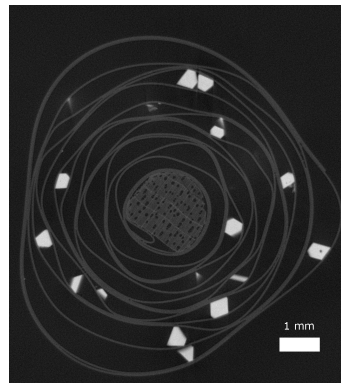


Figure 5.5: One horizontal slice from experiment M5. The porous structure is the toothpick, the onion-like layers are the layers of adhesive tape and the white areas are crystals. Reprinted from [3] with permission from Elsevier.

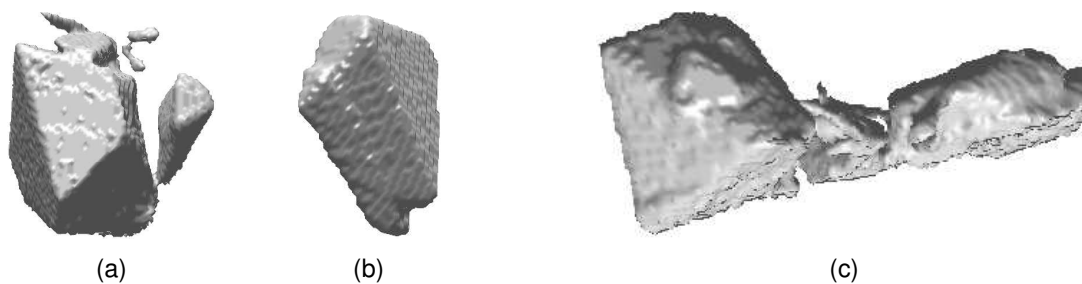


Figure 5.6: Examples of particles from experiment M10 that were excluded from the shape identification procedure as their shape did not correspond to either agglomerated or single crystals [3].

These problems are encountered in case of a large number of primary particles, high primary particle overlap and asymmetrical crystals. Thus, two user interaction steps are introduced here to resolve the segmentation issues. These steps are presented in blue in Figure 5.8a, showing the flow chart of the new procedure. Furthermore, symmetry conditions are removed in order to comply with the large amount of experimentally observed asymmetrical crystals, such as the one shown in Figure 5.7.

The first user interaction step concerns finding concavity points. The user is shown a crystal, together with the automatically detected concavity points, using the approach described in section 4.1.2.1. The user may delete existing concavity points by clicking on them, as well as add new concavity points by clicking on points of the surface point cloud. The algorithm then proceeds with the automatic steps shown in white in Figures 4.2 and 5.8. The result of the automatic segmentation is then presented to the user. The user sees two images, as exemplified in Figure 5.9. In the left image, primary particles are rendered in different colors. In the right image, the surface point cloud parts belonging to different primary particles are presented in different colors. The user can iteratively improve the segmentation. In each step, he or she can either concatenate regions or add new regions by interacting with the image on the right hand-side. If the regions are to be concatenated, the user clicks on some point in the right image for each of the chosen regions, as illustrated in Figure 5.9. When adding new regions, the user clicks on a point in the right image he or she believes should be in a new region. Here, the algorithm will find the region this point belonged to after the procedure that assigns watershed voxels to watershed regions. This region is re-introduced as a new region. Furthermore, if re-introducing a new region caused any of the existing regions to be split into more than one connected component, the resulted connected components are introduced as new regions as well. The user can thus introduce new regions and concatenate regions until the segmentation is satisfactory. At this point, the segmentation information is stored and the algorithm proceeds with the next crystal.

The user interaction steps may introduce some non-reproducibility into the procedure as both user interaction steps depend on the surface points chosen by the user. Even though most steps are obvious, as the one illustrated in Figure 5.9, the procedure is based on the user's decision regarding the

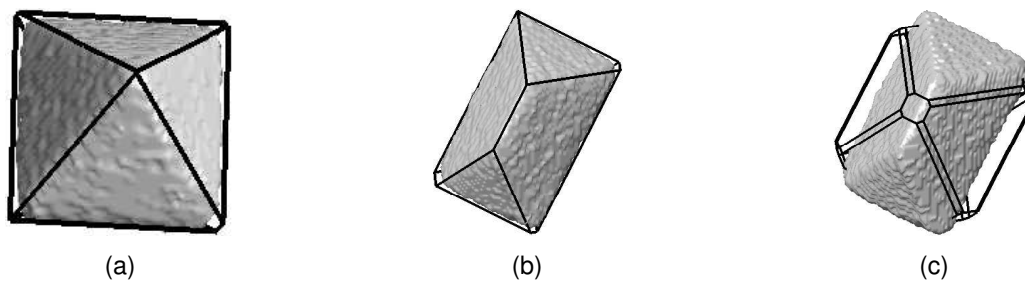


Figure 5.7: A symmetrical (a) and an asymmetrical crystal (b,c), both observed in the experiment M3 [3]. In case of (a) and (b), the identified shapes presented in black are obtained using the asymmetrical crystal model, as explained in section 5.3.3. The shape identification in (c) is performed using the method based on symmetry from chapter 4. Note that the symmetrical crystal model contained 26 faces, while the asymmetrical model consisted of 8 faces.

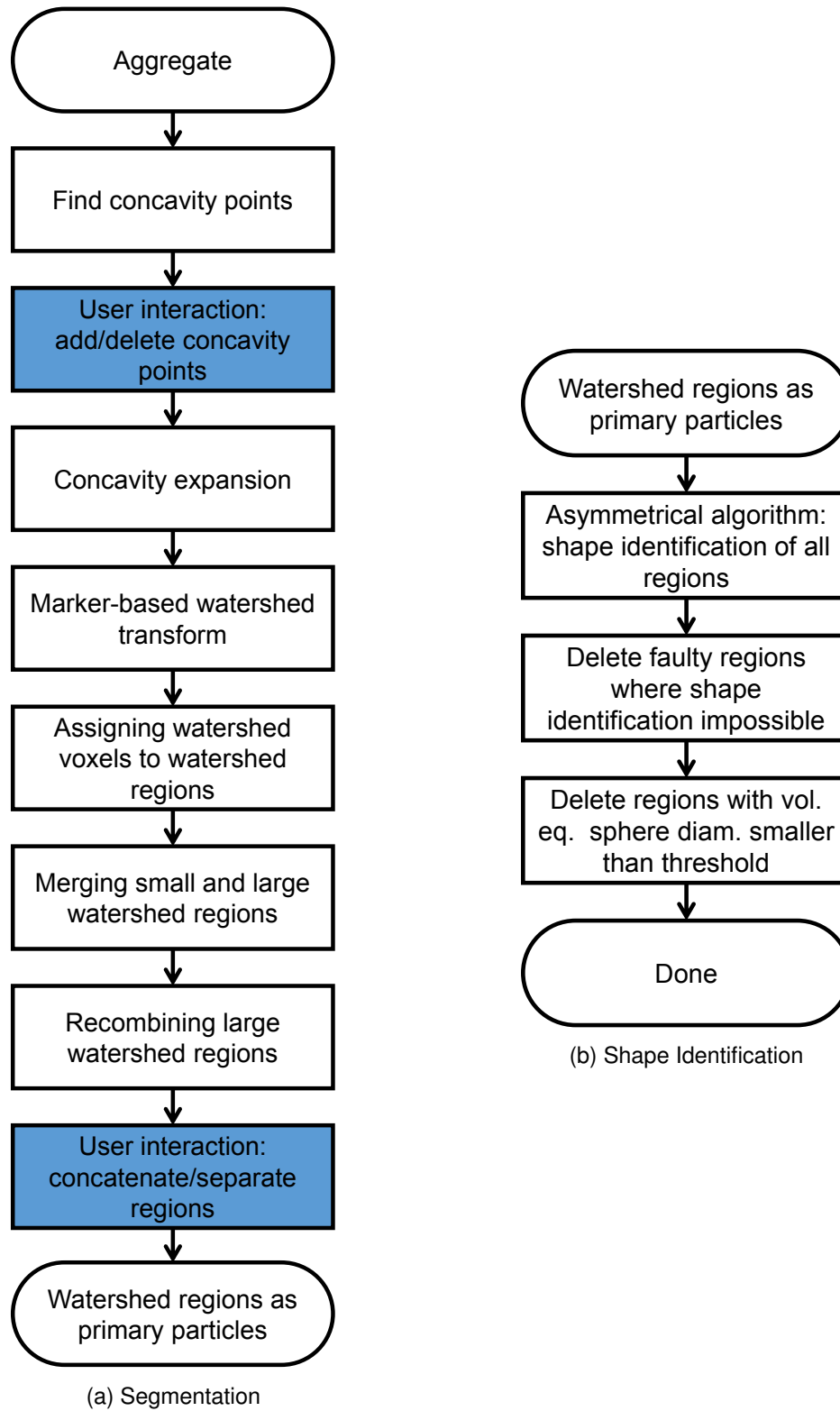


Figure 5.8: The procedure for identifying the shapes of primary particles in crystal aggregates using user interaction.

primary particles of an aggregate. A classical approach to reduce the bias would involve asking several users to perform the segmentation. Such an approach was chosen by Ochsenbein et al. [69] when classifying particles into single crystals and aggregates based on two 2D gray value images taken from perpendicular directions. In that study it was shown that the agreement between experts is reasonably good but not perfect. However, performing the user-assisted segmentation for the four data sets is a time-consuming task. Furthermore, even upon completing the shape identification, ambiguities regarding contact type between primary particles remain. We chose to resolve the contact-type bias by using the approach involving several users. Introducing several versions of primary particle segmentation would significantly increase the number of possibilities, so that this source of bias is not considered here and the segmentation performed by the author is used further.

5.3.3 Shape Identification of Primary Particles

The previously developed shape identification procedure, presented in Figure 4.2, used crystal symmetry as well as a set of post-processing steps whose aim was to correct faulty segmentation. Here, faulty segmentation is corrected by user interaction so that these post-processing steps are no longer needed. Furthermore, these steps were developed for symmetrical crystals and would not be applicable to the asymmetrical crystals considered here. Therefore, the resulting procedure, presented in Figure 5.8b is simpler than the procedures regarding symmetrical crystals, given in chapters 3 and 4. First, face normals are identified by Hough transform and matched to the set of model face normals using the procedure described in 4.1.3.1. It remains to find face distances. Here, the basic case, described in case of single crystals in section 3.1.3 is applied. The face distance h_i corresponds to the Hough bin with the highest number of points, if the rotated model face normal \mathbf{Ra}_i was matched to one of the identified face normals. Otherwise, h_i corresponds to the Hough bin with the highest point density. No symmetry conditions are applied and the origin of the crystal is set to coincide with the arithmetic mean of the coordinates of surface points for the considered primary particle. Note that no distinction between "existing" and "non-existing" faces on the crystal surface is made, as was the case in chapter 4. In case of "non-existing" faces, the approach chosen here may lead to an incorrect face distance as the face is deemed to be covered by another primary particle. However, no information is provided to compute a better approximation if symmetry conditions are not used, so that such face distances are not modified.

Upon completing the shape identification, the algorithm deletes faulty regions where no shape identification was possible. Furthermore, regions whose volume-equivalent sphere diameter, computed using the number of contained voxels to approximate the volume, is smaller than $200 \mu\text{m}$ are also deleted. This threshold corresponds to the mesh width of the finer sieve used for sieving seeds and serves to exclude crystals that cannot be resolved well with the used μCT resolution. Particles smaller than this threshold can result from nucleation and breakage and are observed later in the experiment. However, small particles were also observed at the beginning of the M experiments, as discussed in section 5.1. Note that the region volume is computed using only the visible part of the particle. If primary particles

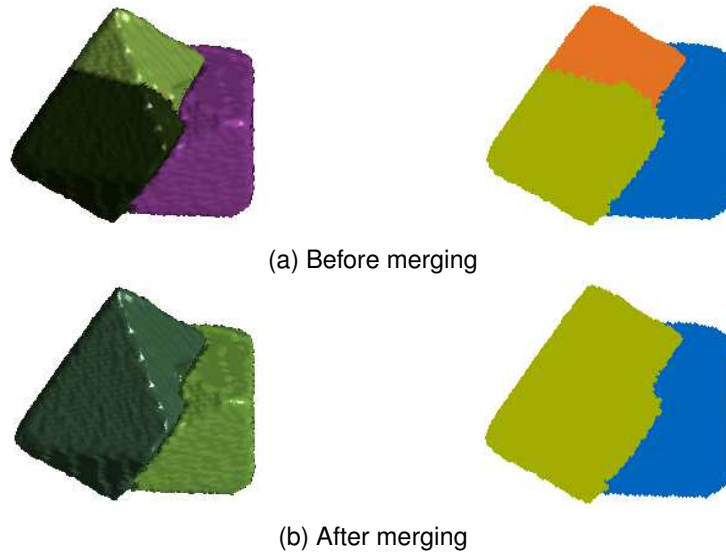


Figure 5.9: (a) The result of the segmentation procedure is displayed twice, as rendered crystals (left) and as a point cloud (right). Different colors are used to ensure sufficient contrast. The user can choose to concatenate the regions or add new regions. In this case, the user chose to concatenate the green and orange regions, upon which the segmentation was accepted. The result is shown in (b). Reprinted from [3] with permission from Elsevier.

overlap, they share some of their volume, but the shared volume cannot be simultaneously assigned to both particles. It is not possible to determine the size of the overlap volume as no boundary between the particles is visible. ²

5.3.4 Fit Quality Measures

In order to evaluate the quality of the shape identification, goodness of fit is approximated, similarly as in the automatic procedure presented in section 4.1.4.2. The volume-based quality measure $q_{v,r}$ is the same as that defined in equation 4.24. The remaining two surface-based fit measures are modified to contain the mean face distance $h_{\text{mean},i}$ value for each primary particle i , leading to [3]:

$$q_{\text{surf},r,\text{all}} = \frac{\sqrt{\frac{1}{N_{\text{surf}}} \sum_{j=1}^{N_{\text{surf}}} \min_i \|\mathbf{p}_j - \mathbf{p}_{\text{proj},j,i}\|^2}}{\max_i h_{\text{mean},i}} \quad (5.4)$$

²The new version of the program corrected the fact that x - and y -axis were erroneously exchanged in [3], as well as another small issue regarding the sampling of Hough distance normals. While the results regarding single crystals and the automatic shape identification of aggregates, presented in the previous chapters, could simply be re-evaluated with the new version, this was not possible here with a reasonable amount of effort as the repetition of user-interaction steps would have been necessary. Thus, the old segmentation, performed in [3] was employed. The flipping of the x - and y -axis was performed on the obtained segmented image instead of prior to segmentation. Then, the new program version regarding the shape identification could be applied. Note that flipping of the x - and y -axis may have an influence on the output of the watershed segmentation algorithm implemented in MATLAB. Particle classification by users, described in section 5.4.1, was then re-performed on the newly fitted crystals, according to the stored classification results from [3], yielding almost identical results.

and

$$q_{\text{surf},r,\text{fit}} = \frac{\sqrt{\frac{1}{N_{\text{surf},\text{fit}}} \sum_{j=1}^{N_{\text{surf},\text{fit}}} \min_i \|\mathbf{p}_j - \mathbf{p}_{\text{proj},j,i}\|^2}}{\max_i h_{\text{mean},i}}. \quad (5.5)$$

Similarly as in the case of automatic shape identification in chapter 4, aggregates were automatically classified into those where shape identification was successful, those that contained primary particles where no shape identification was possible and those where the shape identification was unsuccessful. It can be assumed that all primary particles were segmented by user interaction. Thus, an aggregate is declared to have a successful shape identification if shape identification was possible for each primary particle, while the fit quality measures fulfill $q_{\text{surf},r,\text{all}} < 0.15$ and $q_{v,r} < 0.2$. If shape identification was not possible for some primary particle, but $q_{\text{surf},r,\text{fit}} < 0.15$, an aggregate is considered to contain "missing" primary particles. Otherwise, the aggregate shape identification is considered unsuccessful.

The number of aggregates in each of the three categories is given in Table 5.3. Goodness of fit was measured in order to exclude aggregates where shape identification was unsuccessful from consideration when computing the angles between primary particles. However, as seen in Table 5.3, unsuccessful shape identification was only detected in one case in experiment M3, corresponding to a single crystal.

5.4 Disorientation Angle Distribution Measurement

5.4.1 Classification of Primary Particle Contact

Upon identifying the shape of each primary particle in an aggregate, orientations between pairs of primary particles can be evaluated. Here, all primary particle pairs in a considered aggregate are examined as an aggregate can consist of several primary particles with different mutual orientations. A pair of primary particles in an aggregate does not necessarily have to be in contact. Furthermore, particles that are in contact can either lightly touch or represent truly grown-in agglomerates. Pairs of lightly-touching particles could have come into contact during sampling and preparation as well as during the end phase of the experiment. Particles that are very lightly bound may also move during

Sample/Type	No. all well identified	No. with missing	No. with badly identified
R	46	15	0
M3	101	31	1
M5	104	55	0
M10	101	48	0

Table 5.3: Number of crystals where the shape identification was successful for each primary particle, no shape identification was possible for some regions ("missing") and the shape identification was unsuccessful for some primary particles. Reprinted (adapted) from [3] with permission from Elsevier.

preparation and transport, so that the measurement of their mutual orientation is not reliable. Grown-in primary particle pairs, however, cannot change their relative orientation without breaking the aggregate. Only such grown-in, agglomerated primary particle pairs are considered reliable for the forthcoming orientation analysis. Thus, it is necessary to classify the contact type of each primary particle pair as either "grown-in", "lightly-touching" or "none". Given a small amount of data to be processed, classification is done by human experts with knowledge of crystallization and/or image processing. Three experts, including the author, were asked to classify the primary particle pairs and only those pairs that were classified into the same class by all experts were used for the further analysis. The aim is to reduce bias and exclude problematic cases where the classification is difficult.

Experts were given the following description of each of the particle classes, as well as the examples illustrated in Figure 5.10 [3]:

- "'Lightly-touching' are all pairs of particles that could have been brought into contact without growth, so by simply placing them next to each other or onto each other. They can be separated without harming one of them".
- "'Grown-in' are crystals where one is grown strongly into the other and they cannot be separated without harm".
- "'None' are crystals that are not in contact".

In order to perform classification, experts are shown aggregates where only the considered primary particles are marked by their identified shapes, as illustrated in Figure 5.10. The expert may rotate the aggregate and change the light settings to get a better impression of the contact type. The number of pairs belonging to each class, as identified by each expert, is given in Table 5.4 [3].

5.4.2 Disorientation Angle

Upon determining the type of contact between a primary particle pair, it is possible to quantify their mutual orientation. Here, the shape model used for a primary particle was an octahedron. A regular

³In [3], the numbers of primary particle pairs with contact types "touching" and "none" in the experiment R, as evaluated by the user T, were erroneously exchanged.

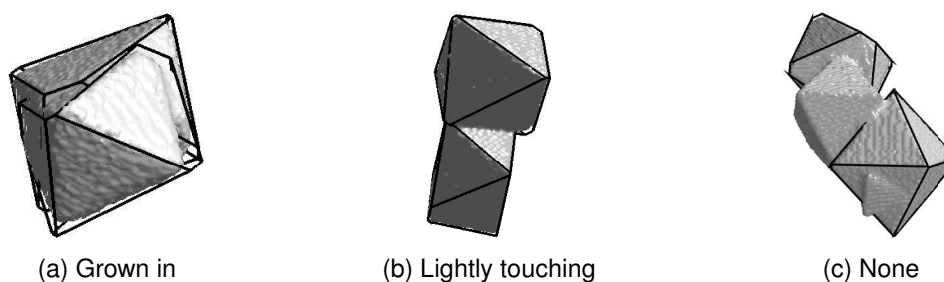


Figure 5.10: Examples of contact classes. Reprinted from [3] with permission from Elsevier.

	R				M3				M5				M10			
Expert	H	M	T	all	H	M	T	all	H	M	T	all	H	M	T	all
grown-in	40	40	40	40	35	33	34	33	30	32	29	28	29	26	26	26
touching	6	7	5	5	40	43	41	40	53	50	53	46	34	35	35	32
none	5	4	6	4	15	14	15	14	31	32	32	27	21	23	23	21

Table 5.4: Number of primary particle pairs from experiments R, M3, M5, and M10 that were classified as "grown-in", "lightly-touching" and "none" by three different experts, denoted by H, M and T. Also, the number of pairs that were classified into the same class by all three experts, denoted by "all" [3].³

octahedron is a dual body of a cube, so that theoretical considerations regarding orientations between pairs of cubes [131, 132, 147] can be applied to regular octahedra. Due to the asymmetry of the primary particles, however, the identified shapes are not regular octahedra as their face distances from the crystal middle point may differ. However, the directions of face normals for each primary particle still represent the rotated versions of the directions of face normals for a regular octahedron. This was ensured by the design of the procedure. Thus, mutual orientations can be determined using the same approach as in case of regular octahedra and cubes.

An orientation between two bodies of the same size, centered at the same point, can be described by the rotation operation necessary to rotate one of the bodies into full overlap with the other. This operation can be expressed as a rotation by some angle Θ around some axis, or, alternatively, as a rotation matrix \mathbf{R} . In case of cubes, $i = 24$ different symmetry operations can be applied to the rotated body, without changing the fact that the two bodies now overlap. This results in $i = 24$ different rotation operations and a set of different angles of rotation Θ_i . The disorientation angle is the smallest possible rotation angle under these symmetry operations [147], so that:

$$\Theta = \min_i \Theta_i. \quad (5.6)$$

A considered set of primary particle pairs can then be described using the disorientation angle distribution, representing the probability density distribution over disorientation angles.

In order to compute the disorientation angle, the two primary particles A and B are moved so that their centers are in the coordinate system origin. Furthermore, each particle is given a unit face distance in each direction so that it now represents a regular octahedron. The obtained primary particles can be described by the matrices of face normals \mathbf{A}_A and \mathbf{A}_B . Note that the order of face normals in these matrices must be such that they represent a rotated version of the matrix of model face normals \mathbf{A}_r . The matrix of face normals \mathbf{A}_r defines the reference octahedron which is presented in Figure 5.11 along with its dual cube. The 24 symmetry operations \mathbf{S}_i on this cube, and thus on the reference octahedron, are [3, 131]:

- Rotation matrix equal to the identity matrix
- Rotations of 90° , 180° and 270° , around the coordinate system axes x , y and z , resulting in a total of 9 rotation matrices

- Rotations of 180° around axes parallel to the six face diagonals, resulting in a total of 6 rotation matrices
- Rotations of 120° and 240° around axes parallel to the four cube diagonals, resulting in a total of 8 rotation matrices

The disorientation angle is the smallest angle that will rotate the octahedron A into coincidence with B under the consideration of symmetry operations. A rotation matrix \mathbf{R}_1 that rotates A into the reference position \mathbf{A}_r is first determined using

$$\mathbf{R}_1 = \left((\mathbf{A}_{A,1:3})^{-1} \mathbf{A}_{r,1:3} \right)^T. \quad (5.7)$$

Here, the index 1 : 3 in the subscript denotes that the first three face normals, corresponding to first three matrix rows, are used to find the rotation matrix. This rotation matrix is then applied to the two octahedra, resulting in the following matrices of face normals:

$$\tilde{\mathbf{A}}_A = \mathbf{A}_A \mathbf{R}_1^T \quad (5.8)$$

and

$$\tilde{\mathbf{A}}_B = \mathbf{A}_B \mathbf{R}_1^T, \quad (5.9)$$

where $\tilde{\mathbf{A}}_A$ now coincides with \mathbf{A}_r . Therefore, the relative orientation between the two octahedra A and B did not change. The octahedron $\tilde{\mathbf{A}}_B$ can be obtained by rotating the reference octahedron \mathbf{A}_r by \mathbf{R} , where:

$$\mathbf{R} = \left((\mathbf{A}_{r,1:3})^{-1} \tilde{\mathbf{A}}_{B,1:3} \right)^T. \quad (5.10)$$

Now, symmetry operations can be applied to the octahedron given by $\tilde{\mathbf{A}}_B$. These symmetry operations are given by rotation matrices $\tilde{\mathbf{S}}_i$. Note that they are not the same symmetry operations as \mathbf{S}_i defined above, as the polytope defined by $\tilde{\mathbf{A}}_B$ is not dual to the reference cube but to the reference cube rotated by \mathbf{R} . Thus, in order to obtain $\tilde{\mathbf{S}}_i$, the rotation is executed around one of the axes defined above, rotated by \mathbf{R} , while the angle of rotation is kept the same. Rotation angles corresponding to the rotation

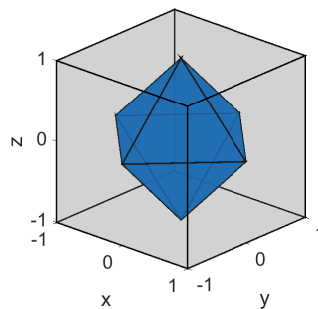


Figure 5.11: Reference orientation of cube and regular octahedron. Reprinted from [3] with permission from Elsevier.

$\tilde{\mathbf{A}}_B \tilde{\mathbf{S}}_i^T = \mathbf{A}_i \mathbf{R}^T \tilde{\mathbf{S}}_i^T$ can be computed by [147]

$$\Theta_i = \arccos \left(\frac{\text{tr}(\tilde{\mathbf{S}}_i \mathbf{R}) - 1}{2} \right). \quad (5.11)$$

The disorientation angle is the smallest of the angles Θ_i , as given by equation 5.6. Here tr is the trace operator, summing the diagonal elements of a matrix. Examples of disorientation angles for pairs of experimentally observed primary particles are given in Figure 5.12.

5.4.3 Disorientation Angle Distribution

Mackenzie and Thomson [147] computed the disorientation angle distribution (DAD) for cubes by a Monte Carlo simulation, while it is assumed that each orientation of the cubes A and B is equally probable. An analytical solution for the DAD, $p(\Theta)$, was derived a year later by both Mackenzie [131] and Handscomb [132] and is given in the appendix B. According to their computations, the largest possible disorientation angle is $\Theta_{\max} = \arccos \left(\frac{1}{4} (2\sqrt{2} - 1) \right) \approx 62.8^\circ$ [132], the probability density is highest for the angle of $\Theta = 45^\circ$ [131], whereas the mean value is 42.7° [131, 132].

This theoretical DAD can be compared to the experimentally obtained values. Here, the experimental DAD was computed for grown-in primary particle pairs, lightly-touching pairs, as well as for a combination of both. In all cases, only those pairs where the three experts agreed upon the type of primary particle contact were used, as explained in section 5.4.1.

5.5 Validation by Simulation

The procedure for computing the disorientation angle using the segmentation and shape identification methods defined in this chapter is validated on simulated crystals. First, it is necessary to ensure that the shape identification procedure correctly identifies particle orientation. To this end, simulated single crystals, cases S1-1 and S1-5 from Table 3.1 were used. In case of single crystals, disorientation angles are measured between the simulated and the identified polytope, thus giving information on the accuracy of orientation of the identified polytope. This furthermore represents a validation mechanism

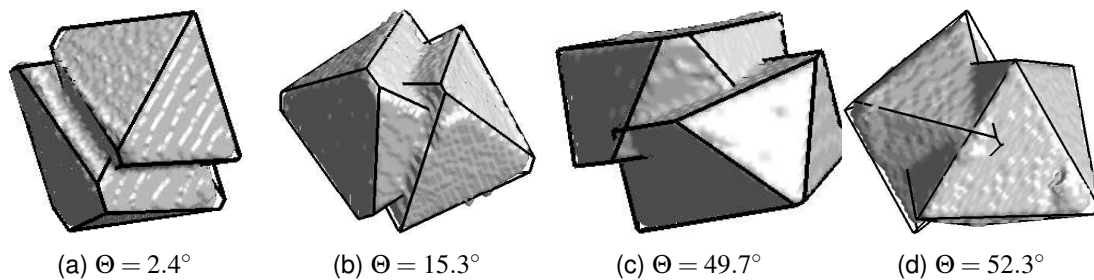


Figure 5.12: Examples of disorientation angles for pairs of primary particles observed in the experiment M10 [3].

showing that disorientation angles close to zero are identified correctly. The results of case S1-1 are shown in Figures 5.13. The error in face direction is computed by comparing the simulated face distance to the mean of the identified ones, while the error in volume is computed as defined in chapter 3. There is no visible difference to the results regarding symmetrical crystals, presented in Figure 3.9. The effect is the same in case of the parameters corresponding to S1-5. This is a result of the fact that the symmetrical shape identification simply averages the asymmetrical face distance and assigns it to all faces, while the orientation is the same, so that only the volume may change, but the change is very small.

In order to validate the procedure for measuring disorientation angles between pairs of primary particles, simulated aggregates, case A1-0 from Table 4.1, are used. Note, however, that an octahedral potash alum model is used here, in contrast to the potash alum model containing 26 faces presented in chapter 4. In case of crystal aggregates, two calculations are performed. First, the DAD is calculated using simulated polytopes. Then, the DAD is measured on the polytopes identified in the simulated 3D images. The results are shown in Figure 5.14. In case (a) of simulated polytopes, the disorientation angle is measured only for polytopes that intersect. In case (b) of the shape identification in simulated images, the results for both "grown-in" and "lightly-touching" primary particle pairs are presented together, where the classification is performed by the author. The blue line in this Figure represents the analytical solution [131, 132]. In both cases, the obtained DAD agrees well with the theoretical prediction. Therefore, it can be concluded that the algorithm is capable of estimating the DAD of a given sample. Some differences in the height of the bar graphs in Figures 5.14a and 5.14b are caused by the error in estimating the disorientation angle.

5.6 Results

Crystal samples obtained in each of the four experiments presented in Table 5.1, were processed as described in section 5.3. The information regarding the number of processed crystals of different types is given in Table 5.5. It can be seen that all samples contained a large number of single crystals. This is especially prominent in case of M experiments, as expected from the preparation procedure. According to Table 5.5, both single crystals and aggregates often contained attached small particles. Note that the segmented parts of small particles may be incomplete, so that the complete particle is larger than the used threshold of $200\ \mu\text{m}$. Furthermore, small particles may in certain cases represent surface roughness and were marked as separate regions by the user in order to improve the likelihood of successful shape identification.

Upon completing the shape identification, crystals were processed as described in section 5.4 in order to obtain the DAD of "grown-in" and "lightly-touching" primary particle pairs, as well as of a combination of both contact types. The results are presented in Figure 5.15. In each case, the graphs show a deviation from the theoretical DAD [131, 132] presented as the blue line, which is expected under the assumption that all primary particle orientations are equally likely. The most likely configuration expected from these analytical considerations is 45° , while low disorientation angles are improbable.

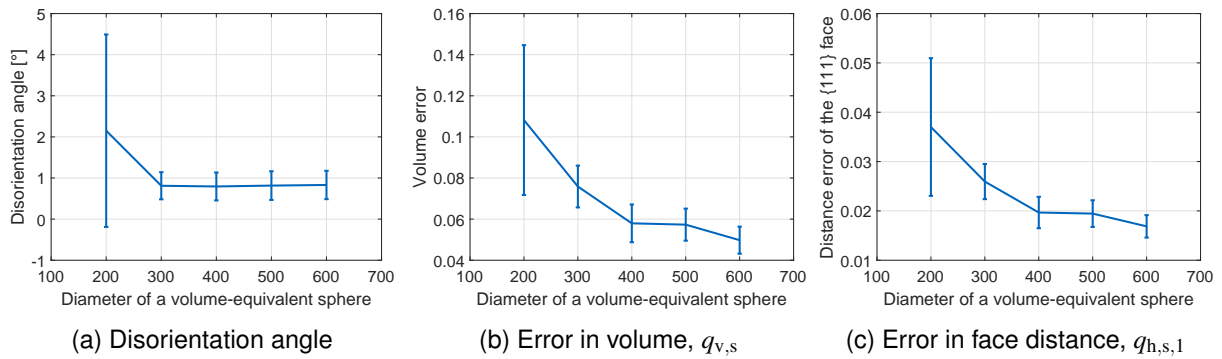


Figure 5.13: Mean error measures for 100 simulated potash alum crystals of different sizes. Error bars represent standard deviation. Simulation and fitting parameters correspond to case S1-1, given in Table 3.1, except that no symmetry conditions are imposed, in accordance to the shape identification procedure presented in this chapter.

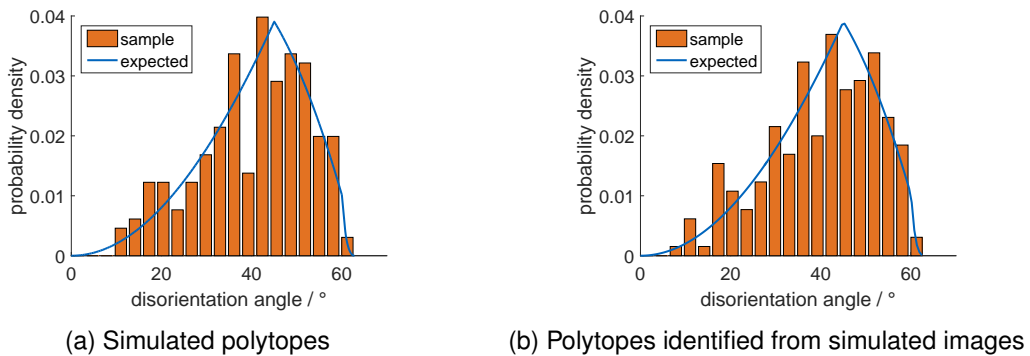


Figure 5.14: DAD computed on the simulated polytopes (a) and DAD computed on the polytopes identified from the simulated 3D images (b). The blue line is the analytical solution for uniform orientations of constituting primary particles [131, 132]. The bar graphs contain data on both grown-in and lightly-touching primary particle pairs. Reprinted (adapted) from [3] with permission from Elsevier.

However, in all cases, we observed a higher than expected occurrence of low disorientation angles, indicating particle alignment. This is especially prominent in case of grown-in pairs. These primary particle pairs could not have been created during sampling and preparation, as it is possible with the lightly-touching pairs. They are thus considered as more reliable. Lightly-touching primary particle pairs show a more even distribution of disorientation angles. However, an unexpectedly high number of small disorientation angles was still observed. It is possible that some of these pairs were created during the end phase of the experiments, so that not enough time remained for particles to fully grow into each other. Furthermore, if the considered pairs are created during sampling, the orientations of primary particles are not random as particles land on the filter paper under the influence of gravity. In these cases the assumption of the theoretical distribution is not fulfilled, so that the experimentally measured DAD would likely deviate from the theoretically expected one.

Experiment R resulted in a higher peak at low disorientation angles than the M experiments. This could have been caused by the different preparation procedure. In case of experiment R, crystals were sorted into single and aggregated particles, which required moving each particle three times with a pair of tweezers. Therefore, particles are more likely to break during this procedure. Breakage is more likely for loosely-bound particles and those where the bond strength is lower, which likely corresponds to non-aligned particle pairs and may lead to less ambiguity regarding particle contact. Furthermore, experiment R used less seed obtained from a different manufacturer. The two manufacturers reported different impurities which might have an influence on the DAD, along with the potential differences in the initial seed shape.

The results from the three M experiments show a similar DAD, so that an influence of the supersaturation profile on the DAD cannot be confirmed. It is possible that such an effect exists, but could not be observed within the used small sample of crystals. As seen in Table 5.5, the small sample size is related to the fact that many single crystals were observed in the experiment, as well as many small particles where no shape identification was possible. Furthermore, a large part of the investigated samples consisted of less reliable lightly-touching primary particle pairs, as seen in Table 5.4. In order to further investigate the influence of supersaturation, a larger sample should be collected. This can be achieved by using more seed crystals and thus promoting agglomeration. Furthermore, experiments should be conducted at constant supersaturation. Higher differences in supersaturation profiles should be created, while paying attention to avoid nucleation, as small crystals cannot be resolved by the same level of μ CT resolution.

Figure 5.16 shows the crystal sizes in experiment M5, without segmentation into primary particles, measured by both μ CT and QICPIC. In case of μ CT, crystal size represents the volume-equivalent sphere diameter obtained using the number of voxels as a volume estimate. In case of QICPIC, crystal size is the diameter of a sphere with an equivalent projected area and it is obtained using videos towards the experiment end. Both methods exclude crystals smaller than $200\mu\text{m}$ in the respective diameter. QICPIC measurements were provided by Viktoria Wiedmeyer and were obtained using an image processing program written by their group. It can be seen that, while the peak is at the similar location for both methods, QICPIC images also contained large crystals that were not present in the

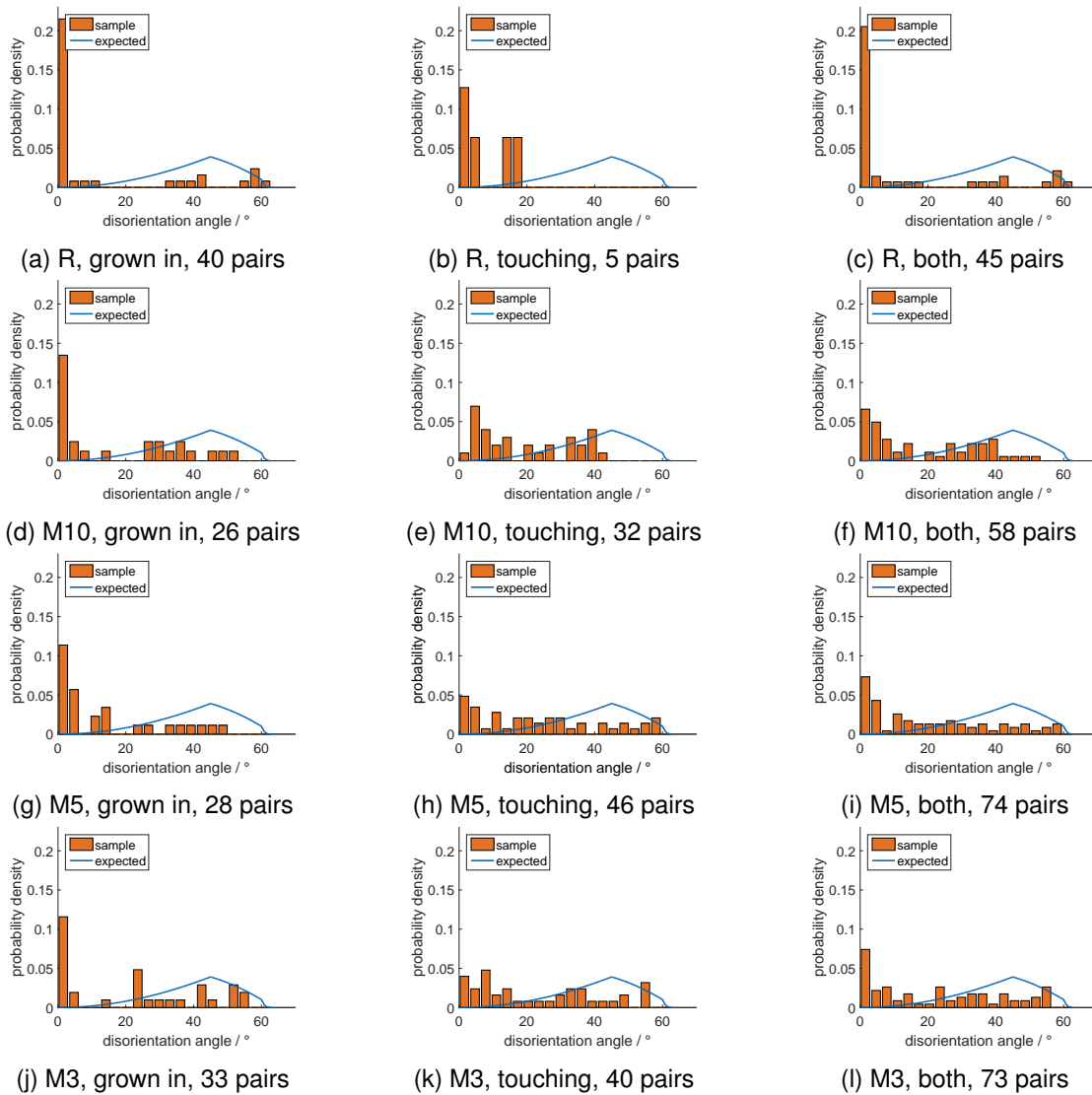


Figure 5.15: DAD for the four conducted experiments given in Table 5.1 is presented in orange. The first column represents grown-in agglomerate pairs, the second are the pairs of lightly-touching crystals, and the third column is the distribution obtained when considering grown-in agglomerates and lightly-touching pairs together. The blue line represents the theoretically expected distribution [131, 132]. The number of primary particle pairs included in the computation can be seen below the graphs. Reprinted (adapted) from [3] with permission from Elsevier.

Sample / Type	single all fitted	single with small	single with failed	agg. all fitted	agg. with small	agg. with failed	no fit due to small	no fit due to failed
R	15	6	1	31	5	3	0	0
M3	68	13	3	34	14	1	0	0
M5	75	20	7	29	24	3	0	1
M10	74	24	5	27	13	6	0	0

Table 5.5: Number of different crystal types in each sample. Columns 2-4 show single crystals, single crystals with attached small particles, and single crystals with particles for which the fitting procedure failed to produce a result. Columns 5-7 show aggregates for which a fit was possible for all primary particles, aggregates that also contained small particles, and aggregates that contained primary particles for which no fit was possible. Columns 8 and 9 show particles for which no shape fit was possible because each particle was too small and the fitting procedure failed to produce a result. Note that "single" refers to crystals for which the shape identification was possible for only one particle, whereas in case of "aggregates" it was possible for multiple particles. Reprinted (adapted) from [3] with permission from Elsevier.

μ CT images. This can be attributed to the inner diameter of the pipette used for sampling, which was between 1 mm and 2 mm, while the QICPIC cell had the dimensions of 5 mm x 5 mm x 2 mm. Furthermore, crystals may not be randomly oriented towards the camera in such a cell. Also, crystals that in reality do not touch may produce projection images which would appear to represent larger aggregates. Future experiments should use a sampling procedure which ensures that larger crystal aggregates are captured as well.

The here observed aligned primary particle pairs cannot be distinguished from aggregates produced by primary aggregation mechanisms such twinning and parallel growth. Potash alum is known to exhibit parallel growth [31, p. 158-160]. We did not observe parallel growth structures of three or more primary particles as depicted in literature. However, it is still possible that these mechanisms are responsible for the structures involving two particles or three primary particles that are not stacked on top of each other. The method presented here is not capable of differentiating between twinning and secondary aggregation mechanisms, which is possible using TEM and the TEM diffraction pattern studies, as shown by Collier et al. [34, 42]. However, the advantage of the here presented method is a significantly simpler sample preparation and measurement. There is no need for slicing the sample into thin sections or manually tilting the sample to obtain useful diffraction patterns. Furthermore, larger samples are feasible by preparing several adhesive tape structures.

5.7 Conclusion

In conclusion, it was demonstrated that the presented method is capable of measuring the disorientation angle distribution of the chosen model compound. Aligned pairs of primary particles were observed more often than expected from theoretical considerations which assume that all primary particle orientations are equally likely. No dependence on supersaturation could be confirmed. Assuming that the

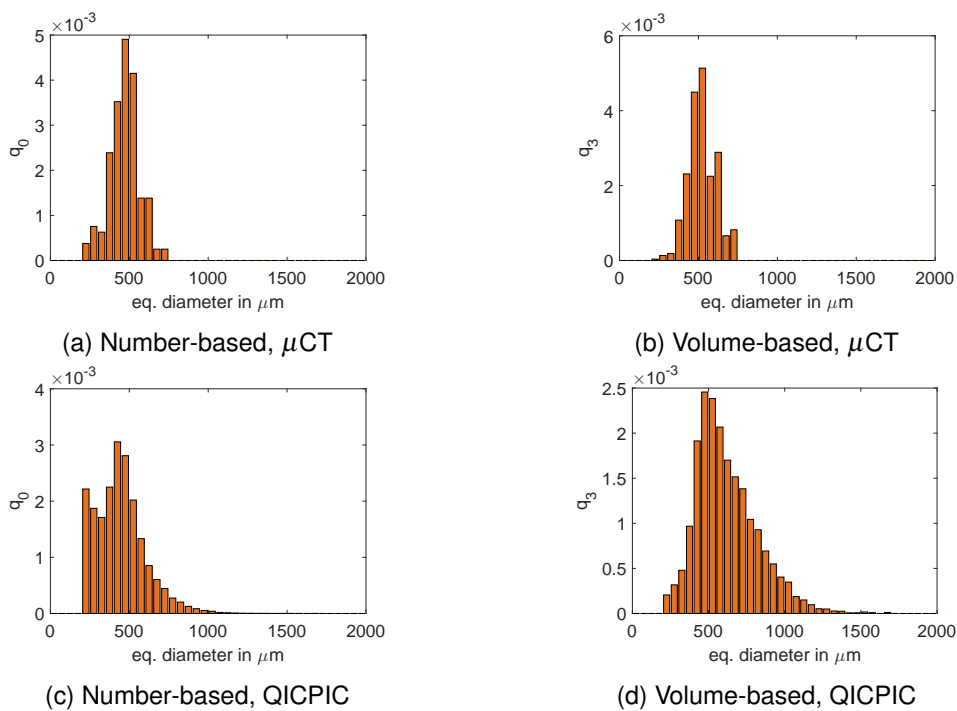


Figure 5.16: Volume-equivalent sphere diameter measured using μCT images (upper row) and a projected area-equivalent sphere diameter measured using a QICPIC video towards the experiment end (lower row), both for experiment M5. Number-based histograms are shown to the left, while volume-based ones are shown to the right. Object areas in pixels, needed to compute the figures in the lower row, were calculated by Viktoria Wiedmeyer and provided to the author.

symmetry operations are known, the procedure can be applied to other compounds as well, yielding information about the tendency of particles to form aligned aggregates and thus shedding light on the aggregation mechanism. In case that no analytical DAD under the assumption of uniform primary particle orientation is available in the literature for the considered crystalline compound, Monte Carlo simulations can be conducted to provide an approximation.

Chapter 6

Classification of Imaged Objects

While 3D imaging techniques deliver the full shape information regarding each primary particle in a crystal aggregate, they cannot be implemented on-line, require sampling of the crystals and are both time- and cost-intensive. A simpler alternative are 2D imaging techniques that capture only one projection image of the particle. These techniques can be implemented on-line and provide enough information to determine the amount of aggregates in the product, which can be considered as the first step towards characterizing the product in terms of aggregates.

In cooperation with the Laboratory of Plant and Process Design, TU Dortmund University, we investigated two methods for determining the agglomeration degree, using two chemical systems, L-alanine/water and adipic acid/water. L-alanine forms equant bipyramidal particles when crystallized from water whereas the crystals of adipic acid obtained in water are plate-like and more prone to agglomeration than those of L-alanine [4]. In this context, the agglomeration degree is defined as the amount of crystals that are aggregates. Both methods are based on automatic classification and are presented in a joint publication with the Dortmund group [4]. The underlying classification procedure considers three classes, namely single crystals, aggregates, and gas bubbles, which may occur when dynamically imaging a particle suspension. Variables used for classification are the image descriptors describing size and shape of the imaged objects. While the research group in Dortmund focused on the discriminant factorial analysis (DFA), artificial neural networks (ANN) were investigated at the Technical University of Munich. Properties of the two methods were reviewed in section 2.3.4.3. DFA is a parameter-based method, estimating the parameters of the underlying normal distribution of each variable and using this information to obtain a linear classifier. ANN can be considered as non-parametric, using an interconnected network of non-linear functions to provide a non-linear classifier. Both methods use supervised learning, where a training set containing objects whose class was manually determined by a human expert are provided as input to the algorithm. In the following, a detailed description of the procedure and the used data is given.

The goals and hypotheses of this study were defined together with the collaborators at TU Dortmund University [4]. The first goal is to compare DFA and ANN, where the hypothesis states that ANN would perform better. The remaining two hypotheses which are of significance to this thesis state that a larger

training set improves the classification accuracy, as well as that the obtained classifiers show poor transferability to experiments performed under different conditions. The hypotheses were motivated by the results of the previous study using DFA by the research group in Dortmund [68] performed on images obtained by an off-line transmitted light scanner. In this thesis, results of the ANN classification are presented along with the necessary data. Results regarding the analysis with DFA can be found in the joint publication [4].

6.1 Crystallization Experiments

The crystallization experiments were performed by the Dortmund research group. Information about the conducted experiments which is presented here was provided by Stefan Heisel, a PhD candidate in Dortmund [4]. Three batch cooling crystallization experiments using L-alanine, denoted as Ala1, Ala2 and Ala3, as well as three gassing crystallization experiments with adipic acid, denoted as AA1, AA2 and AA3, were conducted. The crystallization and post-processing parameters are provided in Table 6.1. Relevant aspects of the experimental procedure are presented below, while more details are given elsewhere [4, 30, 74]. The experimental conditions have an influence on the agglomeration degree A_g , as determined by a preliminary classifier developed by the Dortmund group and presented in Table 6.2. Therefore, these experiments can be used to investigate classifier transferability, where a classifier is transferable if it can be applied to data obtained under different experimental conditions.

In case of L-alanine, all three crystallization experiments were conducted using identical parameters, while the obtained crystals were subjected to different downstream processing. In case of adipic acid, stirring rate and gassing parameters during the crystallization procedure were varied, as shown in Table 6.1. Upon reaching the final crystallization temperature, crystals of both chemical systems were filtered and washed. L-alanine experiments further differ in the applied drying procedure, namely no drying (Ala1), static drying in an oven (Ala2) and drying in a fluidized bed (Ala3). Adipic acid crystals were not dried. Upon washing, and, if applicable, drying, crystals were re-suspended in ethanol, in case of L-alanine, or saturated solution, in case of adipic acid, in order to enable dynamic imaging by QICPIC. Similarly to the procedure described for the experiments conducted in Magdeburg, the solution was pumped through the cuvette where the images were taken with 25 frames/s. Note that the experiments in Magdeburg were monitored on-line by periodically withdrawing the solution, imaging the crystals by QICPIC and returning the solution to the crystallization vessel. Here, the crystals were re-suspended at the end of the experiment and the solution was collected in a separate vessel after imaging in order not to image the same particle twice [74]. On-line measurement or a direct measurement at the end of the experiment was not possible due to the high suspension density of the obtained slurry [74].

As gas bubbles can be observed in such on-line imaging procedures, an additional experiment using only water was performed, where bubble formation was encouraged by allowing the air to be sucked into the measurement setup by slightly opening an appropriate tube [4]. The corresponding experiment is denoted as "Bubble".

Parameter/Chemical system	L-alanine/water			Adipic acid/water		
	Ala1	Ala2	Ala3	AA1	AA2	AA3
Saturation temperature [$^{\circ}\text{C}$]	42.38			50		
Solubility at saturation temperature [$\frac{\text{g}}{\text{kg water}}$]	197.9			80.0		
Start temperature [$^{\circ}\text{C}$]	60					
End temperature [$^{\circ}\text{C}$]	10					
Cooling rate [$\frac{\text{K}}{\text{min}}$]	0.45			0.3		
Stirring rate [rpm]	300			158	158	262
Gas volume flow [$\frac{\text{L}}{\text{h}}$]	-			367	833	367
Supersaturation, where gassing is started [$\frac{\text{g}}{\text{kg water}}$]	-			17.6	2.4	17.6
Duration of gassing process [s]	-			33		
Temperature of wash liquid [$^{\circ}\text{C}$]	25					
Number of wash cycles [-]	2			1		
Amount of wash liquid [mL]	40 */80 **			70		
Drying temperature [$^{\circ}\text{C}$]	-	65 †	68 ‡	-		
Drying duration	-	1 d \diamond	10 min \diamond	-		

Table 6.1: Parameters of the L-alanine cooling crystallization and adipic acid gassing crystallization experiments. † Static drying. ‡ Fluidized bed drying. *Mixture of ethanol/water, volume fraction 4/1. **Pure ethanol. \diamond Rough estimation (process was performed until sample was completely dry). Data provided by Stefan Heisel. Reprinted (adapted) from the supplementary material of [4] with permission from Elsevier.

Chemical system	Data set	d_{50} [μm]	$d_{90} - d_{10}$ [μm]	A_g [-]
L-alanine/water	Ala1	275	218	0.279
L-alanine/water	Ala2	285	226	0.474
L-alanine/water	Ala3	281	226	0.526
Adipic acid/water	AA1	632	623	0.629
Adipic acid/water	AA2	602	503	0.548
Adipic acid/water	AA3	532	435	0.685

Table 6.2: Product characteristics of the examined samples of the chemical systems L-alanine/water and adipic acid/water, calculated in Dortmund using image analysis and preliminary DFA classifiers not presented here. Data provided by Stefan Heisel. Reprinted (adapted) from the supplementary material of [4] with permission from Elsevier.

6.2 Image Processing and Analysis

The obtained QICPIC videos were binarized by the QICPIC software and individual frames were exported using MATLAB. These images were processed in Dortmund by the software ImageJ (version 1.49b). Here, a hole-filling operation was performed and objects that touch the image boundary were deleted, along with objects whose equivalent diameter was smaller than $80\ \mu\text{m}$ [4]. Examples of objects are given in Table 6.3.

19 image descriptors were then computed by the Dortmund research group, using either ImageJ or a MATLAB program. Image descriptor values were provided to the author by Stefan Heisel. A list of used image descriptors is presented in Table 6.4 [4]. Some of these descriptors were also used in a previous study by the Dortmund group [68]. Image descriptors describe either particle size, roughness or proportion, so that they can be divided into these three categories [4], as seen in the right column of the table. An additional category corresponds to the last image descriptor that describes the percentage of white projection area [4], here representing the background as this descriptor is computed on an image where the foreground is black.

Image descriptors computed by MATLAB [4] require additional explanation as they are not standard descriptors. Two of them are based on the concept of concavity points. Similarly as in the 3D case discussed in Chapter 4, concavity points represent locations where primary particles are joined together to form an aggregate. In 2D, concavity points can be detected by measuring the distance between the points on the crystal boundary and the boundary of the image that represents the convex hull of the particle, similarly as in [143] and [151]. This is illustrated in Figure 6.1a, where the boundary of the white object, representing an aggregate, is given in red and the boundary of its convex hull in green. In order to ensure that the distance is measured in the correct direction, both the crystal boundary and the boundary of the convex hull are split into paired segments. These segments are bounded by crystal boundary points that form the convex hull. One such pair of segments is marked by blue and yellow dots in Figure 6.1b, while green points represents the boundaries of the segments. For each point i in a boundary segment, the closest point in the corresponding segment of the convex hull boundary is found



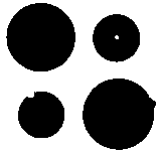


	Single crystals	Agglomerates	Bubbles
L-alanine			
Adipic acid			

Table 6.3: Examples for processed objects (after binarization), imaged by QICPIC. Reprinted (adapted) from [4] with permission from Elsevier.

Descriptor	Symbol	Description	Unit	Type
Area	A	Projection area after hole filling	mm ²	1
Perimeter	P	Length of the object boundary	mm	
Major	L	Major axis length of the best-fit ellipse	mm	
Minor	B	Minor axis length of the best-fit ellipse	mm	
Max Feret	F_{\max}	Maximal Feret diameter	mm	
Min Feret	F_{\min}	Minimal Feret diameter	mm	
Equiv. Diam.*	D_{eq}	Diameter of a coextensive circle, $D_{\text{eq}} = 2 \cdot \sqrt{\frac{A}{\pi}}$	mm	
Circularity	c	Similarity with a circle, $c = \frac{4 \cdot \pi \cdot A}{P^2}$	-	2
Solidity	s	Ratio of a projection area to convex hull area, $s = \frac{A}{A_{\text{convex}}}$ †	-	
Convex box ratio*	a_{LB}	Ratio between convex hull area and ellipse axes, $a_{LB} = \frac{A_{\text{convex}}}{L \cdot B}$ †	-	
NSwC*	$NSwC$	See ref. [68], $NSwC = \frac{2 \cdot A \cdot P}{F_{\max} \cdot L \cdot B} - \frac{4 \cdot A^2}{\pi \cdot L^3 \cdot B}$	-	
Num. conc. points**	n_{conc}	Number of concavity points, see Figure 6.1	-	
Max depth**	d_{max}	Scaled max. concavity depth, see eq. 6.1	-	
Aspect ratio	a_r	Aspect ratio, $a_r = \frac{L}{B}$	-	3
Area ratio*	f_{Deq}	Ratio of max. Feret diam. and equiv. diam., $f_{\text{Deq}} = \frac{F_{\max}}{D_{\text{eq}}}$	-	
Elongation*	e	Particle elongation, $e = \frac{F_{\max}}{F_{\min}}$	-	
Roundness	r	Squared ratio of equiv. diam. and major ellipse axis, $r = \left(\frac{D_{\text{eq}}}{L}\right)^2$	-	
Centroid var.**	d_{cent}	Centroid distance variation, see eq. 6.3	-	
Area fraction	a_f	Percentage of white projection area	%	4

Table 6.4: All 19 image descriptors, their definitions and their type: 1. "Size"; 2. "Roughness"; 3. "Proportion"; 4. Area fraction. † A_{convex} is the convex hull area which is used for calculation but not used as an additional image descriptor. *Calculations based on values provided by ImageJ. **Calculated with MATLAB. Reprinted (adapted) from [4] with permission from Elsevier.

and the distance d_i between these points is computed. This distance d_i represents the concavity depth of the point i . A peak search over the smoothed distances results in concavity points, while a minimal depth of two pixels is required. The curve with distances is smoothed using a moving average filter of length 5. In Figure 6.1b, found concavity points are shown as blue circles, while the corresponding nearest points at the convex hull boundary are shown as yellow rhomboids. Image descriptor "Num. conc. points" is the number of so found concavity points for an object. Descriptor "Max Depth" is the scaled maximal obtained concavity depth, defined as [4]

$$d_{\max} = \frac{\max_i d_i}{D_{\text{eq}}}, \quad (6.1)$$

where d_i is the concavity depth, and D_{eq} is the area-equivalent circle diameter, computed by MATLAB.

The third non-standard image descriptor is the centroid variation ("Centroid var.") and is similar to the "inner distance descriptor" of Huo et al. [70]. For each boundary point with coordinates (x_i, y_i) , a distance to the object centroid, given by coordinates $x_{C,i}, y_{C,i}$ of the arithmetic mean of all boundary points, is computed:

$$d_{C,i} = \sqrt{(x_i - x_{C,i})^2 + (y_i - y_{C,i})^2}. \quad (6.2)$$

The centroid coordinates are obtained using the MATLAB function `regionprops`. The centroid variation d_{cent} is given by [4]:

$$d_{\text{cent}} = \frac{d_{C,\text{std}}}{\bar{d}_C}, \quad (6.3)$$

where $d_{C,\text{std}}$ and \bar{d}_C are the standard deviation and mean of the values $d_{C,i}$. The difference with respect to the work of Huo et al. [70] is that the value $d_{C,\text{std}}$ is used directly, instead of the standard deviation of $d_{C,i} - \bar{d}_C$.

6.3 Training and Test Set Creation

Previous section described the image descriptor data provided by Stefan Heisel, TU Dortmund University. This data is used in the classification procedure, where the chosen method creates a classifier based on a given manually labeled training set. The accuracy of the procedure is investigated on manually labeled test sets. In order to create training and test sets, in each of the six crystal data sets, 600 objects were manually labeled as single crystals and 600 objects were labeled as aggregates. Additionally, 1200 bubbles in the bubble data set were manually labeled, as well as all observed bubbles in the crystal data sets. Labeling was performed by the Dortmund research group.

In collaboration with the research group from Dortmund, two types of training ("TR") and test ("TE") sets were created and are presented in Table 6.5 [4]. Sets of type "S" use only one crystal data set for training. The same 600 single crystals and 600 aggregates are used in the corresponding test set, along with the bubbles from the considered data set. Sets of type "C" combine data from all three crystal data sets in each training set. The number of objects was varied in order to test its influence on

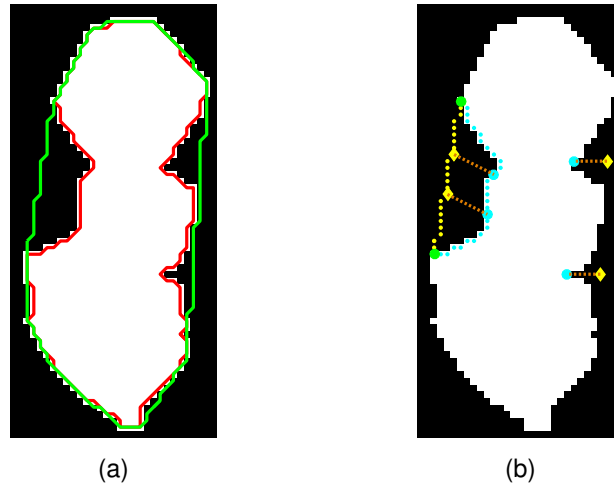


Figure 6.1: Finding concavity depth and concavity points. (a) Red line: object boundary. Green line: boundary of the convex hull image. (b) Yellow dotted line: one segment of the convex hull boundary. Large green circles: ends of the convex hull boundary segment. Light blue dots: one object boundary segment. Large light blue circles: concavity points. Yellow rhomboids: corresponding nearest points on the convex hull boundary. Orange dotted lines: correspondences between boundary points and the nearest convex hull points; their length is the concavity depth. Reprinted from [4] with permission from Elsevier.

the classification accuracy. In each crystal data set, first N objects of one type are placed in the training set, where N is varied between 3 and 200. The corresponding test set contains the last 400 objects of each type in order not to overlap with the training set.

First 600, or the first $3N$ bubbles from bubble data sets are used in both "S" and "C" training sets, as not enough bubbles were present in the crystal data sets. Finally, a bubble test set containing only the second 600 bubbles from the "Bubble" data set is defined. The created two types of training and test sets enable checking whether classification accuracy improves upon combining data from different experiments.

Classification accuracy was determined using the same approach as in the literature [68, 105, 106]. The performance index PI_{All} measures the amount of correctly classified objects [4, 68, 105, 106]:

$$PI_{All} = \frac{N_{\text{correctly classified objects}}}{N_{\text{objects}}}. \quad (6.4)$$

Performance can also be measured with respect to one of the three considered classes. Thus, accuracy with respect to bubbles can be defined using either bubbles from the crystal data sets or bubbles from the bubble test set. The latter is defined as [4]:

$$PI_{\text{Bubbles-bubble data set}} = \frac{N_{\text{correctly classified bubbles from bubble test set}}}{N_{\text{bubbles in bubble test set}}}. \quad (6.5)$$

An average amount of correctly classified bubbles from crystal data sets is similarly defined and denoted

Data set	Ala1			Ala2			Ala3			Bubble
Name	s	a	b	s	a	b	s	a	b	b
C-Ala- N -TR	N	N	0	N	N	0	N	N	0	$3N$
C-Ala1-TE	400	400	22	0	0	0	0	0	0	0
C-Ala2-TE	0	0	0	400	400	7	0	0	0	0
C-Ala3-TE	0	0	0	0	0	0	400	400	3	0
S-Ala1-TR	600	600	0	0	0	0	0	0	0	600
S-Ala2-TR	0	0	0	600	600	0	0	0	0	600
S-Ala3-TR	0	0	0	0	0	0	600	600	0	600
S-Ala1-TE	600	600	22	0	0	0	0	0	0	0
S-Ala2-TE	0	0	0	600	600	7	0	0	0	0
S-Ala3-TE	0	0	0	0	0	0	600	600	3	0
Data set	AA1			AA2			AA3			Bubble
C-AA- N -TR	N	N	0	N	N	0	N	N	0	$3N$
C-AA1-TE	400	400	0	0	0	0	0	0	0	0
C-AA2-TE	0	0	0	400	400	0	0	0	0	0
C-AA3-TE	0	0	0	0	0	0	400	400	10	0
S-AA1-TR	600	600	0	0	0	0	0	0	0	600
S-AA2-TR	0	0	0	600	600	0	0	0	0	600
S-AA3-TR	0	0	0	0	0	0	600	600	0	600
S-AA1-TE	600	600	0	0	0	0	0	0	0	0
S-AA2-TE	0	0	0	600	600	0	0	0	0	0
S-AA3-TE	0	0	0	0	0	0	600	600	10	0
Bubble-TE	0	0	0	0	0	0	0	0	0	600

Table 6.5: "Combined" ("C") and "separate" ("S") training ("TR") and test ("TE") set configurations for L-alanine ("Ala") and adipic acid ("AA"). "s" refers to single crystals, "a" to aggregates, "b" refers to bubbles, and N is the number of objects in the set (N varies between 3 and 200). Reprinted (adapted) from [4] with permission from Elsevier.

by $\overline{PI}_{\text{Bubbles-crystal data sets}}$.

The aim of the classification procedure is to measure the agglomeration degree, defined as [4, 68]:

$$Ag = \frac{N_{\text{aggregates}}}{N_{\text{aggregates}} + N_{\text{single crystals}}}. \quad (6.6)$$

Note that $Ag = 0.5$ for all training and test sets defined in this study, in order to avoid bias towards either single crystals or aggregates. The error in the measurement of the agglomeration degree is defined as [4, 68]:

$$\delta Ag = \frac{|Ag_{\text{calculated}} - Ag_{\text{manual}}|}{Ag_{\text{manual}}}, \quad (6.7)$$

where Ag_{manual} is the agglomeration degree obtained using manual classification, while $Ag_{\text{calculated}}$ is obtained using the created automatic classifier. A classification result can be considered good if PI is close to one and δAg close to zero.

6.4 Image Descriptor Ranking

In section 2.3.4.3, the problem of choosing a subset of image descriptors in order to avoid the peaking phenomenon was introduced. Peaking phenomenon describes a decline in classification accuracy with an increase in the number of used variables, in this case image descriptors. Therefore, as discussed in the review chapter 2, a significant amount of effort can be invested in selecting a suitable variable subset. The research group in Dortmund chose the statistical measure called proportional similarity to quantify the quality of image descriptors [4]. Proportional similarity PS gives a value for the similarity of two discrete distributions computed over the same variable. A detailed description can be found in the work of Vegelius et al. [152]. In order to quantify the quality of an image descriptor, distributions over that image descriptor are compared pairwise, using different data. This is briefly explained in the following and given in more details in the joint publication [4]. Image descriptors must be scaled to a range $[0, 1]$ beforehand. Values PS_{sa} are obtained when comparing the distributions for single crystals and crystal aggregates, with 600 objects per class. Values PS_{cb} are computed by comparing distributions for crystals, comprising both aggregates and single particles, to distributions of gas bubbles, so that there are 1200 objects per class. Both PS_{sa} and PS_{cb} are computed for every crystal data set and averaged over different data sets for each chemical system, which is denoted by the bar over the PS value. Finally, values $\overline{PS}_{\text{sa}}$ and $\overline{PS}_{\text{cb}}$ are averaged to obtain the values $\overline{PS}_{\text{mn}}$, capturing both the single crystal/aggregate and the crystal/bubble differences.

For each chemical system, a ranking of the image descriptors can be performed based on the given $\overline{PS}_{\text{sa}}$, $\overline{PS}_{\text{cb}}$ and $\overline{PS}_{\text{mn}}$ values. The rankings were provided by the research group from Dortmund and are given in Table 6.6 [4]. Here, image descriptor with the rank 1 is supposed to have the highest influence on the classification accuracy as the underlying distributions differ the most, while the descriptor with rank 19 is thought to have the lowest influence. Classification is then performed by selecting the first N_{desc} image descriptors from the chosen ranking type, where N_{desc} is varied to study the influence of the

number of image descriptors on the fit quality.

6.5 ANN

A feedforward artificial neural network consists of interconnected nodes, divided into layers. The network is trained by estimating the node-connecting weights and biases so that the desired output on the training set is matched as closely as possible. Biases can be understood as weights that are only connected to one node. Networks were created by the MATLAB function `patternnet` and trained by the function `train`, where more details about these functions can be found in MATLAB documentation [153]. Here, each network contained three layers: an input layer, a hidden layer and an output layer, as depicted in Figure 6.2. The input layer contains as many neurons N_{desc} as there are image descriptors and receives the input data, where the input data is automatically scaled to the interval $[-1, 1]$. The hidden layer can contain a variable number of nodes, where 10 was finally chosen in this study, as it will be discussed in section 6.6. Each node j in the hidden layer obtains as an input h_j the weighted sum of the scaled image descriptor values x_i :

$$h_j = \sum_{i=1}^{N_{\text{desc}}} w_{I,i,j} x_i + b_{H,j}, \quad (6.8)$$

where $w_{I,i,j}$ are the weights connecting the i -th input to the j -th hidden neuron and $b_{H,j}$ are the bias values. The hidden neuron j computes a non-linear "tansig" function of these inputs to produce the output y_j^h [153]:

$$y_j^h = \text{tansig}(h_j) = \frac{2}{1 + e^{-2h_j}} - 1. \quad (6.9)$$

Finally, the output neuron k obtains a weighted sum s_k of values y_j^h as input:

$$s_k = \sum_{j=1}^{N_{\text{hidden}}} w_{H,j,k} y_j^h + b_{O,k}, \quad (6.10)$$

where $w_{H,j,k}$ are the weights connecting the hidden layer neurons to the output neurons and $b_{O,k}$ are the corresponding biases. There are three output neurons, one for each of the three considered classes. The output neuron k computes a "softmax" function of the values s_k [153][119, p. 236]:

$$y_j^o = \text{softmax}(s_k) = \frac{e^{s_k}}{\sum_{k=1}^{N_{\text{output}}} e^{s_k}}. \quad (6.11)$$

This ensures that the sum of the output values is 1. These output values indicate the probability that some object is in the class j , so that the object is finally classified into the class j with the highest y_j^o .

The training process consists of finding the weight and bias values that minimize the mean cross-

No.	Ala-"sa"	Ala-"cb"	Ala-"mn"	AA-"sa"	AA-"cb"	AA-"mn"
1	d_{\max}	f_{Deq}	c	n_{conc}	f_{Deq}	c
2	n_{conc}	d_{cent}	d_{cent}	d_{\max}	d_{cent}	n_{conc}
3	c	r	n_{conc}	F_{\min}	r	r
4	a_{LB}	c	r	P	c	F_{\max}
5	s	e	d_{\max}	B	e	d_{\max}
6	$NSwC$	a_f	f_{Deq}	$NSwC$	a_r	e
7	d_{cent}	a_r	a_r	F_{\max}	a_f	P
8	r	F_{\max}	e	D_{eq}	F_{\max}	L
9	a_r	L	a_f	A	L	a_r
10	f_{Deq}	P	s	s	P	D_{eq}
11	P	n_{conc}	a_{LB}	c	A	A
12	L	D_{eq}	$NSwC$	L	D_{eq}	d_{cent}
13	F_{\min}	A	F_{\max}	a_{LB}	s	f_{Deq}
14	F_{\max}	F_{\min}	L	e	n_{conc}	F_{\min}
15	D_{eq}	B	P	r	d_{\max}	s
16	e	d_{\max}	D_{eq}	a_r	F_{\min}	B
17	B	s	F_{\min}	a_f	B	$NSwC$
18	a_f	$NSwC$	A	d_{cent}	$NSwC$	a_f
19	A	a_{LB}	B	f_{Deq}	a_{LB}	a_{LB}

Table 6.6: Different \overline{PS} ranking orders for L-alanine (Ala) and adipic acid (AA) crystals, based on data provided by Stefan Heisel [4]. "sa" and "cb" refer to the \overline{PS}_{sa} and \overline{PS}_{cb} values obtained from discrimination of single crystals/agglomerates and crystals/bubbles, respectively. "mn" is the arithmetic mean of "sa" and "cb". The ranking is from best to worst.

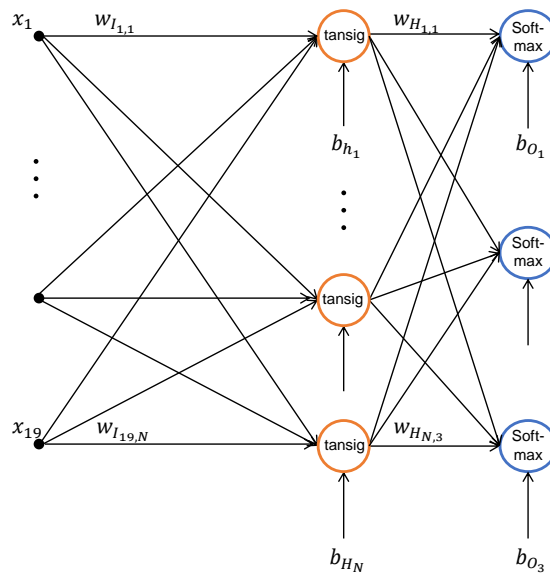


Figure 6.2: Artificial neural network used in this study, containing one input, one hidden and one output layer.

entropy objective function [119, p. 235], computed for each object by [153]:

$$L_E = \frac{1}{N_{\text{output}}} \sum_{j=1}^{N_{\text{output}}} -t_j \cdot \ln(y_j^o). \quad (6.12)$$

Here, t_j are the target outputs of the training set, containing a value 1 for the output neuron j corresponding to the correct class, and values 0 otherwise. This optimization problem is solved by the scaled conjugate gradient algorithm and uses the principle of backpropagation [121]. In order to prevent over-fitting, MATLAB splits the obtained training set internally into a training, test, and validation sets containing 70%, 15%, and 15% of the original training set size. The error obtained on the validation set is monitored so that the training process can be stopped before overfitting occurs [153][119, p. 259-261], while the test set is used to evaluate the classification quality. The output of this test set is ignored and the test sets defined in Table 6.5 are used to enable comparison with the classification by DFA, performed in Dortmund. The output of the training process depends on the initial position in the parameter space when performing parameter optimization. In order to introduce reproducibility, training is repeated 10 times for different random number seeds.

6.6 Results

6.6.1 ANN Size

The used ANN structure with one hidden layer is validated by varying the number of neurons in the hidden layer between 1 and 10 [4]. All 19 image descriptors and the training set "C-Ala-200-TR" are chosen for the analysis. The computation is repeated 10 times using 10 different random number seeds, ensuring different starting points for each of the 10 optimization procedures. The results are shown in Figure 6.3. The error bars represent the standard deviation obtained when repeating the training 10 times. The worst classification is obtained with one hidden neuron. A setup with one hidden neuron results in a linear classifier. The large error bars suggests that even in this case, a good classification accuracy can be obtained for some starting points. This is consistent with the satisfactory results using DFA [4], which is also a linear classifier. It can be seen that the improvement in accuracy stops when using more than three hidden neurons. Small error bars for more than three hidden neurons further indicate that the result does not depend strongly on the chosen random starting point. We conclude that no significant difference in accuracy is expected when using between four and ten neurons. Ten is chosen for the further studies as no significant increase in computation cost was observed. Furthermore, the results indicate that no improvement is to be expected when further increasing the number of neurons in the hidden layer.

In the following, results for one chosen random number seed are shown for each considered training scenario. However, if the ANN is to be used to evaluate newly obtained data sets, it is recommended to choose a specific scenario and repeat the training several times. The ANN leading to the highest

accuracy for the chosen training scenario should then be selected for evaluating newly obtained data sets [4].

6.6.2 Combined Data Sets

The research group in Dortmund developed a criterion for finding the best ranking order and the best number of image descriptors [4], explained in the following. Training sets C-Ala-200-TR and C-AA-200-TR were chosen for the analysis as these were the largest training sets of type "C". For each ranking order and each chemical system, N_{desc} best-ranked descriptors were used, where N_{desc} was successively increased from 1 to 19. Classification was considered satisfactory if $\overline{PI}_{\text{All}} \geq 0.9$, where the bar over the value indicates that the values were averaged over the considered test sets. The results, obtained for one chosen random number seed, are presented in Figures 6.5 and 6.6 for L-alanine and adipic acid, respectively. The finally chosen best number of image descriptors N_{desc} is the smallest N_{desc} for which the following is fulfilled, namely $\overline{PI}_{\text{All}}(N_{\text{desc}}) \geq 0.9$ and

$$\frac{|\overline{PI}_{\text{All}}(N_{\text{desc}} + 1) - \overline{PI}_{\text{All}}(N_{\text{desc}})|}{\overline{PI}_{\text{All}}(N_{\text{desc}})} < 0.02, \quad (6.13)$$

meaning that $\overline{PI}_{\text{All}}$ did not change for more than 2% when an additional image descriptor was introduced. The chosen N_{desc} is presented in Table 6.7, along with the values regarding DFA, as computed by the Dortmund research group and presented in the joint publication [4].

It can be seen that $\overline{PI}_{\text{All}} \geq 0.9$ is achieved rather quickly for all image descriptor rankings, where three to six image descriptors are needed to fulfill the final criterion. The best image descriptor ranking is "sa", in case of L-alanine, and "mn", in case of adipic acid. In both cases, the first three image descriptors lead to the desired results. Interestingly, these are the best ranking orders in case of DFA, as well [4]. The worst image descriptor ranking is "cb", obtained by comparing crystals to bubbles. Here, four or six image descriptors are needed. The best-ranked image descriptors in case of "cb" give less information regarding whether a particle is an aggregate in comparison to image descriptor

	DFA	ANN
Ala	sa: 10	sa: 3
	cb: 18	cb: 6
	mn: 11	mn: 5
AA	sa: 13	sa: 4
	cb: 10	cb: 4
	mn: 8	mn: 3

Table 6.7: Lowest number of image descriptors N_{desc} needed so that $\overline{PI}_{\text{All}} \geq 0.9$ and $\overline{PI}_{\text{All}}$ does not change for more than 2% upon adding an additional image descriptor. Results are presented for different image descriptor ranking orders. Computations regarding DFA are provided by Stefan Heisel. Ranking orders leading to the smallest number of needed descriptors are presented with bold letters. Reprinted (adapted) from [4] with permission from Elsevier.

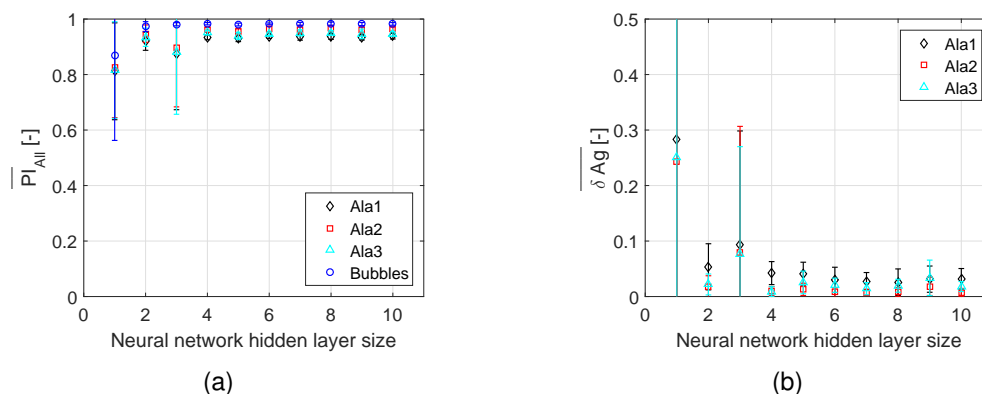


Figure 6.3: Average classification accuracy obtained for different numbers of neurons in the hidden layer. Error bars represent standard deviation obtained upon repeating the training process 10 times for each number of hidden neurons. Classification was performed using the training set "C-Ala-200-TR" and the corresponding test sets of type "C". All 19 image descriptors were employed. Reprinted from [4] with permission from Elsevier.

rankings "sa" and "mn". For each ranking, the finally chosen N_{desc} is significantly smaller in case of ANN than when using DFA. This confirms the hypothesis that ANN outperforms DFA in terms of classification accuracy.

Figure 6.5 shows a significantly lower accuracy in case of bubbles from crystal data sets in comparison to the accuracy for all objects or bubbles from the bubble test set. This can be explained by observing the objects that were labeled as bubbles [4]. Bubbles from crystal data sets, exemplified in Figure 6.4 often differ to the bubbles from the "Bubble" data set, shown in Table 6.3. Note that the training set contained objects from the "Bubble" data set. In crystal data sets, bubbles are often oval or may be attached to crystals, while those from the "Bubble" data set are typically round. Furthermore, bubbles in the "Bubble" data set were typically completely black, as shown in Table 6.3, while bubbles from the crystal data sets occasionally contained white parts in the middle, causing a different value in the image descriptor "Area fraction" [4]. This issue regarding lower accuracy was not observed in case of adipic acid, as seen in Figure 6.6. Adipic acid data sets contained a much smaller number of bubbles which, furthermore, did not show the discussed defects. In case of L-alanine, ranking order "sa",

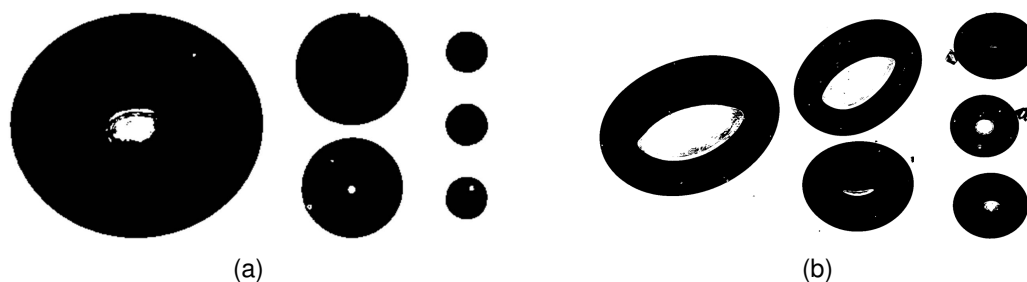


Figure 6.4: Bubbles from data set Ala1 that were correctly (a) and wrongly (b) classified using DFA, as observed by the Dortmund research group [4]. Reprinted from [4] with permission from Elsevier.

$\overline{PI}_{\text{Bubbles-crystal data sets}}$ decreases for more than five image descriptors used. This can be understood as evidence of overfitting and was also observed for all ranking orders in case of DFA [4].

There is some oscillation in the performance index values for an increasing number of image descriptors. The values $\overline{PI}_{\text{Bubbles-bubble test set}}$ and $\overline{PI}_{\text{Bubbles-crystal data sets}}$, in case of ranking order "sa", drop significantly when using two or three image descriptors, only to increase again for a larger number of descriptors. This behavior could be understood as evidence that a combination of individually good image descriptors may not necessarily provide an overall improvement in classification accuracy. However, it is important to note that the ranking order "sa" by design ensures a good classification of single crystals/aggregates and not of bubbles.

Upon agreeing on the best ranking, presented in bold in Table 6.7, a study regarding the training set size was performed for the varying number of image descriptors. This further enabled validating the quality criterion for image descriptor rankings developed at TU Dortmund University and explained above. Here, the number of objects N per class in the training set was varied. A training set is considered to be sufficiently large if $\delta Ag < 0.1$ and $\overline{PI}_{\text{All}} \geq 0.9$ [4]. The results are presented in Figures 6.7 and 6.8, in case of L-alanine and adipic acid, respectively. Here, the gray bar indicates the chosen best number of image descriptors, as provided in Table 6.7. It can be seen that the accuracy improves with the increasing number of objects in the training set. In case of L-alanine, the criterion $\overline{PI}_{\text{All}} \geq 0.9$ and $\delta Ag < 0.1$ is fulfilled for 90 objects in the training set already. However, as this is not fulfilled for $N_{\text{total}} = 180$ objects, it is considered that at least 360 objects are needed to achieve this accuracy. No significant improvement in accuracy is observed upon further increasing the training set size. Results regarding adipic acid show that the criteria is fulfilled already for $N_{\text{total}} = 180$, as well as all larger training sets. In case of only 27 objects in a training set, peaking phenomenon is evident in case of adipic acid where PI_{All} decreased and ΔAg increased with the increasing number of image descriptors. Therefore, the hypothesis stating that a larger training set improves the classification accuracy is confirmed. Similar results are obtained in case of DFA [4].

6.6.3 Separate Data Sets

The study using training sets of type "C", provided in the previous section, has shown that the constructed ANNs are capable of accurately classifying imaged objects even for a small number of image descriptors and a small number of objects in the training set. However, these training sets combined objects from data sets obtained under different experimental conditions. The aim of the following study is to evaluate whether a classifier created using only one data set can be transferred to data obtained under different experimental conditions. This is achieved using the training and test sets of type "S". Image descriptor ranking and the number of image descriptors were chosen according to Table 6.7, while the training sets contained 1800 objects, as given in Table 6.5. The results are shown in Figures 6.9 in case of L-alanine and Figure 6.10 in case of adipic acid. It can be seen that a satisfactory accuracy is achieved in each case. No trend showing improvement in accuracy when training and test set data is derived from the same experiment is observed. Therefore, it is possible to transfer a classifier

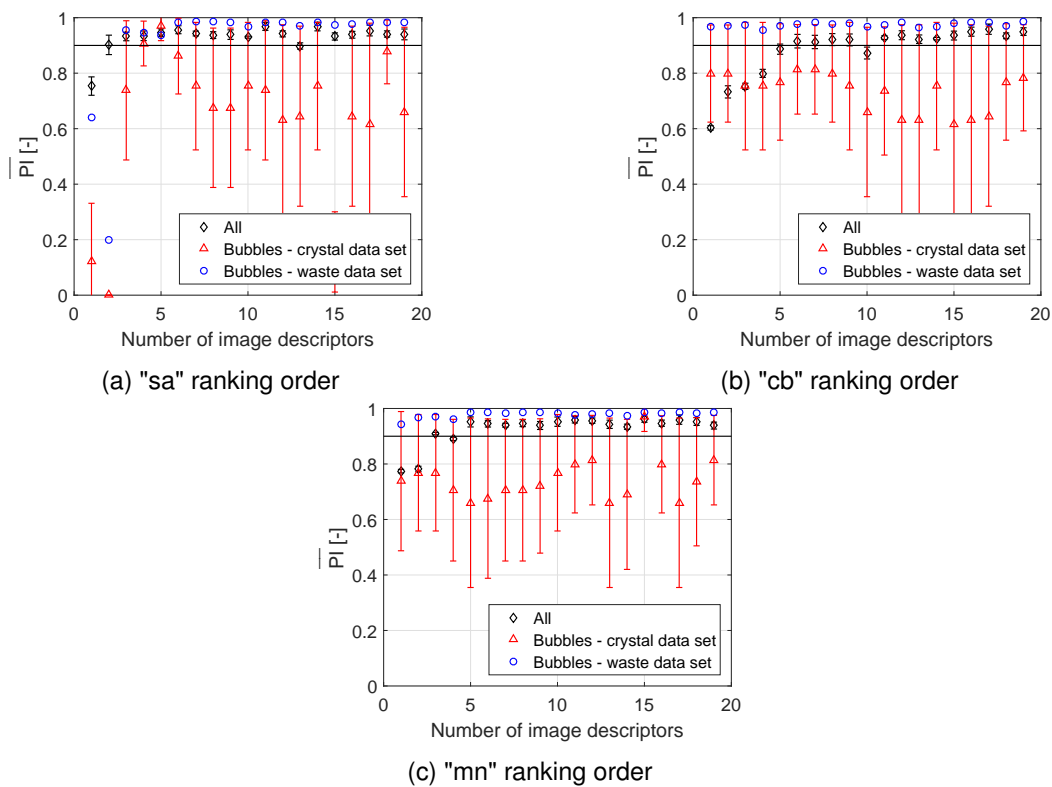


Figure 6.5: ANN results based on the training set C-Ala-200-TR using different numbers of image descriptors. Image descriptors are ranked according to "sa" (a), "cb" (b) as well as "mn" (c). The error bars represent the standard deviation of the values for different test sets. The target value $\bar{P} = 0.9$ is visualized by a horizontal line. Reprinted from the supplementary material of [4] with permission from Elsevier.

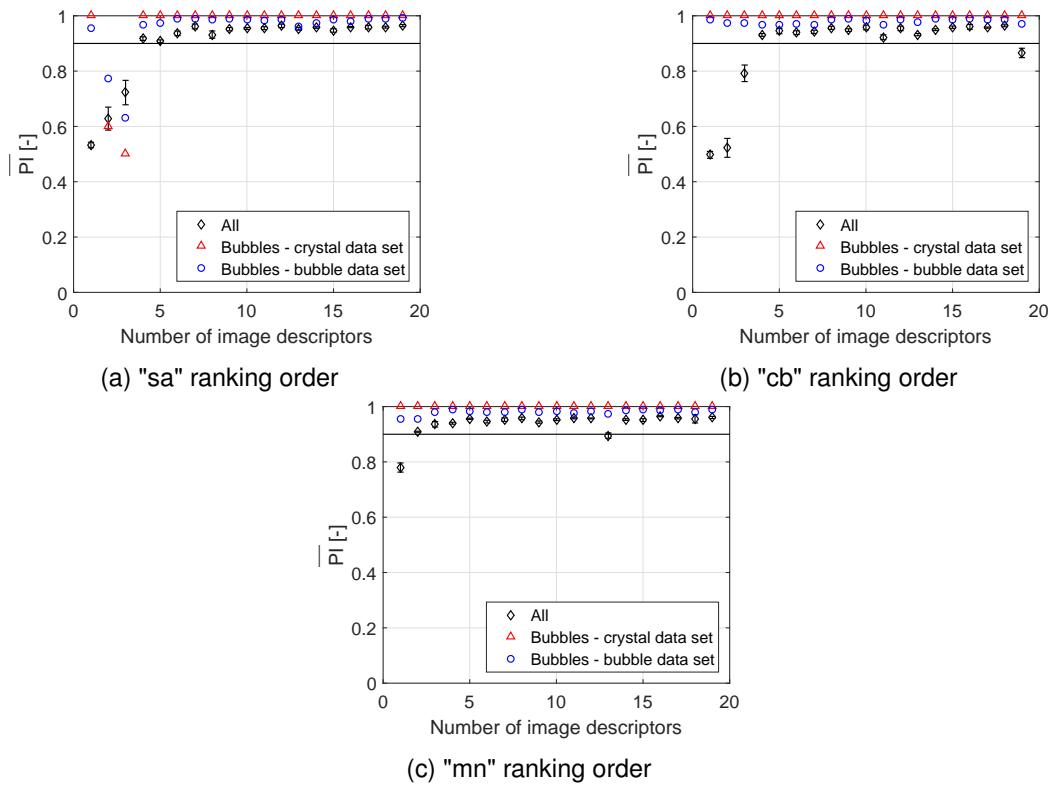


Figure 6.6: ANN results based on the training set C-AA-200-TR using different numbers of image descriptors. Image descriptors are ranked according to "sa" (a), "cb" (b) as well as "mn" (c). The error bars represent the standard deviation of the values for different test sets. The target value $\overline{PI} = 0.9$ is visualized by a horizontal line. Reprinted from the supplementary material of [4] with permission from Elsevier.

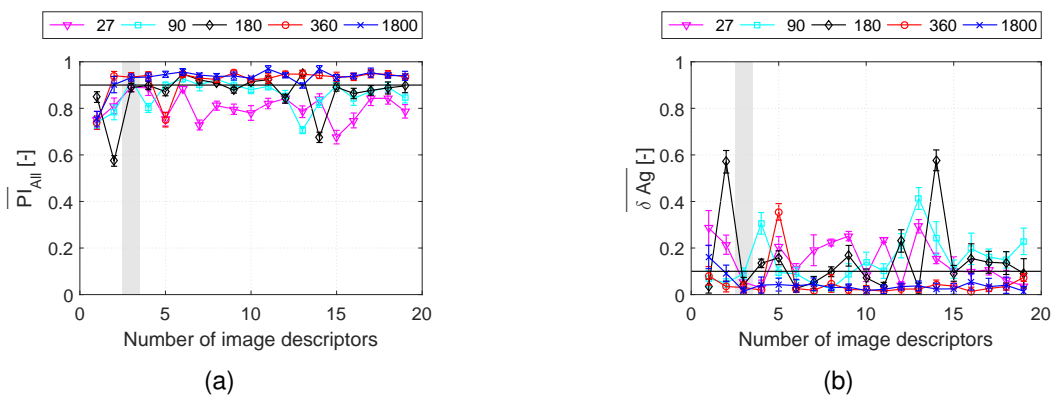


Figure 6.7: ANN results based on the training set C-Ala- N -TR using different training set sizes ($N_{\text{total}} = 9N = 27, 90, 180, 360, 1800$) and numbers of image descriptors; for better visualization, the image descriptor subset selected ("sa" ranking order) is highlighted with a gray bar. The error bars represent the standard deviation of the values for different test sets. The target values $\overline{PI}_{\text{All}} = 0.9$ and $\delta Ag = 0.1$ are visualized by solid horizontal lines. Reprinted from [4] with permission from Elsevier.

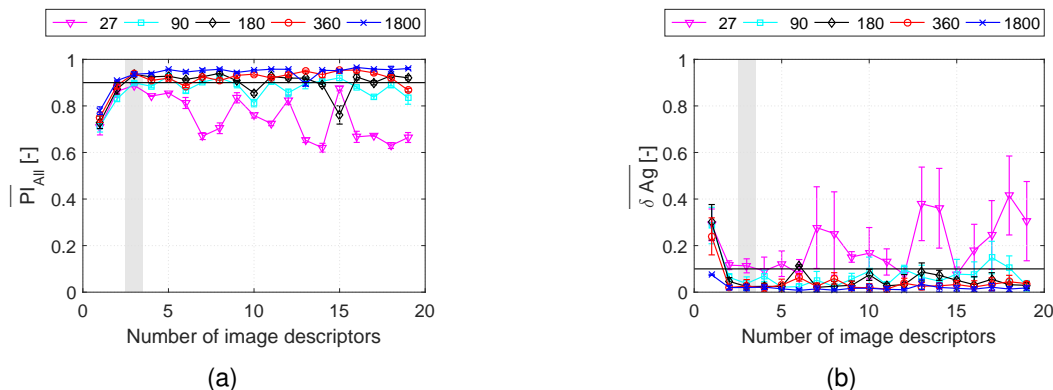


Figure 6.8: ANN results based on the training set C-AA-n-TR using different training set sizes ($N_{total} = 9N = 27, 90, 180, 360, 1800$) and numbers of image descriptors; for better visualization, the image descriptor subset selected ("mn" ranking order) is highlighted with a gray bar. The error bars represent the standard deviation of the values for different test sets. The target values $\overline{PI}_{All} = 0.9$ and $\delta Ag = 0.1$ are visualized by solid horizontal lines. Reprinted from [4] with permission from Elsevier.

to data obtained under different experimental conditions, at least regarding the experimental conditions used in this study. The same conclusion is reached using DFA [4]. This contradicts the last hypothesis stated at the beginning of this chapter.

6.7 Conclusion

The results presented in the previous section show that it is possible to create an ANN classifier which automatically sorts the QICPIC-imaged objects into single crystals, aggregates and bubbles based on the computed set of image descriptors. The classification accuracy depends on the chosen image descriptors and the method for ranking image descriptors developed in Dortmund was validated for the ANN classifier. Using this ranking, it was shown that as few as three image descriptors can lead to a satisfactory accuracy with $\overline{PI} \geq 0.9$ and $\delta Ag < 0.1$. The classification accuracy improves with a larger training set, where at least 360 objects are necessary to achieve this criterion. It was further shown that the classifier created using data obtained under one set of experimental conditions can be transferred to data obtained under a different set of experimental conditions. It should be noted that both the training and test sets contained the same number of single crystals and aggregates, so that different agglomeration degrees of these data sets did not influence the ability of the classifier to learn a certain class. The varied experimental conditions concerned post-processing, in case of L-alanine, and crystallization parameters, in case of adipic acid.

Similar findings were achieved in case of DFA, as discussed further in the joint publication [4] with the Dortmund research group, while ANN classifiers outperform DFA. This is best illustrated by considering the number of image descriptors necessary to achieve $\overline{PI} \geq 0.9$ and $\delta Ag < 0.1$. While only three descriptors are needed in case of ANN, eight or ten descriptors must be used in case of DFA. It is assumed that this behavior is caused by the fact that the data is not linearly separable. This cannot

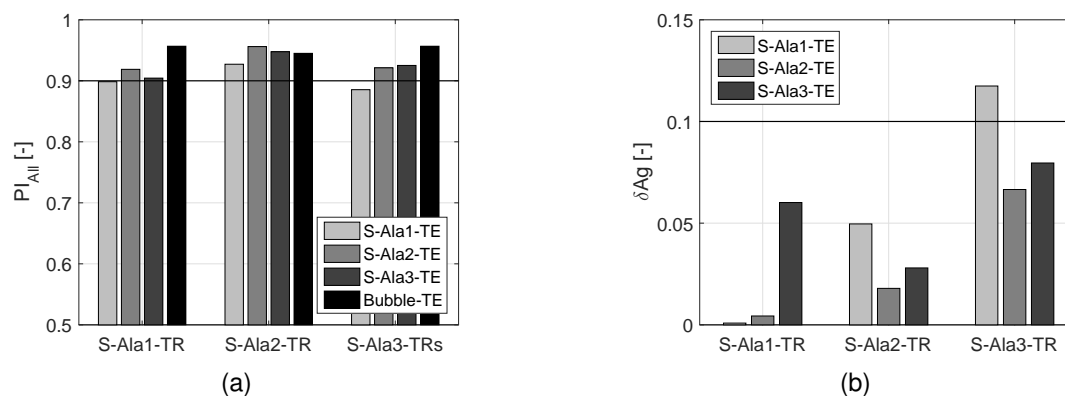


Figure 6.9: ANN results based on the "separate training and test sets" for L-alanine, denoted by "S" in Table 6.5. First three image descriptors from the ranking order "sa" were selected for the analysis. Reprinted from [4] with permission from Elsevier.

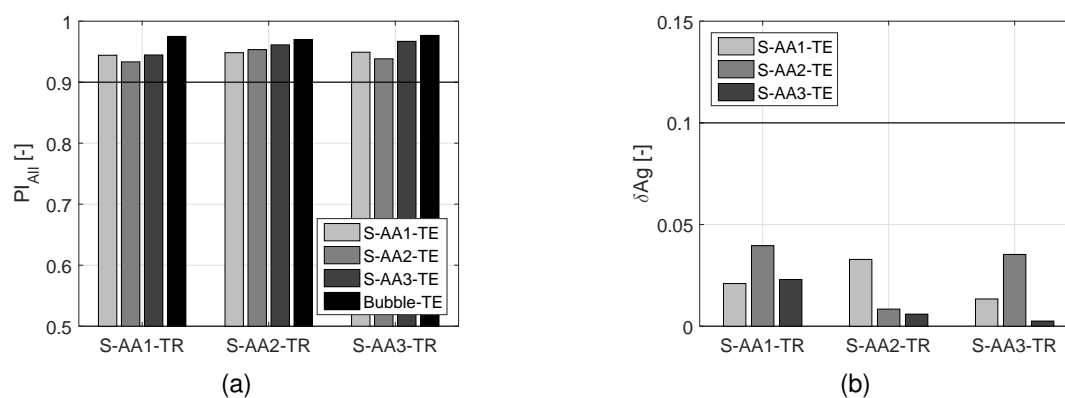


Figure 6.10: ANN results based on the "separate training and test sets" for adipic acid, denoted by "S" in Table 6.5. First three image descriptors from the ranking order "mn" were selected for the analysis. Reprinted from [4] with permission from Elsevier.

be verified by simple visualization, due to the high dimensionality of the problem, represented by the number of image descriptors. Comparison of ANN and DFA [4] further yields the conclusion that the behavior for an increasing training set size is similar, where DFA shows less oscillation in the measures of classification accuracy, thus making the analysis easier. Furthermore, the comparison has shown no significant increase in computational costs is obtained when using ANN [4]. DFA is advantageous in terms of implementation as no parameters regarding the classifier structure are necessary, where in case of ANN, the number of layers and neurons must be determined. However, default parameters suggested by MATLAB, consisting of one hidden layer with 10 neurons, have already given a satisfactory accuracy in the considered case. In case of ANN, the result also depends on the chosen optimization starting point, typically affected by a random number seed, so that it is recommended to repeat the computations several times to ensure no significant improvement can be achieved for a different starting point.

Chapter 7

Conclusion and Outlook

This work presented methods for characterizing the shape of both single crystals and crystal aggregates using 3D microcomputed tomography (μ CT) imaging and dynamic 2D imaging by QICPIC. The 3D imaging method provides full 3D shape information and allows the identification of the full crystal polytope, as presented in chapter 3 and published in [1]. The algorithm identifies crystal faces using Hough transform and matches the faces to those in a crystal model. Finally, symmetry conditions may be applied to yield the best possible fit regarding the assumed shape model. The method was validated on a set of simulated and experimentally measured potash alum crystals. It was shown that the success of the method depends on the shape of the crystals, as well as on the chosen algorithm parameters.

In chapter 4, the procedure was extended to primary particles in crystal aggregates. The procedure was published in [2]. In a first step, the aggregate was segmented into primary particles. This is based on a watershed transform, where the obtained watershed regions are concatenated under consideration of concavity points. Shape identification of each primary particle followed, where the method from chapter 3 was adapted to cases where parts of the particle are not visible. Finally, a set of post-processing steps attempted to resolve issues regarding non-ideal cases. The validation study using simulated crystal aggregates has shown that the method is capable of identifying the shape of each primary particle with sufficient accuracy in 84% to 95% of aggregates. A further validation study has shown that the scheme is capable of identifying the shape of primary particles in experimentally obtained crystals in most cases, as well. Overall, the accuracy of the method decreases with increasing particle overlap and an increasing number of primary particles, while issues can further be caused by crystal asymmetry and lightly touching aggregates.

In chapter 5, this method was extended and used to measure the orientation between primary particles in potash alum aggregates. Samples of crystals were taken from four batch cooling crystallization experiments, performed in cooperation with the group of Prof. Sundmacher, OVGU Magdeburg. Cooling rate was varied in order to obtain different supersaturation profiles. Supersaturation was assumed to have an influence on the orientation between primary particles as it influences the ability of the particles to grow together and form a stable aggregate upon collision in the reactor. Segmentation issues were resolved by allowing the user to manipulate concavity points and region concatenation if not sat-

ified with the result of the automatic procedure. Furthermore, shape identification was adapted to the asymmetrical crystals viewed in these experiments. The orientation between a pair of primary particles was described by a disorientation angle. This is the smallest angle of rotation that leads to an overlap between two particles of the same size and the same center. In order to exclude aggregates that could have occurred during sampling and preparation, primary particle pairs were classified according to their contact type into: "grown-in", "lightly-touching" and "none", the latter meaning that particles are not in contact. Crystal samples were described by disorientation angle distributions and the results were compared to a theoretical distribution from the literature [131, 132, 147], obtained assuming that primary particles have a random orientation. According to this distribution, the amount of primary particles with the same orientation should be very small. A larger than expected amount of similarly oriented primary particles was observed both for "grown-in" and "lightly-touching" pairs, as well as for their combination, where the behavior was most prominent for the "grown-in" pairs. However, no influence of the supersaturation profile could be confirmed. While it was shown that the developed method enables measuring these distributions, more experiments using larger samples, better sampling and constant supersaturation should be conducted in order to further examine this phenomenon. The presented method represents an alternative to the TEM diffraction pattern measurements [34, 42]. Its advantage lies in a simpler preparation and measurement procedure. However, unlike the approach based on diffraction patterns, it is not possible to differentiate between primary and secondary aggregation mechanisms using 3D imaging. The results of the conducted experiments, along with the methodology, were published together with the Magdeburg group [3].

Chapters 3 to 5 have shown that detailed information regarding the shape of crystals can be obtained using 3D imaging techniques. However, such techniques cannot be applied on-line and require sampling of the crystals which may disturb the system, while the measurements may take several hours. In certain cases, it is sufficient to characterize the crystalline product in terms of the agglomeration degree, defined as the relative amount of single crystals and aggregates. Thus, 2D imaging techniques can be applied as the detailed description of each primary particle is not necessary. These techniques enable imaging a much larger amount of particles due to a simpler procedure and can be implemented on-line or in-line. In order to evaluate the agglomeration degree, imaged objects must be classified into single crystals, aggregates and, in case of dynamic image analysis in suspension, gas bubbles. In cooperation with the group of Prof. Schembecker, TU Dortmund University, two methods for automatic classification of imaged objects were investigated and presented in a joint publication [4]. While the Dortmund group focused on the experiments and the linear classification using discriminant factorial analysis (DFA), nonlinear classification using artificial neural networks (ANN) was investigated by our group. Both methods use a training set containing objects manually classified by a human expert, where a set of image descriptors was computed for each object. It was shown that ANN outperforms DFA as fewer image descriptors are needed to achieve the target classification accuracy. Further parameters, including the training set size, number of neurons and the selection of image descriptors were discussed.

This work has shown that it is possible to characterize the shape of both single crystals and crystal

aggregates with different amount of detail depending on the choice of the imaging method. Both imaging methods have advantages and disadvantages, so that an appropriate method should be chosen according to the desired information which is to be obtained. The developed strategy for measuring the agglomeration degree, based on 2D images, should further be extended towards varying crystal size and different chemical systems, which is the focus of the Dortmund research group [4]. This can enable a detailed study of the influence of different crystallization parameters on the agglomeration degree and thus provide some of the information necessary in order to gain control over the PSD and purity of the final product.

In contrast to the discussed 2D techniques, 3D imaging methods provide full geometric information about both single crystals and primary particles that form crystal aggregates. In the future, these techniques can be employed to gain a more fundamental understanding of the mechanisms occurring in crystallization. Potential contributions of the techniques can be considered as twofold. First, they can enable the experimental study of parameters that influence growth and aggregation mechanisms, such as supersaturation, mixing and crystal size. As an example, apart from primary particle orientation, the point of primary particle contact should be investigated and correlated with aggregate strength. This could yield more information about the ability of an aggregate to survive the hydrodynamic forces in a reactor and the subsequent downstream processing steps. The second potential benefit from the 3D imaging techniques lies in the validation of simulation studies. Simulations can be employed to decouple effects which otherwise cannot be studied separately or to gain control over parameters that are experimentally inaccessible. On single crystal scale, it has recently become feasible to simulate crystal growth in face directions using multi-dimensional population balance models. Validation of such simulations is difficult due to the challenges in measurement of the 3D crystal polytope and the face-specific growth rates using 2D imaging techniques. Regarding simulation of crystal aggregates, a Monte Carlo approach can be adopted in order to consider full geometric complexity of the particles and thus correctly close the mass balance occurring under supersaturation-dependent growth [154]. In such simulations, a small subset of particles in a considered reactor setup are individually tracked and allowed to grow and form geometrically realistic aggregates where primary particles overlap [154]. Such simulations are currently too computationally expensive to enable a useful study of the influencing parameters and are thus still in their infancy. However, it is reasonable to expect the computational times to improve in the future. The problem regarding the validation of such simulations could be resolved by employing the here proposed 3D imaging methods that provide the necessary geometric information.

It must be noted, however, that the discussed experimental and validation studies regarding both single crystals and aggregates would be based on off-line measurements of crystal samples and would thus suffer from all sampling-related drawbacks regarding the ability to truly represent the underlying crystal population and the disturbance of the system through impurities. Thus, the best information would be obtained by further coupling the obtained results with 2D imaging techniques that enable in-line or on-line observation of the particles.

Bibliography

- [1] T. KOVAČEVIĆ, A. REINHOLD, AND H. BRIESEN. *Identifying faceted crystal shape from three-dimensional tomography data*. Cryst. Growth Des., 14 (2014), pp. 1666–1675.
- [2] T. KOVAČEVIĆ, J. SCHOCK, F. PFEIFFER, AND H. BRIESEN. *Shape identification of primary particles in potash alum aggregates using three-dimensional tomography data*. Cryst. Growth Des., 16 (2016), pp. 2685–2699.
- [3] T. KOVAČEVIĆ, V. WIEDMEYER, J. SCHOCK, A. VOIGT, F. PFEIFFER, K. SUNDMACHER, AND H. BRIESEN. *Disorientation angle distribution of primary particles in potash alum aggregates*. J Cryst. Growth, 467 (2017), pp. 93 – 106.
- [4] S. HEISEL, T. KOVAČEVIĆ, H. BRIESEN, G. SCHEMBECKER, AND K. WOHLGEMUTH. *Variable selection and training set design for particle classification using a linear and a non-linear classifier*. Chem. Eng. Sci., 173 (2017), pp. 131–144.
- [5] N. VARIANKAVAL, A. S. COTE, AND M. F. DOHERTY. *From form to function: Crystallization of active pharmaceutical ingredients*. AIChE J, 54 (2008), pp. 1682–1688.
- [6] R. W. HARTEL. *Advances in food crystallization*. In *Annu. Rev. Food. Sci. Technol.* (edited by M. P. DOYLE, AND T. R. KLAENHAMMER), volume 4, pp. 277–292. Annual Reviews (2013) .
- [7] B. R. PINZER, A. MEDEBACH, H. J. LIMBACH, C. DUBOIS, M. STAMPANONI, AND M. SCHNEEBELI. *3D-characterization of three-phase systems using X-ray tomography: tracking the microstructural evolution in ice cream*. Soft Matter, 8 (2012), pp. 4584–4594.
- [8] H. D. GOFF AND R. W. HARTEL. *Ice Cream*. Springer, seventh edition (2013).
- [9] R. T. MARSHALL AND W. S. ARBUCKLE. *Ice Cream*. Chapman & Hall, fifth edition (1996).
- [10] Y. KIM, S. HAAM, W.-S. KIM, AND K.-K. KOO. *Parameters determining the agglomeration behavior of anhydrous L-ornithine-L-aspartate (LOLA) crystals prepared by drowning out crystallization*. Korean J. Chem. Eng., 20 (2003), pp. 1111–1117.
- [11] C. WIBOWO, W.-C. CHANG, AND K. M. NG. *Design of integrated crystallization systems*. AIChE J, 47 (2001), pp. 2474–2492.

- [12] J. F. GAMBLE, M. TOBYN, AND R. HAMEY. *Application of image-based particle size and shape characterization systems in the development of small molecule pharmaceuticals*. J. Pharm Sci, 104 (2015), pp. 1563–1574.
- [13] N. RASENACK AND B. W. MÜLLER. *Crystal habit and tableting behavior*. Int. J. Pharm., 244 (2002), pp. 45–57.
- [14] K. FUNAKOSHI, H. TAKIYAMA, AND M. MATSUOKA. *Influences of seed crystals on agglomeration phenomena and product purity of m-chloronitrobenzene crystals in batch crystallization*. Chem. Eng. J., 81 (2001), pp. 307 – 312.
- [15] J. BAUER, S. SPANTON, R. HENRY, J. QUICK, W. DZIKI, W. PORTER, AND J. MORRIS. *Ritonavir: An extraordinary example of conformational polymorphism*. Pharm Res, 18 (2001), pp. 859–866.
- [16] L. L. SIMON, ET AL. *Assessment of recent process analytical technology (PAT) trends: a multi-author review*. Org. Process Res. Dev., 19 (2015), pp. 3–62.
- [17] F. PUEL, P. MARCHAL, AND J. KLEIN. *Habit transient analysis in industrial crystallization using two dimensional crystal sizing technique*. Chem. Eng. Res. Des., 75 (1997), pp. 193–205.
- [18] M. OULLION, F. PUEL, G. FÉVOTTE, S. RIGHINI, AND P. CARVIN. *Industrial batch crystallization of a plate-like organic product. In situ monitoring and 2D-CSD modelling: Part 1: Experimental study*. Chem. Eng. Sci., 62 (2007), pp. 820–832.
- [19] M. R. SINGH, J. CHAKRABORTY, N. NERE, H.-H. TUNG, S. BORDAWEKAR, AND D. RAMKRISHNA. *Image-analysis-based method for 3d crystal morphology measurement and polymorph identification using confocal microscopy*. Cryst. Growth Des., 12 (2012), pp. 3735–3748.
- [20] J. CALDERON DE ANDA, X. Z. WANG, AND K. J. ROBERTS. *Multi-scale segmentation image analysis for the in-process monitoring of particle shape with batch crystallisers*. Chem. Eng. Sci., 60 (2005), pp. 1053–1065.
- [21] P. A. LARSEN, J. B. RAWLINGS, AND N. J. FERRIER. *An algorithm for analyzing noisy, in situ images of high-aspect-ratio crystals to monitor particle size distribution*. Chem. Eng. Sci., 61 (2006), pp. 5236–5248.
- [22] P. A. LARSEN, J. B. RAWLINGS, AND N. J. FERRIER. *Model-based object recognition to measure crystal size and shape distributions from in situ video images*. Chem. Eng. Sci., 62 (2007), pp. 1430–1441.
- [23] R. ZHANG, C. Y. MA, J. J. LIU, AND X. Z. WANG. *On-line measurement of the real size and shape of crystals in stirred tank crystalliser using non-invasive stereo vision imaging*. Chem. Eng. Sci., 137 (2015), pp. 9 – 21.

- [24] D. B. PATIENCE AND J. B. RAWLINGS. *Particle-shape monitoring and control in crystallization processes*. AIChE J, 47 (2001), pp. 2125–2130.
- [25] C. BORCHERT, E. TEMMEL, H. EISENSCHMIDT, H. LORENZ, A. SEIDEL-MORGENSTERN, AND K. SUNDMACHER. *Image-based in situ identification of face specific crystal growth rates from crystal populations*. Cryst. Growth Des., 14 (2014), pp. 952–971.
- [26] S. SCHORSCH, T. VETTER, AND M. MAZZOTTI. *Measuring multidimensional particle size distributions during crystallization*. Chem. Eng. Sci., 77 (2012), pp. 130–142.
- [27] S. SCHORSCH, D. R. OCHSENBEIN, T. VETTER, M. MORARI, AND M. MAZZOTTI. *High accuracy online measurement of multidimensional particle size distributions during crystallization*. Chem. Eng. Sci, 105 (2014), pp. 155 – 168.
- [28] P. DANDEKAR, Z. B. KUVADIA, AND M. F. DOHERTY. *Engineering crystal morphology*. Annu. Rev. Mater. Res., 43 (2013), pp. 359–386.
- [29] J. W. MULLIN. *Crystallization*. Butterworth-Heinemann, third edition (1993).
- [30] K. WOHLGEMUTH, A. KORDYLLA, F. RUETHER, AND G. SCHEMBECKER. *Experimental study of the effect of bubbles on nucleation during batch cooling crystallization*. Chem. Eng. Sci., 64 (2009), pp. 4155 – 4163.
- [31] A. G. JONES. *Crystallization Process Systems*. Butterworth-Heinemann (2002).
- [32] A. REINHOLD, S. SCHORSCH, M. MAZZOTTI, AND H. BRIESEN. *Modeling and measurement of abraded particles*. Powder Technol., 271 (2015), pp. 134 – 140.
- [33] H. S. MUMTAZ, M. J. HOUNSLOW, N. A. SEATON, AND W. R. PATERSON. *Orthokinetic aggregation during precipitation: A computational model for calcium oxalate monohydrate*. Chem. Eng. Res. Des., 75 (1997), pp. 152 – 159.
- [34] A. P. COLLIER, C. J. D. HETHERINGTON, AND M. J. HOUNSLOW. *Alignment mechanisms between particles in crystalline aggregates*. J. Cryst. Growth, 208 (2000), pp. 513 – 519.
- [35] M. J. HOUNSLOW, H. S. MUMTAZ, A. P. COLLIER, J. P. BARRICK, AND A. S. BRAMLEY. *A micro-mechanical model for the rate of aggregation during precipitation from solution*. Chem. Eng. Sci., 56 (2001), pp. 2543 – 2552.
- [36] D. R. OCHSENBEIN, T. VETTER, M. MORARI, AND M. MAZZOTTI. *Agglomeration of needle-like crystals in suspension. II. Modeling*. Cryst. Growth Des., 15 (2015), pp. 4296–4310.
- [37] F. PRATOLA, S. J. R. SIMONS, AND A. G. JONES. *A novel experimental device for measurement of agglomerative crystallization forces*. Chem. Eng. Res. Des., 80 (2002), pp. 441–448.

- [38] E. M. ÅLANDER, M. S. UUSI-PENTTILÄ, AND Å. C. RASMUSON. *Characterization of paracetamol agglomerates by image analysis and strength measurement*. Powder Technol., 130 (2003), pp. 298 – 306.
- [39] E. M. ÅLANDER, M. S. UUSI-PENTTILÄ, AND Å. C. RASMUSON. *Agglomeration of paracetamol during crystallization in pure and mixed solvents*. Ind. Eng. Chem. Res., 43 (2004), pp. 629–637.
- [40] E. M. ÅLANDER AND Å. C. RASMUSON. *Mechanisms of crystal agglomeration of paracetamol in acetone-water mixtures*. Ind. Eng. Chem. Res., 44 (2005), pp. 5788–5794.
- [41] Y. AOKI AND Y. NAKAMUTA. *Penetration twins of potassium chloride*. J. Cryst. Growth, 67 (1984), pp. 579 – 586.
- [42] A. P. COLLIER, C. J. D. HETHERINGTON, AND M. J. HOUNSLOW. *High resolution imaging of crystalline agglomerates*. Chem. Eng. Res. Des., 74 (1996), pp. 759–764.
- [43] H. EISENSCHMIDT, N. BAJCINCA, AND K. SUNDMACHER. *Optimal control of crystal shapes in batch crystallization experiments by growth-dissolution cycles*. Cryst. Growth Des., 16 (2016), pp. 3297–3306.
- [44] C. Y. MA, J. J. LIU, AND X. Z. WANG. *Measurement, modelling, and closed-loop control of crystal shape distribution: Literature review and future perspectives*. Particuology, 26 (2016), pp. 1–18.
- [45] D. R. OCHSENBEIN, S. SCHORSCH, T. VETTER, M. MAZZOTTI, AND M. MORARI. *Growth rate estimation of beta l-glutamic acid from online measurements of multidimensional particle size distributions and concentration*. Ind. Eng. Chem. Res., 53 (2014), pp. 9136–9148.
- [46] C. Y. MA, X. Z. WANG, AND K. J. ROBERTS. *Morphological population balance for modeling crystal growth in face directions*. AIChE J, 54 (2008), pp. 209–222.
- [47] A. REINHOLD AND H. BRIESEN. *High dimensional population balances for the growth of faceted crystals: Combining Monte Carlo integral estimates and the method of characteristics*. Chem. Eng. Sci., 127 (2015), pp. 220–229.
- [48] R. DAVID, P. MARCHAL, J.-P. KLEIN, AND J. VILLERMAUX. *Crystallization and precipitation engineering 3. a discrete formulation of the agglomeration rate of crystals in a crystallization process*. Chem. Eng. Sci., 46 (1991), pp. 205–213.
- [49] H. BRIESEN. *Aggregate structure evolution for size-dependent aggregation by means of Monte Carlo simulations*. KONA Powder Part. J., 25 (2007), pp. 180–189.
- [50] M. J. HOUNSLOW, E. J. W. WYNN, M. KUBO, AND K. PITT. *Aggregation of growing crystals in suspension: I. Mumtaz revisited*. Chem. Eng. Sci., 101 (2013), pp. 731–743.

- [51] H. BRIESEN. *Hierarchical characterization of aggregates for Monte Carlo simulations*. AIChE J, 52 (2006), pp. 2436–2446.
- [52] A. REINHOLD. *Crystal shape modeling and convex geometry - analysis of geometric state spaces*. Ph.D. thesis, Technische Universität München (2015).
- [53] A. REINHOLD AND H. BRIESEN. *Convex geometry for the morphological modeling and characterization of crystal shapes*. Part. Part. Syst. Charact., 28 (2011), pp. 37–56.
- [54] K. FUKUDA. *cddlib* (2012). v.094g, https://www.inf.ethz.ch/personal/fukudak/cdd_home/, last accessed: 23.08.2017.
- [55] M. R. SINGH, P. VERMA, H.-H. TUNG, S. BORDAWEKAR, AND D. RAMKRISHNA. *Screening crystal morphologies from crystal structure*. Cryst. Growth Des., 13 (2013), pp. 1390–1396.
- [56] M. STIESS. *Mechanische Verfahrenstechnik - Partikeltechnologie 1*. Springer-Verlag Berlin Heidelberg, third edition (2009).
- [57] O. S. AGIMELEN, P. HAMILTON, I. HALEY, A. NORDON, M. VASILE, J. SEFCIK, AND A. J. MULLHOLLAND. *Estimation of particle size distribution and aspect ratio of non-spherical particles from chord length distribution*. Chem. Eng. Sci., 123 (2015), pp. 629–640.
- [58] I. DE ALBUQUERQUE, M. MAZZOTTI, D. R. OCHSENBEIN, AND M. MORARI. *Effect of needle-like crystal shape on measured particle size distributions*. AIChE J, 62 (2016), pp. 2974–2985.
- [59] H. LI, M. A. GROVER, Y. KAWAJIRI, AND R. W. ROUSSEAU. *Development of an empirical method relating crystal size distributions and FBRM measurements*. Chem. Eng. Sci., 89 (2013), pp. 142 – 151.
- [60] H. LI, Y. KAWAJIRI, M. A. GROVER, AND R. W. ROUSSEAU. *Application of an empirical FBRM model to estimate crystal size distributions in batch crystallization*. Cryst. Growth Des., 14 (2014), pp. 607–616.
- [61] M. KEMPKES, J. EGGERS, AND M. MAZZOTTI. *Measurement of particle size and shape by FBRM and in situ microscopy*. Chem. Eng. Sci., 63 (2008), pp. 4656 – 4675.
- [62] N. KAIL, H. BRIESEN, AND W. MARQUARDT. *Advanced geometrical modeling of Focused Beam Reflectance Measurements (FBRM)*. Part. Part. Syst. Charact., 24 (2007), pp. 184–192.
- [63] A. JILLAVENKATESA, S. J. DAPKUNAS, AND L.-S. H. LUM. *NIST Recommended Practice Guide: Particle Size Characterization, NIST Special Publication 960-1*. National Institute of Standards and Technology, U.S. Government Printing Office (2001).
- [64] P. MOUGIN, D. WILKINSON, K. J. ROBERTS, AND R. TWEEDIE. *Characterization of particle size and its distribution during the crystallization of organic fine chemical products as measured in situ using ultrasonic attenuation spectroscopy*. J. Acoust. Soc. Am., 109 (2001), pp. 274–282.

- [65] D. Y. KIM AND D. R. YANG. *A novel method for measurement of crystal growth rate*. J. Cryst. Growth, 373 (2013), pp. 54–58.
- [66] S. K. BRAR AND M. VERMA. *Measurement of nanoparticles by light-scattering techniques*. Trends Anal. Chem., 30 (2011), pp. 4 – 17.
- [67] E. AAMIR, Z. K. NAGY, AND C. D. RIELLY. *Evaluation of the effect of seed preparation method on the product crystal size distribution for batch cooling crystallization processes*. Cryst. Growth Des., 10 (2010), pp. 4728–4740.
- [68] L.-M. TERDENG, S. HEISEL, G. SCHEMBECKER, AND K. WOHLGEMUTH. *Agglomeration degree distribution as quality criterion to evaluate crystalline products*. Chem. Eng. Sci., 133 (2015), pp. 157 – 169.
- [69] D. R. OCHSENBEIN, T. VETTER, S. SCHORSCH, M. MORARI, AND M. MAZZOTTI. *Agglomeration of needle-like crystals in suspension. I. Measurements*. Cryst. Growth Des., 15 (2015), pp. 1923–1933.
- [70] Y. HUO, T. LIU, H. LIU, C. Y. MA, AND X. Z. WANG. *In-situ crystal morphology identification using imaging analysis with application to the L-glutamic acid crystallization*. Chem. Eng. Sci., 148 (2016), pp. 126–139.
- [71] C. Y. MA, J. WAN, AND X. Z. WANG. *Faceted growth rate estimation of potash alum crystals grown from solution in a hot-stage reactor*. Powder Technol., 227 (2012), pp. 96–103.
- [72] J. EGGERS, M. KEMPKES, AND M. MAZZOTTI. *Measurement of size and shape distributions of particles through image analysis*. Chem. Eng. Sci., 63 (2008), pp. 5513–5521.
- [73] D. J. BROWN, G. T. VICKERS, A. P. COLLIER, AND G. K. REYNOLDS. *Measurement of the size, shape and orientation of convex bodies*. Chem. Eng. Sci., 60 (2005), pp. 289–292.
- [74] L.-M. TERDENG AND K. WOHLGEMUTH. *Impact of agglomeration on crystalline product quality within the crystallization process chain*. Cryst. Res. Technol., (2016), pp. 1–11.
- [75] X. Z. WANG, K. J. ROBERTS, AND C. MA. *Crystal growth measurement using 2D and 3D imaging and the perspectives for shape control*. Chem. Eng. Sci., 63 (2008), pp. 1173–1184.
- [76] C. Y. MA, J. J. LIU, AND X. Z. WANG. *Stereo imaging of crystal growth*. AIChE J, 62 (2016), pp. 18–25.
- [77] K. WU, C. Y. MA, J. J. LIU, Y. ZHANG, AND X. Z. WANG. *Measurement of crystal face specific growth kinetics*. Cryst. Growth Des., 16 (2016), pp. 4855–4868.
- [78] C. INDHUMATHI, Y. Y. CAI, Y. Q. GUAN, AND M. OPAS. *An automatic segmentation algorithm for 3D cell cluster splitting using volumetric confocal images*. J. Microsc., 243 (2011), pp. 60–76.

- [79] M. BARIGOU AND M. DOUAIRE. *9 - X-ray micro-computed tomography for resolving food microstructures*. In *Food Microstructures* (edited by V. J. MORRIS AND K. GROVES), Woodhead Publishing Series in Food Science, Technology and Nutrition, pp. 246 – 272. Woodhead Publishing (2013) .
- [80] S. STOCK. *MicroComputed Tomography*. CRC Press, Taylor & Francis Group (2009).
- [81] R. C. GONZALEZ, R. E. WOODS, AND S. L. EDDINS. *Digital Image Processing Using Matlab*. McGraw Hill Education (India), Gatesmark, LLC (2010).
- [82] S. T. ERDOGAN, X. NIE, P. E. STUTZMAN, AND E. J. GARBOCZI. *Micrometer-scale 3-D shape characterization of eight cements: Particle shape and cement chemistry, and the effect of particle shape on laser diffraction particle size measurement*. *Cem. Concr. Res.*, 40 (2010), pp. 731–739.
- [83] D. A. JERRAM, A. MOCK, G. R. DAVIS, M. FIELD, AND R. J. BROWN. *3D crystal size distributions: A case study on quantifying olivine populations in kimberlites*. *LITHOS*, 112 (2009), pp. 223–235.
- [84] C. REDENBACH, R. OHSER-WIEDEMANN, R. LÖFFLER, T. BERNTHALER, AND A. NAGEL. *Characterization of powders using micro computed tomography*. *Part. Part. Syst. Charact.*, 28 (2011), pp. 3–12.
- [85] P. FRISULLO, M. BARNABÀ, L. NAVARINI, AND M. A. DEL NOBILE. *Coffea arabica beans microstructural changes induced by roasting: An X-ray microtomographic investigation*. *J. Food Eng.*, 108 (2012), pp. 232 – 237.
- [86] W. BURGER AND M. BURGE. *Principles of Digital Image Processing - Advanced Methods*. Springer, Undergraduate Topics in Computer Science, London (2013).
- [87] N. OTSU. *A threshold selection method from gray-level histograms*. *IEEE Transactions on Systems, Man, and Cybernetics*, 9 (1979), pp. 62–66.
- [88] J. OHSER AND K. SCHLADITZ. *3D Images of Material Structures: Processing and Analysis*. WILEY-VCH Verlag (2009).
- [89] L. VINCENT AND P. SOILLE. *Watersheds in digital spaces - an efficient algorithm based on immersion simulations*. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13 (1991), pp. 583–598.
- [90] A. SPETTL, S. BACHSTEIN, M. DOSTA, M. GOSLINSKA, S. HEINRICH, AND V. SCHMIDT. *Bonded-particle extraction and stochastic modeling of internal agglomerate structures*. *Adv. Powder Technol.*, 27 (2016), pp. 1761 – 1774.
- [91] P. S. UMESH ADIGA AND B. B. CHAUDHURI. *An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images*. *Pattern Recognit.*, 34 (2001), pp. 1449–1458.

- [92] C. WÄHLBY, I.-M. SINTORN, F. ERLANDSSON, G. BORGEFORS, AND E. BENGTTSSON. *Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections*. J. Microsc., 215 (2004), pp. 67–76.
- [93] F. LONG, H. PENG, AND E. MYERS. *Automatic segmentation of nuclei in 3D microscopy images of C.elegans*. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Vols 1-3, Arlington, VA, 2007*. IEEE (2007) pp. 536–539.
- [94] C. ZHANG, C. SUN, AND T. D. PHAM. *Segmentation of clustered nuclei based on concave curve expansion*. J. Microsc., 251 (2013), pp. 57–67.
- [95] H. IRSHAD, A. VEILLARD, L. ROUX, AND D. RACOCEANU. *Methods for nuclei detection, segmentation, and classification in digital histopathology: A review - current status and future potential*. IEEE Rev. Biomed. Eng., 7 (2014), pp. 97–114.
- [96] FRAUNHOFER ITWM. *MAVI - Modular Algorithms for Volume Images, v. 1.4.1*. V. 1.4.1.
- [97] O. S. AHMAD, J. DEBAYLE, AND J.-C. PINOLI. *A geometric-based method for recognizing overlapping polygonal-shaped and semi-transparent particles in gray tone images*. Pattern Recogn. Lett., 32 (2011), pp. 2068–2079.
- [98] W. BURGER AND M. BURGE. *Principles of Digital Image Processing - Core Algorithms*. Springer, Undergraduate Topics in Computer Science, London (2009).
- [99] J. EGGERS, M. KEMPKES, J. CORNEL, M. MAZZOTTI, I. KOSCHINSKI, AND E. VERDURAND. *Monitoring size and shape during cooling crystallization of ascorbic acid*. Chem. Eng. Sci., 64 (2009), pp. 163 – 171.
- [100] M. KEMPKES, T. VETTER, AND M. MAZZOTTI. *Measurement of 3D particle size distributions by stereoscopic imaging*. Chem. Eng. Sci., 65 (2010), pp. 1362–1373.
- [101] M. ĐURIŠ, Z. ARSENIJEVIĆ, D. JAĆIMOVSKI, AND T. KALUĐEROVIĆ RADOIČIĆ. *Optimal pixel resolution for sand particles size and shape analysis*. Powder Technol., 302 (2016), pp. 177 – 186.
- [102] R. LI, G. THOMSON, G. WHITE, X. WANG, J. CALDERON DE ANDA, AND K. ROBERTS. *Integration of crystal morphology modeling and on-line shape measurement*. AIChE J, 52 (2006), pp. 2297–2305.
- [103] C. BORCHERT AND K. SUNDMACHER. *Crystal aggregation in a flow tube: Image-based observation*. Chem. Eng. Technol., 34 (2011), pp. 545–556.
- [104] M.-N. PONS, H. VIVIER, AND J. DODDS. *Particle shape characterization using morphological descriptors*. Part. Part. Syst. Charact., 14 (1997), pp. 272–277.

- [105] N. FARIA, M. N. PONS, S. FEYO DE AZEVEDO, F. A. ROCHA, AND H. VIVIER. *Quantification of the morphology of sucrose crystals by image analysis*. Powder Technol., 133 (2003), pp. 54–67.
- [106] A. FERREIRA, N. FARIA, F. ROCHA, AND J. A. TEIXEIRA. *Using an online image analysis technique to characterize sucrose crystal morphology during a crystallization run*. Ind. Eng. Chem. Res., 50 (2011), pp. 6990–7002.
- [107] P. HOUGH. *Methods and means for recognizing complex patterns* (US Patent 3069654, 1962).
- [108] O. S. AHMAD, J. DEBAYLE, N. GHERRAS, B. PRESLES, G. FÉVOTTE, AND J.-C. PINOLI. *Quantification of overlapping polygonal-shaped particles based on a new segmentation method of in situ images during crystallization*. J. Electron. Imag., 21 (2012), pp. 021115–1–021115–11.
- [109] S. SCHORSCH, J.-H. HOURS, T. VETTER, M. MAZZOTTI, AND C. N. JONES. *An optimization-based approach to extract faceted crystal shapes from stereoscopic images*. Comput. Chem. Eng., 75 (2015), pp. 171–183.
- [110] R. HARTLEY AND A. ZISSERMAN. *Multiple View Geometry in computer vision*. Cambridge University Press, second edition (2003).
- [111] B. BUJAK AND M. BOTTLINGER. *Three-dimensional measurement of particle shape*. Part. Part. Syst. Charact., 25 (2008), pp. 293–297.
- [112] J. CHAKRABORTY, D. SARKAR, A. SINGH, AND A. K. BHARTI. *Measuring the three-dimensional morphology of crystals using regular reflection of light*. Cryst. Growth Des., 12 (2012), pp. 6042–6049.
- [113] O. S. AGIMELLEN, ET AL. *Integration of in situ imaging and chord length distribution measurements for estimation of particle size and shape*. Chem. Eng. Sci., 144 (2016), pp. 87–100.
- [114] P. BARRETT AND B. GLENNON. *Characterizing the metastable zone width and solubility curve using Lasentec FBRM and PVM*. Chem. Eng. Res. Des., 80 (2002), pp. 799–805.
- [115] G. M. MAGGIONI AND M. MAZZOTTI. *Modelling the stochastic behaviour of primary nucleation*. Faraday Discuss., 179 (2015), pp. 359–382.
- [116] H. EISENSCHMIDT, A. VOIGT, AND K. SUNDMACHER. *Face-specific growth and dissolution kinetics of potassium dihydrogen phosphate crystals from batch crystallization experiments*. Cryst. Growth Des., 15 (2015), pp. 219–227.
- [117] B. BERNARD-MICHEL, S. ROHANI, M.-N. PONS, H. VIVIER, AND H. S. HUNDAL. *Classification of crystal shape using Fourier descriptors and mathematical morphology*. Part. Part. Syst. Charact., 14 (1997), pp. 193–200.

- [118] Y. ZHANG, J. J. LIU, L. ZHANG, J. CALDERON DE ANDA, AND X. Z. WANG. *Particle shape characterisation and classification using automated microscopy and shape descriptors in batch manufacture of particulate solids*. *Particuology*, 24 (2016), pp. 61 – 68.
- [119] C. M. BISHOP. *Pattern Recognition and Machine Learning*. Springer (2006).
- [120] W. J. KRAZANOWSKI. *Principles of Multivariate Analysis*. Oxford University Press, revised edition (1988).
- [121] D. E. RUMELHART, G. E. HINTON, AND R. J. WILLIAMS. *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1*. chapter 8: Learning Internal Representations by Error Propagation, pp. 318–362. MIT Press, Cambridge, MA, USA (1986) .
- [122] S. RAUDYS AND A. K. JAIN. *Small sample size problems in designing artificial neural networks*. In *Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections* (edited by I. K. SETHI AND A. K. JAIN), volume 11 of *Machine Intelligence and Pattern Recognition*, pp. 33 – 50. North-Holland (1991) .
- [123] F. ROS, S. GUILLAUME, G. RABATEL, AND F. SEVILA. *Recognition of overlapping particles in granular product images using statistics and neural networks*. *Food Control*, 6 (1995), pp. 37 – 43.
- [124] G. M. FOODY, M. B. MCCULLOH, AND W. B. YATES. *The effect of training set size and composition on artificial neural-netowrk classification*. *Int. J. Remote Sensing*, 16 (1995), pp. 1707–1723.
- [125] C. SIMA AND E. R. DOUGHERTY. *The peaking phenomenon in the presence of feature-selection*. *Pattern Recogn. Lett.*, 29 (2008), pp. 1667 – 1674.
- [126] Y. HAMAMOTO, S. UCHIMURA, AND S. TOMITA. *On the behavior of artificial neural network classifiers in high-dimensional spaces*. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18 (1996), pp. 571–574.
- [127] Z. Q. YU, P. S. CHOW, AND R. B. TAN. *Quantification of particle morphology by boundary Fourier transform and generic Fourier transform*. *Chem. Eng. Sci.*, 62 (2007), pp. 3777 – 3786.
- [128] H. S. HUNDAL, S. ROHANI, H. C. WOOD, AND M. N. PONS. *Particle shape characterization using image analysis and neural networks*. *Powder Technol.*, 91 (1997), pp. 217 – 227.
- [129] H.-J. BUNGE. *Texture Analysis in Material Science: Mathematical Methods*. Butterworths (1982).
- [130] V. K. IVANOV, P. P. FEDOROV, A. Y. BARANCHIKOV, AND V. V. OSIKO. *Oriented attachment of particles: 100 years of investigations of non-classical crystal growth*. *Russ. Chem. Rev.*, 83 (2014), pp. 1204–1222.

- [131] J. K. MACKENZIE. *Second paper on statistics associated with the random disorientation of cubes*. *Biometrika*, 45 (1958), pp. 229–240.
- [132] D. C. HANDSCOMB. *On the random disorientation of two cubes*. *Canad. J. Math.* 10, (1958), pp. 85–88.
- [133] D. BORRMANN, J. ELSEBERG, K. LINGEMANN, AND A. NÜCHTER. *The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design*. *3D Res.*, 2 (2011), pp. 1–13.
- [134] K. M. GÓRSKI, E. HIVON, A. J. BANDAY, B. D. WANDEL, F. K. HANSEN, M. REINECKE, AND M. BARTELMANN. *HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere*. *Astrophys. J.*, 622 (2005), pp. 759–771.
- [135] Y. NARUSE. *Healpix library for matlab* (2011). Version 1.0, <http://sufoo.c.ooco.jp/program/healpix.html>, last accessed: 23.08.2017.
- [136] G. WAHBA. *A least squares estimate of satellite attitude*. *SIAM Rev.*, 7 (1965), pp. 409–409.
- [137] F. L. MARKLEY. *Attitude determination using vector observations and the singular value decomposition*. *J. Astronaut. Sci.*, 36 (1988), pp. 245–258.
- [138] X. PEREZ-SALA, L. IGUAL, S. ESCALERA, AND C. ANGULO. *Uniform Sampling of Rotations for Discrete and Continuous Learning of 2D Shape Models*. In *Robotic Vision: Technologies for Machine Learning and Vision Applications* (edited by J. GARCIA-RODRIGUEZ, AND M. A. C. QUEVEDO), pp. 23–42 (2013) .
- [139] VOLUME GRAPHICS GMBH, HEIDELBERG, GERMANY. *MyVGL*. Version 2.1.
- [140] T. KOVAČEVIĆ, A. REINHOLD, AND H. BRIESEN. *Shape identification of primary particles in crystal aggregates*. In *ISIC 19th International Symposium on Industrial Crystallization* (2014) pp. 112–115. Toulouse, France.
- [141] G. FERNÁNDEZ, M. KUNT, AND J.-P. ZRÝD. *A new plant cell image segmentation algorithm*. In *Image Analysis and Processing. 8th International Conference, ICIAP '95. Proceedings* (1995) pp. 229–34.
- [142] M. FARHAN, O. YLI-HARJA, AND A. NIEMISTÖ. *A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search*. *Pattern Recognit.*, 46 (2013), pp. 741–751.
- [143] S. KUMAR, S. H. ONG, S. RANGANATH, T. C. ONG, AND F. T. CHEW. *A rule-based approach for robust clump splitting*. *Pattern Recognit.*, 39 (2006), pp. 1088–1098.

- [144] S. WIENERT, ET AL. *Detection and segmentation of cell nuclei in virtual microscopy images: A minimum-model approach*. *Sci. Rep.*, 2 (2012), pp. 1–7.
- [145] J. QI. *Dense nuclei segmentation based on graph cut and convexity-concavity analysis*. *J. Microsc.*, 253 (2014), pp. 42–53.
- [146] FEI VSG. *AVIZO Fire, v. 8.1.0*.
- [147] J. K. MACKENZIE AND M. J. THOMSON. *Some statistics associated with the random disorientation of cubes*. *Biometrika*, 44 (1957), pp. 205–210.
- [148] T. KOVAČEVIĆ, V. WIEDMEYER, J. SCHOCK, F. PFEIFFER, A. VOIGT, K. SUNDMACHER, AND H. BRIESEN. *Orientation of primary particles in potash alum aggregates*. In *BIWIC 2016, 23rd International Workshop on Industrial Crystallization* (edited by H. LORENZ AND H. BUCHHOLZ) (2016) pp. 167–172. Magdeburg, Germany.
- [149] J. W. MULLIN, J. GARSIDE, AND R. UNAHABHOKHA. *Diffusivities of ammonium and potassium alums in aqueous solutions*. *J. appl. Chem.*, 15 (1965), pp. 502–505.
- [150] THE MATHWORKS, NATICK, MASSACHUSETTS, USA. *MATLAB*. Version: R2015b.
- [151] A. ROSENFELD. *Measuring the sizes of concavities*. *Pattern Recogn. Lett.*, 3 (1985), pp. 71 – 75.
- [152] J. VEGELIUS, S. JANSON, AND F. JOHANSSON. *Measures of similarity between distributions*. *Qual. Quant.*, 20 (1986), pp. 437–441.
- [153] THE MATHWORKS, NATICK, MASSACHUSETTS, USA. *MATLAB*. Version: R2015a, documentation.
- [154] T. KOVAČEVIĆ AND H. BRIESEN. *Determining the full geometric complexity of crystal aggregates via monte carlo simulation*. In *PARTEC: International Congress of Particle Technology* (2016) Nürnberg, Germany.

Appendix A

Projecting a Point onto a Polytope

In order to measure the quality of the shapes identified by image processing, points on the crystal surface are projected to the identified polytope, as explained in [1]¹. Before performing the projections, an H -representation of the polytope containing only the non-redundant face directions must be obtained. This polytope is considered to be centered at the origin of the coordinate system. Thus, the points \mathbf{p}_i must be moved so that they are in the coordinate system whose origin coincides with the polytope center. The resulting points are denoted as $\hat{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{x}_{\text{orig}}$.

A point $\hat{\mathbf{p}}_i$ inside the polytope is projected onto the face with the normal \mathbf{a}_j towards which it has the smallest distance d :

$$d = \min_{j \in [1, N_H]} (h_j - \langle \mathbf{a}_j, \hat{\mathbf{p}}_i \rangle), \quad (\text{A.1})$$

where h_j are face distances of the N_H polytope faces. The projection is then obtained as

$$\hat{\mathbf{p}}_{i,\text{proj}} = \hat{\mathbf{p}}_i - \langle (\hat{\mathbf{p}}_i - h_j \mathbf{a}_j), \mathbf{a}_j \rangle \cdot \mathbf{a}_j \quad (\text{A.2})$$

The point $\hat{\mathbf{p}}_i$ outside the polytope can be projected either on a face, an edge or a vertex. Each face, edge and vertex defines an unbounded influence region in space containing points that should be projected onto it. This is illustrated in Figure A.1, where the influence regions are presented in different colors. Each influence region is defined by a set of normals \mathbf{b}_k to region faces, presented as arrows in Figure A.1, as well as a set of corresponding points $\mathbf{b}_{p,k}$ on region faces. This can be used to check whether a point $\hat{\mathbf{p}}_i$ is in the region j . Region faces are parts of planes that divide the space in two and a point is inside the region if it is on the correct side of each face. Thus, it is in the region if the following is fulfilled for each pair $(\mathbf{b}_k, \mathbf{b}_{p,k})$:

$$\langle \mathbf{b}_k, (\hat{\mathbf{p}}_i - \mathbf{b}_{p,k}) \rangle > 0. \quad (\text{A.3})$$

A point is projected onto a face j if it is in the influence region of the face j , as shown in the left of Figure A.1. This influence region is bounded by the face j as well as by planes that contain edges of

¹Note that there are corrections with respect to the corresponding publication [1] regarding the projection onto an edge.

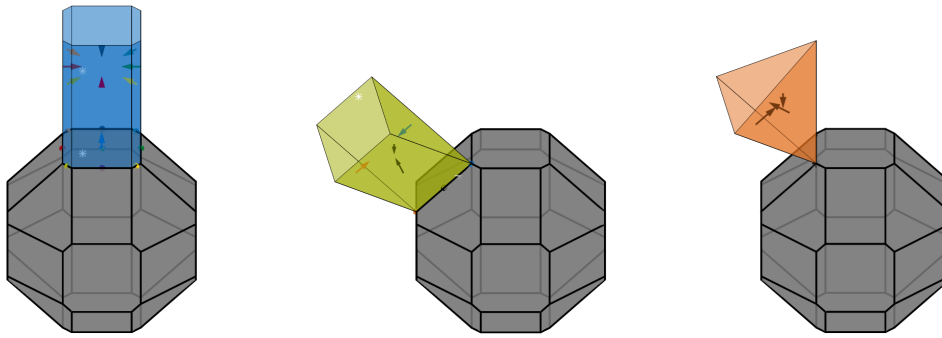


Figure A.1: A point can be projected onto a face (left), edge (middle) or vertex (right), depending on its location. The influence region of a face, edge and vertex is illustrated in blue, green and orange, respectively. Influence regions are defined by their face normals, illustrated as arrows with different colors, as well as corresponding points on each region face, illustrated with the same colors as the normals. An example of a point being projected onto a face and edge is presented using white stars. Any point in the vertex influence region is projected to that vertex, so that no examples are illustrated for that case.

the face j and are perpendicular to j . The corresponding normals \mathbf{b} therefore contain the face normal \mathbf{a}_j , while the corresponding point \mathbf{b}_p is the arithmetic mean of the vertices of the face j . Normals \mathbf{b} are furthermore obtained using the cross product of \mathbf{a}_j and the edges of the face j . The corresponding points \mathbf{b}_p are the middle points of edges of the face j . In the left part of Figure A.1, normals \mathbf{b} and the corresponding points \mathbf{b}_p are illustrated with the same color. The projection is computed using equation A.2. An example regarding this type of projection is illustrated by white stars in the left of Figure A.1.

A point is projected onto the edge \mathbf{e}_j if it is in the influence region of this edge, as shown in the middle part of Figure A.1. This edge is an intersection of faces k and l and connects vertices \mathbf{v}_m and \mathbf{v}_n . The influence region contains two types of boundaries. Two normals \mathbf{b} are normals to planes which are perpendicular to the edge \mathbf{e}_j and contain its vertices. Thus, these normals \mathbf{b} are defined by the edge direction. The corresponding points \mathbf{b}_p are the vertices of the edge. These $(\mathbf{b}, \mathbf{b}_p)$ pairs are illustrated in blue and orange in the middle part of Figure A.1. The other boundary type consists of planes that are perpendicular to crystal faces k and l and contain the edge \mathbf{e}_j . Normals \mathbf{b} to these planes are obtained using a cross product of the edge and the corresponding crystal face normal. The corresponding points \mathbf{b}_p represent the middle point of the edge and are the same for both faces. These $(\mathbf{b}, \mathbf{b}_p)$ pairs are illustrated in black in the middle part of Figure A.1. Finally, the projection for a point $\hat{\mathbf{p}}_i$ in the influence region of edge \mathbf{e}_j is obtained by

$$\hat{\mathbf{p}}_{i,\text{proj}} = \mathbf{v}_m + \frac{\langle (\hat{\mathbf{p}}_i - \mathbf{v}_m), \mathbf{e}_j \rangle}{\|\mathbf{e}_j\|^2} \cdot \mathbf{e}_j. \quad (\text{A.4})$$

An example regarding projection to an edge is illustrated by white stars in the middle part of Figure A.1.

A point is projected onto the vertex \mathbf{v}_j if it is in the influence region of the vertex, as illustrated in the

right of Figure A.1. The influence region consists of planes that contain the vertex \mathbf{v}_j and are normal to the edges that intersect in the vertex \mathbf{v}_j . As already explained in case of projection onto an edge, the normals \mathbf{b} in such a case are defined by the edge direction. The corresponding points \mathbf{b}_p are identical and equal to the vertex \mathbf{v}_j . Thus, the $(\mathbf{b}, \mathbf{b}_p)$ pairs are illustrated in black in the right of Figure A.1. A point in the influence region of a vertex is always projected onto the vertex itself.

Note that the error measures in chapters 3, 4 and 5 are defined using surface points \mathbf{p}_i before translation and their projections. The coordinates of the projection of such a point can be obtained from $\hat{\mathbf{p}}_{i,\text{proj}}$ by appropriate translation. However, the distance between the original and the projected point does not change upon translation, so that computing the distance between $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_{i,\text{proj}}$ is sufficient for computing error measures.

Appendix B

Disorientation Angle Distribution

The DAD $p(\Theta)$ for cubes, under the assumption that all orientations of each cube are equally probable, was derived by Handscomb [132] and Mackenzie [131] and is presented below [132]. For an angle $\Theta \leq 45^\circ$:

$$p(\Theta) = \frac{2}{15}(1 - \cos(\Theta)).$$

For $45^\circ < \Theta \leq 60^\circ$:

$$p(\Theta) = \frac{2}{15}(1 - \cos(\Theta)) \left(3(\sqrt{2} - 1) \cot\left(\frac{1}{2}\Theta\right) - 2 \right).$$

For $60^\circ < \Theta \leq 60.6^\circ$:

$$p(\Theta) = \frac{2}{15} \left(\left\{ 3(\sqrt{2} - 1) + \frac{4}{\sqrt{3}} \right\} \sin(\Theta) - 6(1 - \cos(\Theta)) \right).$$

For $60.6^\circ < \Theta \leq \Theta_{\max}$, where $\Theta_{\max} = \arccos\left(\frac{1}{4}(2\sqrt{2} - 1)\right) \approx 62.8^\circ$:

$$\begin{aligned} p(\Theta) = & \frac{2}{15} \left[\left\{ 3(\sqrt{2} - 1) + \frac{4}{\sqrt{3}} \right\} \sin(\Theta) - 6(1 - \cos(\Theta)) \right] \\ & + \frac{8}{5\pi}(1 - \cos(\Theta)) \left\{ \arccos\left(\frac{\cot^2(\frac{1}{2}\Theta)}{3 + 2\sqrt{2} - \cot^2(\frac{1}{2}\Theta)}\right) \right. \\ & \quad \left. + \frac{1}{2} \arccos\left(\frac{\cot^2(\frac{1}{2}\Theta) - 2\sqrt{2}}{3 - \cot^2(\frac{1}{2}\Theta)}\right) \right\} \\ & - \frac{8}{5\pi} \sin(\Theta) \left\{ 2(\sqrt{2} - 1) \arccos\left(\frac{(\sqrt{2} - 1) \cot(\frac{1}{2}\Theta)}{\left(1 - (\sqrt{2} - 1)^2 \cot^2(\frac{1}{2}\Theta)\right)^{\frac{1}{2}}}\right) \right. \\ & \quad \left. + \frac{1}{\sqrt{3}} \arccos\left(\frac{(\sqrt{2} - 1)^2 \cot(\frac{1}{2}\Theta)}{(3 - \cot^2(\frac{1}{2}\Theta))^{\frac{1}{2}}}\right) \right\}. \end{aligned}$$