

Technische Universität München

TUM School of Computation, Information and Technology

Fast Constrained Clustering of Big Data: Theory and Application to Grain Mapping

Maximilian Fiedler

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Michael Ulbrich

Prüfende der Dissertation: 1. Prof. Dr. Peter Gritzmann
2. Prof. Dr. Andreas Brieden
3. Assoc. Prof. Dr. David Bourne

Die Dissertation wurde am 20.06.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 14.11.2024 angenommen.

I hereby confirm that this is my own work, and that I used only the cited sources and materials.

München, 20.06.2024

Maximilian Fiedler

Acknowledgments

I want to acknowledge the support I obtained during my research.

This endeavor would not have been possible without Prof. Peter Gritzmans continuous mentorship and support. His outstanding guidance, expertise, and commitment significantly shaped my path toward this thesis.

Additionally, I would like to express my deepest appreciation to my other coauthors, Dr. Andreas Alpers and Dr. Fabian Klemm, who generously provided knowledge and expertise. Moreover, I am extremely grateful to the generous support from the Studienstiftung des deutschen Volkes, who helped finance my research and broadened my network.

Many thanks should also go to Prof. Yinyu Ye for his fruitful thoughts and support in realizing my research stay at Stanford University.

Finally, I express my profound gratitude to my parents, who always stood behind me and supported me from day one and far beyond this thesis – many thanks for enabling my journey.

Abstract

English We design a framework for fast constrained clustering of big data. Constrained clusterings have a strong relationship with diagrams, making them popular for many applications like consolidating farmland, representing polycrystals, or generating superpixels. However, for large data sets, the size of the underlying linear program is beyond what is practically feasible to solve. To address this, we design coresets, small data sets that allow for a $(1 + \epsilon)$ approximation of the clustering problem. We apply our framework in the context of representing polycrystals, enabling us to compute accurate representations of three-dimensional materials quickly. With the representation obtained, we build a new dynamic model, uncovering that the growth process converges towards simple diagram types. We also design a new superpixel generation algorithm called Power-SLIC based on constrained clustering outperforming existing state-of-the-art object-based superpixel algorithms. Combining Power-SLIC with our coresets, we obtain a significant runtime speed-up for high-resolution images.

German Wir entwickeln ein Framework, um große Datenmengen unter Nebenbedingungen effizient zu clustern. Die resultierenden Clusterings ermöglichen eine gleichzeitige Berechnung von Diagrammen, womit sie für viele Anwendungen wie die Konsolidierung landwirtschaftlicher Flächen, der Darstellung von Polykristallen oder der Generierung von Superpixeln relevant sind. Allerdings wächst die Komplexität des zugrunde liegenden linearen Programms mit der Datenmenge, was das Lösen des Problems in der Praxis erschwert. Um dieses Problem zu adressieren, entwerfen wir Coresets, kompakte Datensätze, die eine $(1 + \epsilon)$ -Approximation des Clusteringproblems ermöglichen. Unser Framework erlaubt eine präzise Darstellung von Polykristallen in dreidimensionalen Materialien. Auf dieser Basis entwickeln wir ein neues dynamisches Modell und zeigen, dass der Wachstumsprozess zu einfachen Diagrammtypen konvergiert. Zusätzlich stellen wir einen neuen Superpixel-Algorithmus namens Power-SLIC vor, der unter Nebenbedingungen clustert und damit eine höhere Segmentierungsgenauigkeit als aktuelle objektbasierte Algorithmen erreicht. Die Kombination von Power-SLIC mit unseren Coresets reduziert die Laufzeit für hochauflösende Bilder signifikant.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Our approach	2
1.3	Constrained clustering and its connection to diagrams	4
1.4	Compression into coresets	7
1.5	Approximate constrained clustering: Grainscans and superpixels	13
1.6	Outlook and the structure of this thesis	19
1.7	List of core publications, one-page summaries, and approvals	20
2	Coresets for Weight-Constrained Anisotropic Assignment and Clustering	27
2.1	Introduction	27
2.2	Notation, background, and main results	30
2.3	Estimates, approximation, neighborhood mergers, and other ingredients	35
2.4	Small coresets	40
2.5	Final remarks	53
2.6	Technical results	53
3	Resolution Coresets	61
3.1	Introduction	61
3.2	Motivation, notation, and main results	62
3.3	Some basics for the analysis of resolution coresets	69
3.4	Proof of Theorem 3.4: The case $d = 1$	74
3.5	Proof of Theorem 3.4: The general case	82
3.6	Proof of Theorem 3.5	88
3.7	Final remarks	89
4	Turning Grain Scans Into Diagrams	91
4.1	Introduction	91
4.2	Turning grain scans into diagrams	92
4.3	Diagrams on sparse image supports	97
4.4	A direct model for APD CLASSIFICATION	104
4.5	Computational results	107
4.6	Outlook	115

5	Dynamic Grain Models via Fast Heuristics for Diagram Representations	117
5.1	Introduction	117
5.2	Diagrams and grain scans	118
5.3	Optimization-free heuristics	119
5.4	Comparison of H1 and H2	123
5.5	Dynamic grain growth	126
5.6	Final remarks	133
6	Power-SLIC: Fast Superpixel Segmentations by Diagrams	135
6.1	Introduction	135
6.2	Overview of our Approach	139
6.3	SLIC's Initialization & Assignment Phase	140
6.4	Generalized Balanced Power Diagrams	140
6.5	Optimal Power-SLIC and Power-SLIC	141
6.6	Evaluation	147
6.7	Conclusion	154
A	Inclusion of Published Core Publications	159
	Bibliography	209

Chapter 1

Introduction

1.1 Motivation

Data analysis has become more useful for discovering new insights, revealing hidden connections, and making informed choices. As a result, data sets of larger size and quantity have been gathered, pushing conventional methods to their computational limits. This thesis is motivated by the need to process large data sets efficiently, which has attracted considerable interest in the community. Surprisingly, we will show that smaller data sets can achieve almost the same quality of analysis, while greatly reducing the algorithm time. The analysis we are studying is clustering, often applied to large data sets to obtain manageable groups. Clustering partitions data points into clusters based on their similarity, enabling the detection of patterns and structures in a data set. Clustering is an unsupervised method and does not require labeling points, depending only on the defined similarity measure. This makes it a common pre-processing step: Meaningful groups can represent large amounts of unstructured data, and other methods can examine these groups more closely.

Additionally, clustering is a powerful tool by itself, for example, in partitioning the underlying space into regions. It can consolidate farmland, create voting districts, compute a market-clearing price vector, and represent polycrystals and many more [AR14; Alp+15; BBG14; BGK17; CCD16; EZZ04; GMW07]. Such applications can involve giant data sets; for instance, scans of polycrystals can easily contain around half a billion data points.

Whether clustering is the main analysis or a pre-processing step to analyze large amounts of data, the high complexity of many clustering algorithms hinders their widespread application to such large data sets. For example, for the popular least-squares clustering, we are given a $k \in \mathbb{N}$ and n points in d dimensional space and want to find k clusters that minimize the sum of the squared Euclidean distances of each point to its nearest center. It has been shown that this problem is NP-hard, even when $d = 2$ and k is part of the input [MNV12] or when d is part of the input and $k = 2$ [Alo+09; Das07]. It even exists an $\epsilon > 0$ for which it is NP-hard to approximate the problem up to $(1 + \epsilon)$ when both k and d are part of the input [Awa+15].

Therefore, new techniques are required to cluster large data efficiently, as typical

methods fail. Although research for the classical k -means problem progresses fast, different clustering techniques, such as clustering with constraints, have yet to be addressed or pose new challenges to overcome. In the following, we summarize our approach to tackling these challenges.

1.2 Our approach

We design a new framework for computing diagrams via constrained clustering fast. We achieve this by reformulating the constrained clustering problem on a much smaller, computationally manageable instance. We generate such an instance by compressing the full data set into a smaller one called coreset. Our approach for creating the coresets provides provable performance guarantees for the clustering problem. Finally, we evoke an existing relationship between constrained clusterings and diagrams for the small instance to obtain diagrams. This new framework is depicted in Fig. 1.1.

To validate our framework, we demonstrate our techniques in two applications. First, we use it to quickly convert three-dimensional grain maps into diagram representations, and second, we use it to quickly segment high-resolution images into superpixels.

We first explain the link between diagrams and clusterings that this research relies on, before we describe our compression methods in detail. The core of this elegant relation is fast to describe: An optimal clustering of a specific type is connected to a diagram of a particular type. The consequence of this relation is equally simple and powerful: Given an optimal clustering, we can compute a diagram dissecting the underlying space, and vice versa, we can compute an optimal clustering for a data set, given a diagram.

Multiple applications like electoral district design or farmland consolidation have successfully used this framework. A constrained data set clustering is computed to obtain a diagram dissecting the underlying space in both applications. While this approach works great for small data sets, the underlying constrained clustering problem is notoriously hard. The involved algorithm requires solving a memory-hungry linear program when the size of the data set grows. As a consequence, current applications are limited to smaller data sets.

Our approach to tackling this problem is to approximate the underlying optimization problem of constrained clustering. While there has been some progress in developing classical approximation algorithms for clustering, a more promising technique is too first to compress the data set and apply the standard clustering algorithm to the compressed data set. Although the compression may discard some of the important information, for clustering, it can be demonstrated that proper compression preserves most of the information that matters. We design such compressed data sets for the constrained clustering problem – called coresets – that have performance guarantees and result in small runtimes. The coresets enable us to use the existing relationship

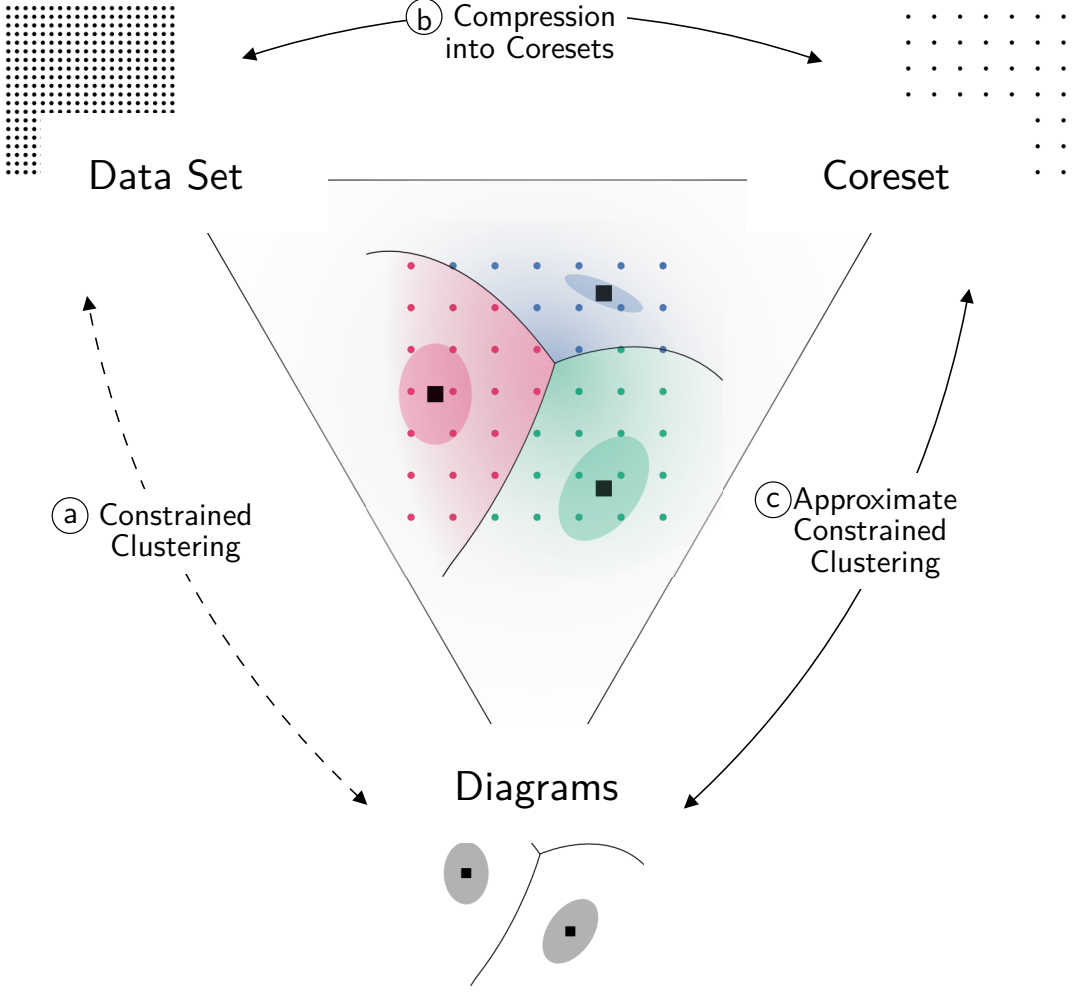


Figure 1.1: Structure of the work presented in this thesis. The dotted arrow shows existing work.

between clustering and diagrams to generate diagrams via constrained clustering of the coreset quickly.

We now introduce our framework’s elements shown in Fig. 1.1 more thoroughly, list our contributions, and relate them to existing literature. We begin by clarifying (a) constrained clustering and how it relates to diagrams. Next, we describe (b) our compression methods. Finally, we assess (c) the approximate constrained clustering in two different applications. In the first application, we examine how to represent polycrystalline materials using diagrams, and in the second, we develop a new superpixel algorithm to segment images based on our framework.

1.3 Constrained clustering and its connection to diagrams

We start by explaining our main clustering problem and its connection to diagrams. Clustering is a widespread method of unsupervised learning aiming to classify a data set into groups such that data points within each group or cluster are similar. In contrast, data points from two different groups are dissimilar. Despite being unsupervised, defining similarity is left to the user.

The seemingly most well-known measure of similarity is the sum of least squares used by the famous k -means algorithm [Jam+13]. The algorithm identifies k different groups of points such that the sum of the squared Euclidean distance of each point to its cluster centroid, defined as the weighted sum of cluster points, is minimized. Therefore, each point is assigned to the closest cluster centroid. Another implication is that the problem can be reduced to finding k optimal cluster centroids as the resulting assignment of points to cluster becomes immediate.

A different perspective on this is that an optimal clustering provides a dissection of the underlying point space that assigns each point to the closest centroid, with some points lying on the hyperplane that is equidistant from two centroids. Such a dissection of the space has been well studied and is known, besides many other names, as the Voronoi diagram in the literature [AKL13]. Voronoi diagrams have been used in many contexts, particularly in understanding naturally occurring structures in biology, materials science, geography, and social sciences [Kin84; Now33; NOW76; Oka+00; Sai82].

In the context of clustering, we also know that an assignment to k points, often called sites, is least squares minimal if and only if all points assigned to a site also lie in the cell of the Voronoi diagram for the same k sites. This structural property of the sum of least squares objective makes it useful in applications where a finite point set is provided, but a complete dissection of the space is required. For example, a diagram is a convenient tool to assign future data points to identified groups.

However, the Voronoi diagram structure of k -means clustering can also limit the flexibility of the clustering types we can find. One element we cannot control with Voronoi diagrams is the number of points in each cluster, often termed the weight of each cluster. This is helpful in many situations, such as electoral district design, where each district must contain a similar number of citizens by law [BGK17]. As this clustering problem defines lower and upper bounds on the weight – here, the number of citizens – in each cluster, we also call it weight-constrained clustering.

Another example of constrained clustering is fair clustering, which aims to avoid under- or over-representing specific characteristics such as ethnicity or educational background by setting lower and upper quantitative bounds for each item of the named attributes [Chi+17].

Fortunately, adding weight constraints to least squares clustering still results in optimal clusterings corresponding to diagrams often called power diagrams [Aur87].

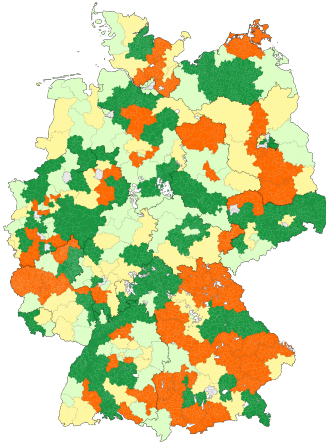
Power diagrams are an extension of Voronoi diagrams: Each site is equipped with a size parameter that controls the unit ball size of the Euclidean distance associated with each site in the diagram. Reducing the size of one unit ball will reduce the volume of the corresponding diagram cell while increasing the volume of others. Remarkably, we can always find size parameters such that the weight assigned to each cluster matches the total point weight in the corresponding diagram cells [BG12].

In fact, given an optimal weight-constrained clustering, we can always compute a diagram dissecting the underlying space, and vice versa, we can compute an optimal weight-constrained clustering of a data set, given a diagram. Even more generally, this relationship is sustained when ellipsoids are used as the unit balls for the diagram [BGK17]. The least squares objective becomes anisotropic; consequently, the diagrams are often called anisotropic power diagrams [Alp+15; BWY06; CG11; Kle20; LS03; Wor08].

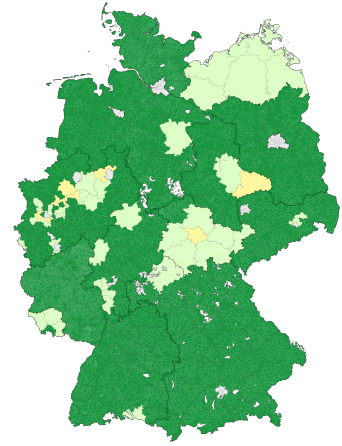
One application that successfully uses weight-constrained clustering to obtain anisotropic power diagrams dissecting the underlying space is electoral district design [BGK17] as depicted in Fig. 1.2. The challenge in this application is to design electoral districts that contain a similar number of citizens and are relatively compact to avoid forms of Gerrymandering. The authors of [BGK17] use the centroids of current electoral districts with their second moments to define a unit ball per district. Weight-constrained clustering, where each citizen corresponds to a point with weight 1 and each cluster is required to have an identical weight up to some allowed deviation, ensures regular voting districts with a similar number of citizens per district. At the same time, the choice of centroids and second moments as shape information ensures that the new districts are geographically similar to the previous districts. Hence, many citizens remain in their current district, and existing infrastructure, such as voting offices, can be reused.

Another application that obtains anisotropic power diagrams via weight-constrained clustering is representing grain scans via diagrams in materials sciences. The underlying goal is to understand the structure of polycrystalline materials for which empirical evidence suggests that such structures, called grains, can be nicely represented by anisotropic diagrams. The authors of [Alp+15] use weight-constrained clustering to generate a diagram representation from a two-dimensional finite-resolution scan of a material. The advantage of representing such a grain scan with a diagram is two-fold. First, a diagram has only a few parameters per grain compared to hundreds of pixels, making it much more space-efficient to store. Second, such a low-parameter representation can be easily used in dynamic models to analyze grain growth, something we consider in Chapter 5.

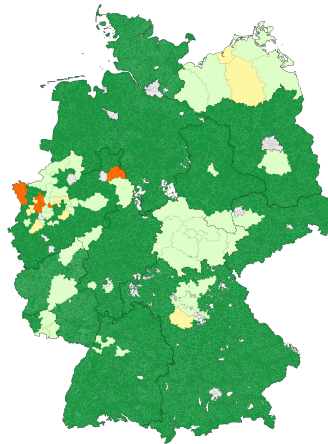
We have introduced the notion of weight-constrained clustering and pointed out multiple applications. For a mathematical introduction to the exact problem, we refer to Chapter 2. For an in-depth background on the application of representing grain scans, we refer to Chapter 4.



(a) German voting districts in 2017.



(b) Refined voting districts using power diagrams computed via constrained clustering.



(c) Refined voting districts using anisotropic power diagrams computed via constrained clustering.

Figure 1.2: Voting districts in 2017 and refined via constrained clustering. The color shows the deviation from a uniform district size. Dark green: $[0\%, 5\%]$, Light green: $(5\%, 10\%]$, Yellow: $(10\%, 15\%]$, Orange: $(15\%, 25\%]$. The power diagram shows the least deviation. However, it changes the district the most compared to the 2017 districts. The anisotropic power diagram provides a tradeoff between both. Pictures are taken from [Kle20].

1.4 Compression into coresets

We now describe and motivate the second element of our framework, the compression into coresets. We already mentioned that least-squares clustering is hard. Since the unconstrained case is a special case of the constrained clustering problem we will examine, all hardness results also apply to it. Beyond that, solving a single assignment for given clustering centroids will require solving a linear program with the number of variables being the product of the number of points in the data set and the number of clusters. Data sets with more points quickly hit practical limitations such as memory limits. In such cases, computing diagrams via constrained clustering becomes challenging.

Thus, we study whether the underlying clustering problem can be approximated. More specifically, we examine whether one can compress the underlying data set such that a constrained clustering of the compressed data set is a good approximation for a constrained clustering of the full data set. Compressed data sets that allow for such approximation are called coresets in the literature [AHPV05; Phi17]. More precisely, we speak of an ϵ -coreset if the coreset allows an ϵ approximation.

Coresets first evolved while studying the geometric properties of point sets [AHPV05], for example, when finding the circumradius of a point set's convex hull. In this context, a coreset is described as a small point set with a comparable circumradius, usually a subset of the original point set. This allowed us to obtain an approximation of the true circumradius by computing it on the much smaller point set with a much smaller computational cost.

Suppose we interpret the circumradius not as a property but as an optimization problem – minimizing the ball's radius such that it still contains the point set. In that case, it is natural to introduce the concept of coresets to further optimization problems on data. Coresets have been studied for problems such as linear regression, logistic regression, kernel regression [BDMI13; Das+08; Mun+18; ZP17], support vector machines for two and several classes [AMS07; TKH05; TKZ06], least-squares clustering [Fel+11; FSS13; Fen+19; FS06; FS05; HPM04; HV20], projective clustering, Gaussian mixture models, subspace approximation and other geometric problems [AB09; AHPY08; Ass+19; Bay+18; Bā+02; Cla10; DP21; Des+06; Fel+06; FL11; Fel+10; FSS13; Hua+18; Mus+22; PT18].

This thesis develops the first coreset for weight-constrained anisotropic least-squares clustering. We build on results for unconstrained least-squares clustering that we survey next while pointing out the additional challenges that constraints impose. We differentiate between geometric and sampling techniques and explain why the latter cannot be generalized to the constrained case, requiring us to develop new strategies.

Geometric techniques for unconstrained clustering

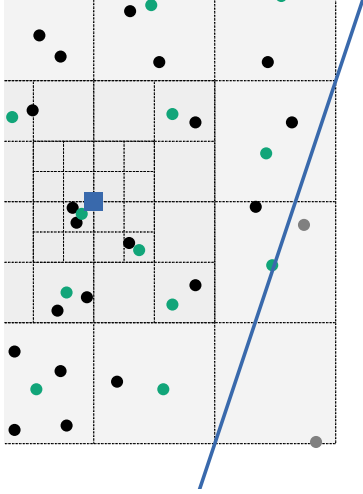
The first ϵ -coreset for least-squares clustering was developed in [HPM04]. The idea is that the cost of expensive points in a least-squares clustering, i.e., those far from their nearest cluster centroid, can absolutely deviate more than points close to their nearest cluster centroid to achieve a relative ϵ -approximation. The authors also use that considering the cost of points in a constant-factor approximation, for which fast algorithms are available, is sufficient. For each centroid in such a constant-factor approximation, they construct an exponential grid depending on the chosen ϵ that is dense near the clustering centroid of a constant-factor approximation and becomes looser with distance. The coreset is obtained by replacing all points in a common grid cell with a single weighted point. This constructed point set has $\mathcal{O}(\log n \epsilon^{-d})$ points, with n being the number of points in the original data set and d its dimension. In the following, we also say that a coreset is of size t when the coreset has t points. The construction is visualized in Fig. 1.3a.

This technique has been improved by replacing the exponential grid with a pencil of lines that all meet in the cluster centroid [HPK07]. Data points will then be snapped to their nearest line. Points on each line are grouped and replaced by two points per group that fulfill specific properties in terms of position and error. The advantage of this method is that the error in each such group can be relatively large as at most $k - 1$ such groups per line will be incorrectly clustered by an optimal clustering on the coreset. This is a consequence of the relation between least-squares clusterings and diagrams described in the previous section.

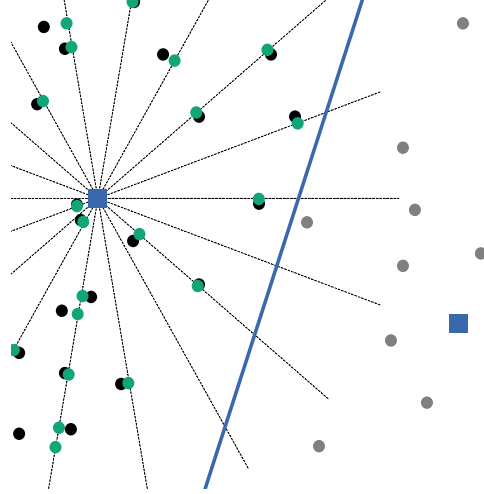
The resulting coreset has $\mathcal{O}(k^3 \epsilon^{-(d+1)})$ points, which was the first coreset for least-squares clustering with size independent of the number of points in the original data set. The construction is visualized in Fig. 1.3b

Sampling techniques for unconstrained clustering

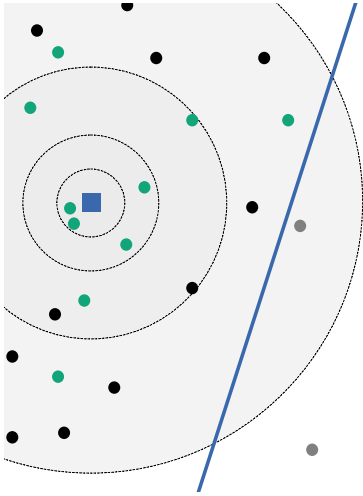
While the ray construction removed the dependence on the weight and number of points in the data set, the exponential dependence on the dimension remained. Chen first developed a coreset that does not depend exponentially on the dimension but logarithmically on the number of points in the data set [Che09]. They used an idea similar to exponential grids but assigned points to spherical zones rather than grids. Again, these zones become exponentially larger with distance from a centroid of a constant factor approximation. A constant number of points is independently and uniformly sampled for each spherical zone with replacement. If a zone contains fewer points than the number that is supposed to be sampled, all points in the zone are included instead. After sampling, points are reweighted to represent the original total point weight. All in all, this results in a coreset of size $\mathcal{O}(dk^2 \epsilon^{-2} \log n \log(k/\epsilon))$. Figure 1.3c depicts this construction.



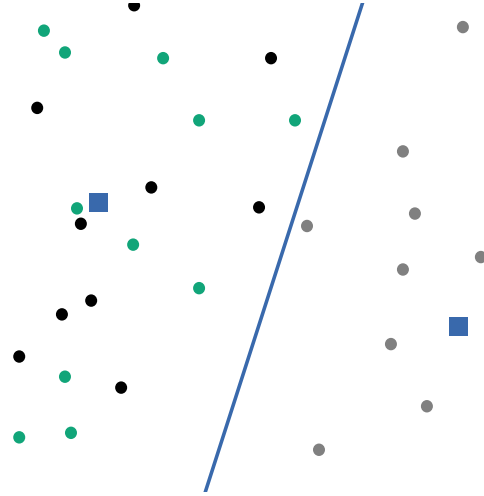
(a) An exponential grid is constructed. Replacing all points in a grid cell with a weighted representative leads to a coresets. The point weight is not being visualized. The coreset is shown in green.



(b) Rays are constructed. Projecting each point onto its closest ray leads to several one-dimensional point groups. A rather complex grouping process for each line reduces the number of points on lines, which is not shown here, results in the coreset.



(c) For each circle, a constant number of points is uniformly sampled with replacement or, if the number of points is smaller than the number of points to be chosen, all points in the circle are included in the coreset. Note that the chosen green points are a subset of the original data set.



(d) The importance is computed based on the cost in a constant factor approximation, and points are sampled proportionally to this importance.

Figure 1.3: Different coreset techniques. Each image shows a point set depicted by black and grey points. A constant factor approximation is shown with blue centroids. The black points are assigned to the left cluster for which a coreset technique is illustrated.

This construction implicitly captures the notion of sensitivity or importance, which was developed shortly afterward. The construction in [Che09] essentially works because points in the same spherical zone have similar sensitivity and can thus be uniformly sampled. This notion of sensitivity is first introduced in [FL11] and described next.

We can precisely specify the cost of a point in a clustering as the distance toward its assigned site. The idea is now to measure the cost share of a point of the total clustering cost, i.e., the distances of all points to their closest sites. This information is not particularly helpful for a single choice of k sites. However, suppose we know the supremum of the relative cost contribution of a point to all possible sets of k sites. In that case, we can judge the importance of a point: If a point contributes a high relative cost to at least one set of k sites, we consider the point important; if a point contributes a low relative cost to all possible sets of k sites we consider the point as less important.

Connecting this to the notion of sensitivity, an unimportant point is considered insensitive towards the choice of the clustering sites as its relative cost contribution is always low. For unconstrained clustering, it can now be shown that the sum of sensitivity of all points in the full data set is bounded by $\mathcal{O}(k)$, i.e., the number of required clusters. Consequently, there are always points insensitive toward the choice of the sites. Importance or sensitivity sampling samples the points proportionally to their importance, ensuring to collect important points with high probability and those insensitive to the choice of sites with low probability. After sampling, points are reweighted to represent the original total point weight. The resulting data set can be shown to be a coresset of size $\mathcal{O}(d\epsilon^{-2}k \log k)$ [Bra+21] showing an improvement from quadratic to linear dependence over earlier results [HV20; SW18].

In practice, a coresset can be computed by upper bounding the importance or sensitivity of the points by considering their costs in a constant factor approximation. Then, the point set is sampled proportionally to the so-estimated sensitivity, only leading to an increase in coresset size by a constant depending on the quality of the bound on the sensitivity.

This fact connects importance sampling to the uniform sampling in the construction by Chen. As the bound on the importance of each point is derived from a constant factor approximation, we observe that points with a similar distance from their nearest center have a similar sensitivity bound. Hence, importance sampling is similar to uniformly sampling the points in the spherical zones given by Chen. The advantage of importance sampling is that it removes the need to construct these zones and thus does not incur a dependence on the number of points.

Coresets for constrained clustering

While a range of coresset results for the unconstrained clustering problem exists, research for the constrained problem could be more advanced. Schmidt, Schwiegelshohn, and

Sohler designed a coreset for fair clustering [SSS18]. Fair clustering is a constrained clustering problem ensuring no cluster over- or underrepresents certain point characteristics. Such point characteristics are associated with each point in advance and are categorial in nature, such as age groups or ethnicity when points represent people. Their coreset is of size $\mathcal{O}(lk\epsilon^{d-1}\log n)$, where l is the cardinality of the set characteristic combinations.

Essentially, their proof reduces fair clustering to constrained clustering with a single characteristic (a special case of the weight-constrained clustering problem in which each point has weight 1) by showing that a coreset for each l characteristic combination in the point set can be constructed separately. Then, they show that coresets constructed by moving each point only a small amount compared to the optimal clustering cost are also coresets for the constrained problem. For example, the exponential grid coreset from [HPM04] fulfills this property which also explains the coreset size in [SSS18] for fair clustering.

This result has been followed by a result in [HJV19] constructing a coreset of size $\mathcal{O}(lk^3\epsilon^{-d-1})$ by extending the pencil techniques in [HPK07] to constrained least-squares clustering.

We further improve both results in [BFS24; HJV19] by improving the techniques from [HPK07] using simpler and smaller constructions. This allows us to reduce the coreset dependence on k from cubic to quadratic. Thus, we obtain a coreset for weight-constrained clustering of size $\mathcal{O}(k^2\epsilon^{-d-1})$ and a coreset for fair clustering of size $\mathcal{O}(lk^2\epsilon^{-d-1})$. At the same time, our simplified construction is less dependent on the Euclidean space, which means that our design is the first coreset for the anisotropic objective function to the weight-constrained least-squares clustering problem. Of course, our improved geometric coreset technique also applies to unconstrained least-squares clustering. To our knowledge, our coreset currently is the smallest coreset for weight-balanced and fair clustering and is the smallest coreset for general least-squares clustering constructed geometrically without the need for sampling.

Why are geometric techniques still relevant, given that sampling leads to much smaller coresets for unconstrained techniques? In fact, sensitivity sampling has been applied very indirectly to a constrained clustering problem in [BFS24] already to obtain a coreset without the exponential dependence on d . This proof essentially generalizes that of [Che09] to the constrained case.

Unfortunately, the resulting coreset is of size $\mathcal{O}(\epsilon^{-5}k^7(\log n)^6(\log n + d\log(\epsilon^{-1})))$ in the Euclidean space and hence, while logarithmically, still very dependent on n . For our application, where both k and n are large, and d is small, this does not result in an improvement.

Additionally, it remains an open question whether importance sampling can be directly applied to construct coresets for the constrained case as opposed to going through the techniques used by [Che09]. We show that this is not the case as in such a setting, each point in a data set can have a sensitivity arbitrarily close to 1. This

breaks a crucial prerequisite of importance sampling, requiring the sensitivity to be bounded. This result emphasizes why geometric techniques, as studied in this thesis, remain highly relevant.

Beyond these contributions, we also develop coresets that adhere to grid structures. These are targeted to applications such as representing polycrystals that rely on such structures. While coresets adhering to them generally result in worse dependence on the dimension, we show that for the relevant cases, i.e., when the dimension is two or three, grid coresets can contain fewer points. In the following, we provide a list of our contributions.

Our contributions to designing coresets

Here, we summarize our contributions towards designing coresets for weight-balanced anisotropic least-squares clustering that we detail in Chapters 2 and 3.

- We give a new generalized coreset definition that considers non-Euclidean anisotropic objective functions. Moreover, our definition explicitly requires mappings between the data set and its coreset. In the constrained case, this is crucial for quickly mapping solutions between both data sets.
- We give a new simplified one-dimensional coreset construction, which requires half as many points as the existing construction. This is critical as it significantly reduces the number of variables in the involved linear program, which decides if the linear program fits into memory.
- Using this and additional techniques, we present the first coreset construction for the case of the anisotropic objective function and weight constraints of size $\mathcal{O}(k^2\epsilon^{-d-1})$.
- As a corollary, we obtain the first coreset construction for fair clustering of size $\mathcal{O}(lk^2\epsilon^{-d-1})$ reducing the dependence on k from cubic to quadratic.
- As a second corollary, we obtain the first coreset construction for unconstrained least-squares clustering of size $\mathcal{O}(k^2\epsilon^{-d-1})$ that does not require randomized sampling reducing the dependence on k from cubic to quadratic.
- For the unbalanced least-squares clustering, many smaller coresets have been designed by bounding a value called sensitivity. We show that this concept cannot be applied to constrained clustering by providing an example for which the sensitivity is arbitrarily large. This negative result highlights a large gap between constrained and unconstrained clustering and opens new research directions. Additionally, it focuses our attention on constructing smaller geometric coresets.

- We also show that we can construct coresets for the weight-constrained clustering with a grid structure. These special structures are advantageous when representing polycrystals, as the relevant data sets are naturally collected in a pixel or voxel image/grid structure. Consequently, we can scan materials upfront in a lower resolution, reducing the data collection effort significantly. Theoretically, these grid coresets are of size $\mathcal{O}((k\epsilon^{-2/3})^d)$ which in the relevant application parameters, in particular, $d = 2$ or $d = 3$ and ϵ around 0.001 results in even smaller coresets than our more general coreset we mentioned above of size $\mathcal{O}(k^2\epsilon^{-d-1})$.

Going beyond what is described in this thesis, our coreset results also imply a polyhedral approximation for the so-called gravity polytope (see [BG12] for a definition of this polytope).

1.5 Approximate constrained clustering: Grainscans and superpixels

Having built the foundation to approximate weight-constrained anisotropic clustering using coresets, we pivot towards using our techniques in two applications. First, we use our coresets to facilitate the representation of polycrystalline materials by developing new algorithms and designing dynamic models. Second, we develop a new algorithm to generate superpixels via diagrams outperforming the currently best diagram techniques.

1.5.1 Representing grain scans

Materials scientists aim to understand material characteristics and their evolution under certain influences. They often rely on mathematical models to study the structure of polycrystals [Ban08; Han+09; Pou04]. Modeling requires a representation of the material, such as a voxel-based or object-based representation. The latter is usually preferred, resulting in much more space-efficient representations. At the same time, objects simplify the task of modeling as they are generally parametrized [BBH16; Gui+11; MS07; MSF12; TR18; XL09; Zha+18].

In particular, geometric objects such as Voronoi diagrams and their derivations have been used multiple times to represent grain scans as the polycrystalline materials seem to naturally adhere to a diagram-like structure [Alp+15; Alt+14; KS08; Lyc+11; Pet+21; Šed+16; Šed+17; Spe+16; Spe+14; TGB15; TR18; Tel+92]. The authors of [Alp+15] have achieved state-of-the-art representation accuracy using their GBPD algorithm, which comes at a high computational cost. Consequently, applications have been limited to two-dimensional representations of materials.

Essentially, their algorithm uses weight-constrained anisotropic clustering to compute a dissection of the grain scan into anisotropic power diagrams. Such scans can easily have millions of voxels and, thus, render the required linear program often untraceable. We

approach this problem by introducing our coresets techniques to the field. Consequently, we develop an approximate version of the GBPD algorithm to compute a high-accuracy diagram representation of a 68,000,000-voxel grain scan in less than 3 minutes.

Besides our general coreset and grid-based coreset techniques, as introduced in the previous section, we also develop and mathematically justify a technique called interior removal that exploits information about the optimal sites. A visual comparison of all our methods is depicted in Fig. 1.4.

Despite the clustering techniques providing excellent accuracy results, they optimize solely over the size parameters of the diagrams. The sites and ellipsoids, i.e., the location and shape information, are estimated from the data set in that technique. Benefiting from our compressed data sets, we develop a new optimization model optimizing all diagram parameters.

We are verifying and comparing our techniques on a real-world data set. We address the performance of our algorithms and compare them with the heuristics established so far in the literature. We also address how far we can compress a grain map. Our evaluation shows that we can compress a real-world grain map from 68,000,000 parameters to below 100,000 data points. We conclude that in most cases, the established heuristics work very well, being now able to compare its performance to multiple optimization techniques.

Our analysis and experiments show that an even simpler heuristic often suffices. We analyze an existing heuristic by [Lyc+11] by deriving it using discriminant analysis. Arguing for a different prior, we obtain a new heuristic.

Ultimately, we would like to benefit from these algorithms that accurately and quickly generate representations of materials. Specifically, modeling the grain growth is of high interest as this process determines the properties of the material [Pou04; RRH04]. Initial models have been studied by Smith, von Neumann, and Mullins in the 1950s [Mul56; Neu52; Smi52]. Despite continuing interest in this topic, a study in [Zha+18] concludes that classical models by Hillert [Hil65] and the MacPherson-Srolovitz model [MS07] do not adequately capture anisotropic grain growth. As such, modeling grain growth remains a vital area [HTS18; Hu+21; RZ18].

We design a continuous grain growth model by interpolating and extrapolating from discrete time steps obtained from a grain scan. Our inter- and extrapolation is based on the diagram parameters that we obtain from our algorithms. We compare various interpolation techniques and generally observe a good fit even for time steps excluded in the fitting process.

Based on this model, we also show a correlation between the size parameter of the diagram and grain appearance. This opens the interpretation that grains with small size parameters in their representation are energetically more favorable and thus remain in the polycrystalline structure. In contrast, others are more likely to disappear.

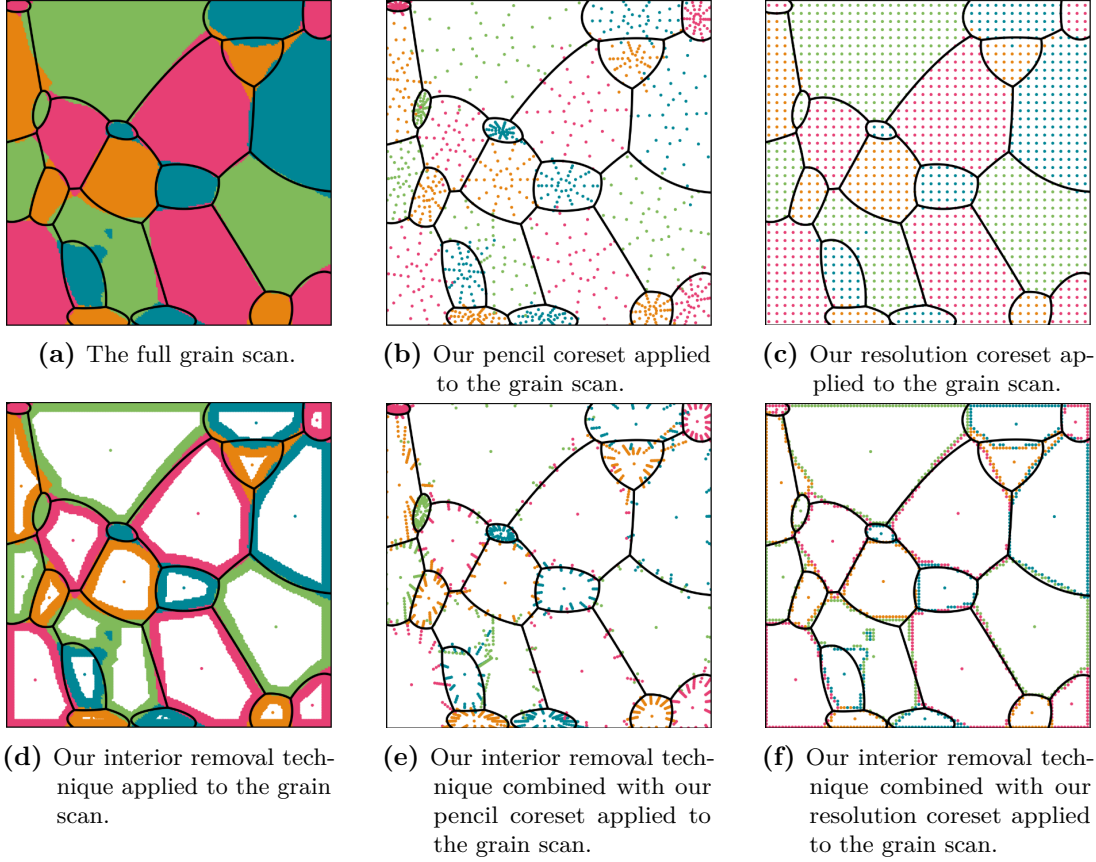


Figure 1.4: Multiple point reduction techniques applied to a two-dimensional grain scan. Shown is also the resulting diagram computed using each sparsified point set. One observes that the resulting diagram is almost identical for all cases, showing that our point reduction methods effectively compute diagram representation for grain scans.

Our contribution to representing grain scans

We summarize our contributions in Chapters 4 and 5 towards this field in the following.

- We design a new algorithm converting grain maps into diagrams with significantly decreased running times. For this, we leverage our developed coreset techniques to approximate a clustering problem in a method by [Lyc+11] to convert grain maps into diagrams.
- Additionally, we introduce additional problem-specific techniques sparsifying the grain map data further using the concept of interior removal.

- We develop a new optimization model optimizing over a larger parameter space, i.e., all diagram parameters. Combined with our compression technique, we achieve good running times.
- We verify and compare our techniques on a real-world data set. Our evaluation shows that we can compress a real-world grain map from 60,000,000 parameters to below just 100,000 data points.
- We conclude that in most cases, the established heuristics worked very well, being now able to compare its performance to multiple optimization techniques.
- We design a new heuristic for generating anisotropic power diagrams by deriving a heuristic from [Lyc+11] via discriminant analysis and arguing for a different prior.
- We develop a new dynamic grain diagram-based growth model that inter- and extrapolates to obtain a continuous model.
- Looking at the change in diagram parameters throughout the model, we state the hypothesis that the grain growth process converges towards a specific diagram type. We conclude that diagrams with size parameters close to zero seem energetically favorable.

1.5.2 Generating superpixels

In our second application, we introduce the concept of constrained clustering and coresets to the domain of superpixel generation. Superpixel algorithms meaningfully group image pixels into superpixels, reducing the complexity of up-stream image analysis tasks such as image segmentation, object detection, image classification, super-resolution, image compression, and many more [Ach+12; Che+18; CSL20; Don+20; FVS09; Gou+08; Gou+14; Li+16; MRR21; MG15; Mor05; RM03; SARS20; Shu+11; SHL18; Yan+15; ZK07]. The concept of superpixels was first introduced by Ren and Malik in [RM03] and has significantly gained attention until today.

One such recent research direction focuses on developing geometric superpixels as these are generally more compact and already save additional second-level information that upstream algorithms can use [AS17; BL18; DL15; KM20; Ma+21]. Due to the continuous nature of geometric objects, they provide resolution-independent superpixel representation and, thus, a better alternative for many applications. Examples include diverse computer vision tasks such as object contouring [BL18; Ma+21], image vectorization [CXC14], image compression [FKF16; Ma+21], and scene segmentation [DL15].

We design a new superpixel generation algorithm that creates superpixels defined by diagram structures called Power-SLIC. For that, we build on a popular superpixel algorithm called SLIC [Ach+12]. Since its introduction, various adjustments have

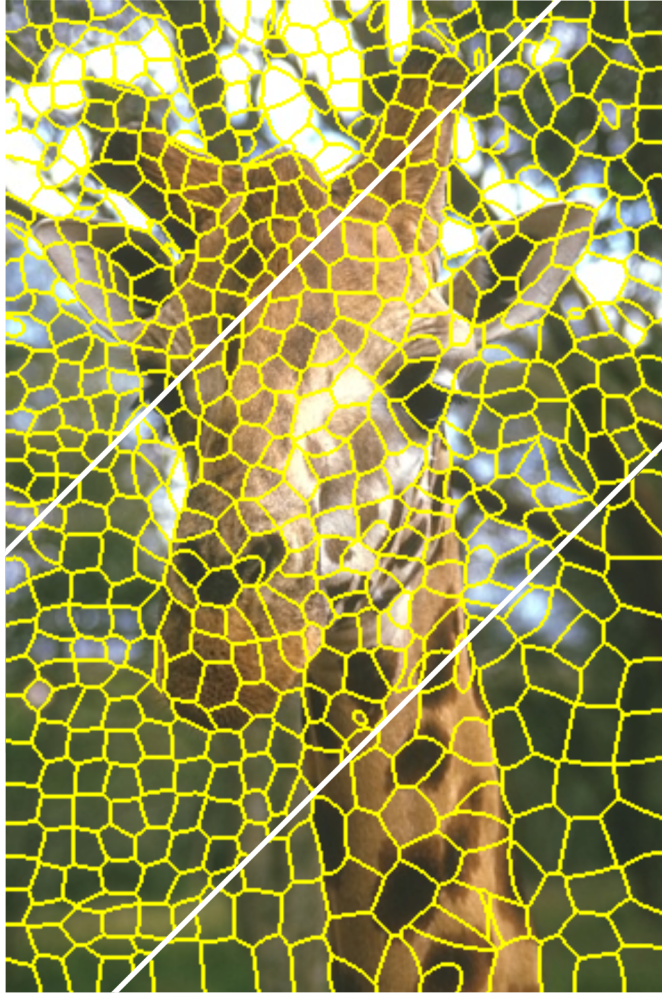


Figure 1.5: An example of an image segmentation into superpixel. The segmentation has been generated using our superpixel algorithm Power-SLIC with 900, 600, and 300 superpixels. The image is taken from the BSDS500 data set [Mar+01].

been introduced [Ach+20; AS17; DZ16; Don+20; HZX21; Liu+16; Liu+18; Mai+18; Zha+21; ZCL19].

Essentially, SLIC uses least-squares clustering to group image pixels based on the pixel’s spatial and color information. A heuristic version of Lloyd’s algorithm is used to retrieve the superpixels very quickly. Consequently, SLIC is valued for its speed, accuracy, and ease of use. Due to SLIC’s approach of assigning pixels in the spatial and color space to groups and then projecting that assignment back into the spatial image space to create superpixels, they can be somewhat irregular and non-compact.

We replace this crucial step by using diagrams. We use the information generated in the spatial and color space to compute sites, weights, and second-moment information to define an anisotropic power diagram in the image space. We then use constrained clustering to obtain an assignment of pixels to superpixels when required.

This approach generates more compact and regular superpixels and provides a resolution-independent and space-efficient representation of the superpixels by a diagram structure. We can exploit this to compute superpixel segmentation for high-resolution images quickly. Instead of extracting the diagram via clustering the entire image, we can compute it on a much lower-resolution version. Once we collect the diagram parameters, we can easily scale them to the original resolution image and assign each pixel to its diagram cell to compute a pixel segmentation into superpixels. This allows us to compute superpixel segmentations for high-resolution images in milliseconds instead of seconds. We summarize our empirical evaluation, comparing it to other state-of-the-art superpixel algorithms and all properties of our algorithm in the following. Superpixels generated by Power-SLIC are depicted in Fig. 1.5.

Power-SLIC: Evaluation and properties

Superpixel algorithms are typically assessed using three criteria: Boundary adherence, regularity, and computation speed. Using diagrams in combination with coresets guarantees some very favorable properties of Power-SLIC. On the BSDS500 data set, a popular benchmark, our algorithm outperforms multiple state-of-the-art algorithms in particular:

- **Boundary Precision:** Power-SLIC outperforms the other evaluated state-of-the-art algorithms regarding boundary precision. As the diagram structure acts as regularization for the superpixel boundaries, Power-SLIC requires fewer boundary pixels in image areas that do not contain image boundaries than other algorithms.
- **Boundary Recall:** Due to this regularization, Power-SLIC is the most robust against varying noise levels in terms of boundary recall. While there are algorithms that outperform Power-SLIC in terms of boundary recall on either clean images or noise images by algorithms designed for these situations, Power-SLIC is the best choice for scenarios with varying noise levels. Such scenarios are common, as any camera operating in varying light conditions usually adjusts the ISO value in darker environments, which generally adds noise.
- **Regularity:** As resulting superpixels are diagram cells, the corresponding superpixels are very compact and regular. Moreover, superpixel boundaries are polynomials of degrees at most two. Overall, Power-SLIC outperforms the other algorithm in terms of regularity.

- **Speed:** Power-SLIC is based on SLIC. While the final steps significantly differ, the runtime of both algorithms is dominated by the common assignment step. Since SLIC is well-known for its speed and Power-SLIC's runtime is essentially identical, our algorithm is competitive in terms of speed.
- **Resolution-Independence:** Our diagram representation of superpixels adds multiple advantages. Most importantly, this representation is resolution-independent. It allows us to compute a superpixel segmentation on a low-resolution image instead of a high-resolution image. Combining this with our coresets from the previous section, we get quality guarantees for this approach. Moreover, Power-SLIC significantly outperforms the other algorithms in our evaluation regarding speed on high-resolution images.

1.6 Outlook and the structure of this thesis

We introduce a framework for computing constrained clustering and corresponding diagrams even on massive data sets very quickly. We design a new and currently smallest coreset for weight-constrained anisotropic least-squares clustering. Beyond that, we create grid-based coresets that produce provably smaller coresets for the relevant parameters in our application. We empirically prove our techniques and develop new algorithms for representing grain scans. Lastly, we design a new superpixel algorithm based on our framework generating compact and resolution-independent superpixels resulting in significantly faster running times for high-resolution images.

Of course, further reducing the size of coresets for weight-constrained clustering is an important open research question. In particular, the large gap of achievable coreset sizes between the constrained and unconstrained case due to the sensitivity being bounded for the former and unbounded for the latter piques the interest in new coreset techniques.

However, the primal problem of approximating the coreset cost is still much better understood as the dual approximation of finding appropriate diagrams: While our theory guarantees $(1 + \epsilon)$ approximations and our empirical evaluation evidently shows that the resulting diagrams provide a very good fit, it remains unanswered if the diagrams generated via approximative constrained clustering of the coreset are close to those generated via constrained clustering of the full data set.

Our early follow-up research beyond this thesis shows that an arbitrarily small perturbation of a single point can result in an arbitrarily large distance between the same bisector when comparing the diagram generated using the full data set and the diagram generated using the coreset. When defining the closeness of two diagrams by comparing the distance of their bisectors, these examples show that an approximation of diagrams cannot be easily guaranteed.

Considering our good empirical results, we raise the question of which conditions on the point set or the coresets also ensure good approximations of the diagrams. In other words, what conditions avoid the worst case currently constructed in our early examples based on local point sparsity and limit cases with arbitrarily large distances between sites or a growing number of clusters? We detail further open research questions at the end of each chapter. The rest of this thesis is structured as follows.

This chapter continues with a 1-page summary of each following chapter. In Chapters 2 and 3, we prove our coresets results. This builds the foundation for our work on representing grain maps via diagrams in Chapters 4 and 5. Lastly, we introduce a new application of our framework: Generating superpixels in Chapter 6.

1.7 List of core publications, one-page summaries, and approvals

The remainder of this chapter provides one-page summaries of all chapters / (core) publications to fulfill the formal requirements of a cumulative dissertation. For the two core publications, it provides usage approvals from the publishers that can be found right after the respective one-page summaries. The two core publications “Turning Grain Maps into Diagrams” [Alp+23b] and “Dynamic grain models via fast heuristics for diagram representations” [Alp+23a] are included in Chapters 4 and 5 as text, formatted as previous chapters, and in Appendix A as their published journal versions.

Summary of Chapter 2

The text is available here [FG22a] and is currently being prepared for publication. It results from joint work with Peter Gritzmann. I was significantly contributing to all parts involved in creating this work.

This chapter lays the foundations for the following sections. We design so-called coresets, which are small data sets that carry sufficient information to approximate the optimization problem on the full data set. Our main contributions are:

The first coreset for anisotropic objective function and weight constraints.

We propose a new generalized coreset definition that also works for non-Euclidean anisotropic objective functions. Our definition enforces a direct mapping between the data set and its coreset. As even the assignment problem is notoriously difficult in the constrained case, such a mapping allows quick conversion of solutions between both data sets.

Based on this definition, we design the first coreset for the least-squares clustering problem with anisotropic objective function and weight constraints. Its size, i.e., the number of points in the coreset, is $\mathcal{O}(k^2 \epsilon^{-d-1})$.

While we base our technique on those in [HPK07], we significantly simplify the one-dimensional coreset construction. As a result, our construction uses half as many points as the previously existing one. Surely, this is only a constant factor reduction which is, however, highly relevant in practice as the number of variables in the resulting linear program depends on the number of points. Additionally, we reduce the dependence on k from cubic to squared. As a corollary, we also obtain a new coreset for fair clustering.

A counterexample showing that sensitivity sampling cannot be directly applied to the point set in the constrained case.

In the literature, many smaller coresets have been designed using a technique called sensitivity sampling. Unfortunately, this concept cannot be directly applied to the constrained case: We show that in the constrained case the sensitivity can become arbitrarily large in the number of points. As such, we outline a gap between constrained and unconstrained clustering that requires further research to close and emphasizes the importance of our coreset result.

Summary of Chapter 3

The text is available here [\[FGK22\]](#) and is currently being prepared for publication. It results from joint work with Peter Gritzmann and Fabian Klemm. I was significantly contributing to all parts involved in creating this work.

This text builds on results from [\[FG22a\]](#) but is motivated more strongly by our application of grain scan analysis. While the coresets in Chapter 2 are more general, the underlying topology of the coreset could be considered artificial in the context of grain scans as rays are not the outcome of scanning grains.

Grain scans are usually scanned as three-dimensional images at a certain resolution with the effort increasing as the required resolution increases. As representation quality has become more important, more and more effort has been put into scanning the material. Here, we prove that even low-resolution scans lead to a high-quality representation and that increasing the resolution beyond a certain point leads only to marginally better results when representing them using our clustering approach.

We show that reducing the resolution of a grain scan results in a coreset for the full resolution image. We apply our 1-dimensional coreset results to the natural lines we find in grid structures. Iterating this construction in each dimension leads to a d -dimensional coreset.

Theoretically, these grid coresets are of size $\mathcal{O}((k\epsilon^{-2/3})^d)$, which grows faster in the dimension d than our general coreset construction of size $\mathcal{O}(k^2\epsilon^{-d-1})$ in Chapter 2. However, in the relevant application parameters, for example, $d = 2$ or $d = 3$ and ϵ around 0.001, the coreset size is even smaller than our more general coreset construction. Consequently, we can scan materials upfront in a lower resolution, reducing the data collection effort significantly. We also show the effectiveness of this approach in [\[Alp+23b\]](#)/Chapter 4.

Summary of Chapter 4: Turning grain maps into diagrams (core publication)

The summarized publication has been published in [Alp+23b]. It results from joint work with Peter Gritzmam, Andreas Alpers, and Fabian Klemm. I was significantly contributing to all parts involved in creating this work.

The publication is motivated by the need to understand materials' characteristics to predict their behavior in an application. Specifically, we want to represent polycrystalline materials in a low-dimensional representation for improved storing, analysis, and modeling. We study methods for obtaining such a low-dimensional presentation using diagrams. The three main results can be summarized as follows.

Designing a new algorithm converting grain maps into diagrams with significantly decreased running times. We leverage our developed coresets techniques to approximate a clustering problem required in a method by [Lyc+11] to convert grain maps into diagrams. Our coresets procedures approximate the optimization problem well, as shown in Chapters 2 and 3, and compress the grain map data for significantly improved running times. We introduce additional problem-specific techniques sparsifying the grain map data further.

Developing a new optimization model optimizing all diagram parameters. Our compressing methods allow the use of more complex models in practice that optimize all diagram parameters to improve the fit of the diagrams to the grain maps. We develop and rationalize a model that optimizes all diagram parameters to address the underlying more directly than in the method used by [Lyc+11]. Combined with our compression technique, we achieve good running times.

Verifying and comparing our techniques on a real-world data set. We address the performance of our algorithms and compare them with the heuristics established so far in the literature. We also address how far we can compress a grain map. Our evaluation shows that we can compress a real-world grain map from 70 mn parameters to below just 100k data points. We conclude that in most cases, the established heuristics worked very well, being now able to compare its performance to multiple optimization techniques.

Summary of Chapter 5: Dynamic grain models via fast heuristics for diagram representations (core publication)

The summarized publication has been published in [Alp+23a]. It results from joint work with Peter Gritzmann, Andreas Alpers, and Fabian Klemm. I was significantly contributing to all parts involved in creating this work.

The publication is motivated by understanding grain growth. Using various modeling techniques, material scientists have studied material growth to identify their characteristics. We introduce a method that turns measured grain maps at discrete time steps into a continuous growth model of the grains in a material. We base this technique on the inter- and extrapolation of anisotropic power diagrams that we compute from the discrete-time grain maps.

The three main results can be summarized as follows.

Developing a new heuristic for generating anisotropic power diagrams. We analyze a heuristic in [Lyc+11] by deriving it via discriminant analysis for a certain prior. This sheds some additional light on which assumptions the heuristic performs well. At the same time, we propose new and simpler heuristics that perform well with different assumptions we have experienced in practice. We conclude with a comparison between both heuristics.

Developing a new dynamic grain growth model. We develop a dynamic growth model that inter- and extrapolates to obtain a continuous model. Our approach assumes that grain maps measured at discrete time steps during the growth process of a material are given. The studied heuristic converts the grain maps at each time step into an anisotropic power diagram. As the power diagram can be represented using a low-dimensional parameter space, we can use classical inter- and extrapolation techniques to extend the discrete diagram representation to a continuous one. We evaluate multiple such methods with varying degrees of freedom and study the fit of our approach at time steps that have been excluded in model-fitting.

Concluding that the grain growth process seems to favor a simple type of diagrams. We state and evaluate the hypothesis that the grain growth process converges towards a specific diagram type. We build a logistic regression model showing a relationship between grain disappearance and large-size parameters in the representation. This suggests that grains requiring a larger size parameter to be modeled are energetically non-ideal and that the growth converges towards a relatively simple diagram type.

Summary of Chapter 6:

The work summarized here is currently available in [FA20]. It results from joint work with Andreas Alpers. I was significantly contributing to all parts involved in creating this work.

We exploit the connection between clusterings and diagrams and introduce a new algorithm for generating superpixels called Power-SLIC. Generally, superpixels group multiple pixels of similar color so that the resulting superpixel boundaries reflect the overall image boundaries. This is a common pre-processing step in image analysis. Generating superpixels significantly reduces the number of objects a subsequent algorithm needs to process while adding additional contextual information.

Our main contribution is the design of a new superpixel algorithm based on anisotropic power diagrams that outperform existing diagram techniques in various criteria:

Boundary Precision. Power-SLIC outperforms other evaluated state-of-the-art algorithms in terms of boundary precision. As the diagram structure acts as regularization for the superpixel boundaries, Power-SLIC requires fewer boundary pixels in image areas not containing image boundaries.

Boundary Recall. Due to this regularization, Power-SLIC is the most robust against varying noise levels in terms of boundary recall. Varying noise scenarios are common, as the amount of noise captured depends on the environment's brightness.

Regularity. As Power-SLIC's superpixels are diagram cells, they are very compact and regular. Moreover, superpixel boundaries are polynomials of degrees at most two. Overall, Power-SLIC outperforms the other algorithm in terms of regularity.

Speed. Power-SLIC is based on SLIC. While the final steps significantly differ, the runtime of both algorithms is dominated by the common assignment step. Since SLIC is well-known for its speed and Power-SLIC's runtime is essentially identical, our algorithm is competitive in terms of its speed.

Resolution-Independence. The resolution independence of the diagram representation allows computing a superpixel segmentation on a low-resolution image instead of the high-resolution image. This compression enables faster computation, particularly on high-resolution images. Our coresets explain why this approach works particularly well with our algorithm.

Chapter 2

Coresets for Weight-Constrained Anisotropic Assignment and Clustering

This chapter constructs small coresets for weight-constrained anisotropic assignment and clustering with a specific view toward applications in materials science. Building on previous results for unconstrained least-squares clustering of Har-Peled & Kushal, we obtain coresets even for weight-constrained anisotropic clustering and, particularly, reduce the dependence of their sizes on the number of clusters from cubic to quadratic, an improvement which is decisive for applications in small dimensions such as grain mapping.

2.1 Introduction

This text constructs coresets for weight-constrained anisotropic least-squares assignment and clustering of n points in $\mathbb{R}^d$ into k clusters, for short, *WCA ASSIGNMENT* and *WCA CLUSTERING*, whose sizes are independent of n .

Focus: While we will derive our results in greater generality, our primary focus lies on the sizes of coresets for weight-constrained anisotropic least-squares assignment and clustering in small dimensions. This is motivated by our main application of grain mapping in materials science (briefly indicated in the next paragraph; see [Alp+15] for details) which requires the handling of 2D and 3D data sets (typically with $k \sim 500$, and $n \sim 10^9$, [Alp+23b].) Hence, our focus is on the influence of k (rather than d) to render the problem tractable in practice.

Clustering is a well-established tool for unsupervised learning and has been widely used for decades. More recent applications require additional conditions on the cluster sizes; see e.g., [Alp+15; BBG11; BBG14; BG04; BG12; BG20; BG21; BGK17; Chi+17; HJV19; SSS20], and [BDW09] for a general background on constrained clustering, and the handbook article [GK17] for a concise overview focusing on concepts and results of particular relevance in the present context. While computational issues are highly relevant in all such applications, the representation of grain maps, a problem in material science, particularly motivates the present chapter.

Grain mapping concerns structural representations of polycrystalline materials, which are only accessible through specific measurements. In fact, the available tomographic data allows us to determine the number k of grains, approximations of grain centroids, volumes, and measures of anisotropy. As demonstrated in [Alp+15], appropriate anisotropic diagrams that correspond to constrained anisotropic clusterings represent grain maps extremely well. In the introduced model, each of the n voxels is represented by k “indicator” variables, one for each grain. Hence, a reasonable 3D resolution generally leads to a prohibitively large optimization problem, particularly if grain dynamics requiring solutions for many time steps are studied. With this and related applications in mind, the present text constructs coresets for WCA ASSIGNMENT and WCA CLUSTERING, which are the bases of grain mapping for standard real-world instances in practice; see [Alp+23b].

Coresets can be seen as compressed data on which computations can be based without losing too much accuracy. Coresets have been developed for various problems [BLK17; FSS13; FSS20; Fic+13; HPK07; SW18], most notably for approximate unconstrained least-squares clustering. Constraints on the cluster sizes, as they appear naturally in the anisotropic weight-constraint grain mapping problems add, however, additional difficulties. For instance, if the clustering quality is measured with respect to k given *sites*, the unconstrained problem becomes trivial. The points are simply assigned to a nearest site. In contrast, for WCA ASSIGNMENT, we need to solve a linear program in $\mathcal{O}(nk)$ variables, which quickly becomes practically intractable for the large data sets in real-world applications. Hence, one aims at computing (near) optimal clusterings on small coresets and extending them (quickly) to clusterings on the full data set in order to render the total computation time practical.

Main contribution: We introduce a general coreset definition which is adequate for WCA ASSIGNMENT and WCA CLUSTERING, and construct coresets which depend only quadratically on the number k of clusters. While we are mainly interested in applications in dimensions 2 and 3, we prove our results in general dimension and, in particular, obtain coresets of size $\mathcal{O}(k^2/\epsilon^{d+1})$ for the weight-constrained anisotropic least-squares assignment and clustering of n points in $\mathbb{R}^d$ into k clusters. Our main Theorem 2.2, generalizes and improves results of [HPK07]; it is based on a somewhat sparser construction, an improved analysis and a specific utilization of a certain duality between weight-constrained anisotropic clustering and anisotropic power diagrams and hence reduces the dependency on k even in the “classical” unconstrained least-squares case.

Let us point out that our techniques can, in principle, be combined with the singular value decomposition from [Coh+15] in order to remove the dependence on d . This comes, however, at the expense of the largely increased dependence on k , resulting in coreset sizes in 2D and 3D which are far too large for practical applications in materials science where k is much larger than d . In view of our main application, we do not

particularly focus on the dependency on the dimension.

Our coresets have the favorable property that their WCA assignments and clusterings can be mapped very quickly to competitive WCA assignments and clusterings of the original data set without having to solve a “remaining” expensive linear program on the full data set. Thus, our coresets allow us to avoid large-scale computations that would often be virtually impossible for practical applications in materials science.

Relation to previous Work: Coresets for clustering with constraints have also been studied in [BFS24; HJV19; SSS20] in the context of fair clustering. As such, they provide the first extension of coreset constructions under additional constraints. While [HJV19] also extends [HPK07], we provide a simplified construction that results in the reduction of the dependence on k from cubic to quadratic. Also, our results are more general due to the anisotropic objective functions. In fact, the anisotropic distance measures cause a fundamental change in the geometry of the underlying diagrams that represent the clusterings. As a consequence, we need to deploy more sophisticated techniques.

As unconstrained least-squares clustering is a special case of WCA CLUSTERING, our worst-case results even improve on the previously best-known bounds on coreset sizes for unconstrained least-squares clustering with respect to k . In a probabilistic setting, and for unconstrained least-squares clustering, however, smaller coresets can be obtained with high probability via *importance sampling* [FL11]. As examples show, these techniques can, unfortunately, not be directly extended to weight-constrained clustering as the total sensitivity of a given instance may be as large as the size of the original data set. While a local indirect application of sensitivity is possible, [BFS24], it comes at the cost of a large dependence on k providing no advantage in our situation.

Organization: The present text is organized as follows. Section 2.2 introduces necessary notation, formally specifies our main problems, provides relevant background information, and states our main results. Section 2.3 proves first results, constructs coresets via neighborhood mergers, introduces other ingredients for the proof of our main theorem and explains the relation between WCA CLUSTERING and geometric diagrams. Section 2.4 provides the proof of Theorem 2.2, and Section 2.5 concludes with some remarks and open problems.

The proofs of our main results are based on novel ideas, an improved analysis and the utilization of a strong correspondence between clustering and diagrams. In part, they also rely on generalizations of certain auxiliary estimates that are mostly technical. To highlight the conceptual work and new techniques required to extend to anisotropic diagrams, we moved purely technical improvements to the appendix (but included all arguments in detail as a service to the reader).

2.2 Notation, background, and main results

In the following, we formally introduce the basic problems and state our main results. While we provide the most relevant references, we also refer the reader to [GK17] and for additional pointers to literature.

Weight-constrained anisotropic assignment and clustering: Let $d, n \in \mathbb{N}$ and, for $j \in [n] = \{1, \dots, n\}$, let $x_j \in \mathbb{R}^d$ and $\omega_j \in (0, \infty)$. In the following, we will collect the data in two families $X = \{x_1, \dots, x_n\}$ and $\Omega = \{\omega_1, \dots, \omega_n\}$ and refer to ω_j as the *weight* of the *data point* x_j . We refrain from a formal description of the points and weights by means of functions on $[n]$ and tacitly assume that the correspondences are always indicated by the index, i.e. ω_j is the weight of x_j . With this understanding, the pair (X, Ω) is called *weighted data set*, and $\omega(X) = \sum_{j=1}^n \omega_j$ is its *total weight*.

Further, $k \in \mathbb{N}$ always specifies the *number of clusters* we want to compute. Then, a k -*clustering* C of (X, Ω) is a vector

$$C = (C_1, \dots, C_k) = (\xi_{11}, \dots, \xi_{1n}, \dots, \xi_{k1}, \dots, \xi_{kn}),$$

where the component ξ_{ij} specifies the fraction of x_j that is assigned to the i th cluster $C_i = (\xi_{i1}, \dots, \xi_{in})$. More formally, the *assignment conditions* are

$$\xi_{ij} \geq 0 \quad \forall i \in [k], j \in [n] \quad \text{and} \quad \sum_{i=1}^k \xi_{ij} = 1 \quad \forall j \in [n].$$

If $C_i = 0$, the i th cluster is *void*. The *weight* $\omega(C_i)$ and, if C_i is not void, the *centroid* c_i of the cluster C_i are given by

$$\omega(C_i) = \sum_{j=1}^n \xi_{ij} \omega_j \quad \text{and} \quad c_i = c(C_i) = \frac{1}{\omega(C_i)} \sum_{j=1}^n \xi_{ij} \omega_j x_j,$$

respectively. If C_i is void, we set $c_i = 0$. Further, we use the abbreviation $c(C) = \{c_1, \dots, c_k\}$ for the family of centroids of the clustering C , and sometimes we write $c_i = c(C_i)$. The set of all k -clusterings of (X, Ω) will be denoted by $\mathcal{C}(k, X, \Omega)$. Whenever the ingredients are clear from the context, we will speak of the elements $C \in \mathcal{C}(k, X, \Omega)$ of a clustering of X or simply of a clustering.

As we are particularly interested in weight-constrained clustering, suppose we are further given lower and upper bounds $\kappa_i^-, \kappa_i^+ \in \mathbb{R}$ with $\kappa_i^- \leq \kappa_i^+$ for each cluster, again, collected in a family

$$K = \{\kappa_1^-, \kappa_1^+, \dots, \kappa_k^-, \kappa_k^+\}.$$

Then, we require that the clusters satisfy the *weight constraints*

$$\kappa_i^- \leq w(C_i) = \sum_{j=1}^n \omega_j \xi_{ij} \leq \kappa_i^+ \quad \forall i \in [k].$$

The set of all such *weight-constrained clusterings* will be denoted by $\mathcal{C}_K(k, X, \Omega)$. Note that the simple and easy-to-verify condition

$$\sum_{i=1}^k \kappa_i^- \leq \omega(X) \leq \sum_{i=1}^k \kappa_i^+$$

is equivalent to the existence of such clusterings, i.e., to $\mathcal{C}_K(k, X, \Omega) \neq \emptyset$. To avoid trivialities, we will always assume that this condition holds.

Of course, if $\kappa_i^- = 0$, and $\kappa_i^+ \geq \omega(X)$ for an $i \in [k]$ the i th weight constraint is redundant and can be omitted. We will signify this situation by writing $\kappa_i^+ = \infty$, i.e., formally allowing $\kappa_i^+ \in \mathbb{R} \cup \{\infty\}$. Hence, the choice of

$$K_\infty = \{0, \infty, \dots, 0, \infty\}$$

refers to the unconstrained case.

Next, we introduce measures for the quality of a given clustering. They are based on distances that may utilize a different norm for each cluster. So, for $i \in [k]$, let $A_i \in \mathbb{R}^{d \times d}$ be a positive definite symmetric matrix, and let $\|\cdot\|_{A_i}$ denote the associated *ellipsoidal norm*, i.e. $\|x\|_{A_i} = x^T A_i x$ for any $x \in \mathbb{R}^d$. Again, we collect all k such matrices in the family $\mathcal{A} = \{A_1, \dots, A_k\}$. Clearly, if $E = E_d$ denotes the $d \times d$ unit matrix, $\mathcal{E} = \{E, \dots, E\}$ refers to the classic Euclidean *least-squares* case, in which each cluster norm is $\|x\|_2 = \|x\|_E$. Of course, if $\lambda^-(A_i)$ and $\lambda^+(A_i)$ denote the smallest and largest eigenvalue of A_i , then

$$\lambda^-(A_i) \cdot \|x\|_2^2 \leq \|x\|_{A_i}^2 \leq \lambda^+(A_i) \cdot \|x\|_2^2.$$

In grain mapping, the different matrices reflect measured knowledge about the moments of the grains in materials, see [Alp+15], and enable the representation of anisotropic growth.

We can now measure the quality of a weight-constrained clustering $C \in \mathcal{C}_K(k, X, \Omega)$ with respect to a set of reference points $S = \{s_1, \dots, s_k\} \subset \mathbb{R}^d$, called *sites*, in terms of

$$\text{cost}_{\mathcal{A},K}(X, \Omega, C, S) = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{A_i}^2.$$

Note that $\text{cost}_{\mathcal{A},K}(X, \Omega, C, S)$ depends on K only indirectly, as $C \in \mathcal{C}_K(k, X, \Omega)$ is given. We use the subscript K consistently in order to indicate that we are considering the constrained case.

Then, *WCA ASSIGNMENT* is the following problem:

Given $k, X, \Omega, \mathcal{A}, K, S$, find $C \in \mathcal{C}_K(k, X, \Omega)$ that minimizes $\text{cost}_{\mathcal{A},K}(X, \Omega, C, S)$.

In this setting, the optimization involves only the nk variables ξ_{ij} . Hence, in effect, WCA ASSIGNMENT is a linear program that can be solved in polynomial time. By our general assumption, $\mathcal{C}_K(k, X, \Omega) \neq \emptyset$, a finite minimum exists, which we denote by $\text{cost}_{\mathcal{A},K}(X, \Omega, S)$, i.e.,

$$\text{cost}_{\mathcal{A},K}(X, \Omega, S) = \min_{C \in \mathcal{C}_K(k, X, \Omega)} \text{cost}_{\mathcal{A},K}(X, \Omega, C, S).$$

Instead of fixing the sites and optimizing over the clusterings, we can also fix the clustering, optimize over the sites, and set

$$\text{cost}_{\mathcal{A},K}(X, \Omega, C) = \min_S \text{cost}_{\mathcal{A},K}(X, \Omega, C, S).$$

In Lemma 2.5, we will see that the optimal sites are, indeed, the centroids of the clusters.

Optimizing the latter term over all clusterings results in an optimal weight-constrained anisotropic clustering; its objective function value will be denoted by

$$\text{OPT}_{\mathcal{A},K}(X) = \min_{C \in \mathcal{C}_K(k, X, \Omega)} \text{cost}_{\mathcal{A},K}(X, \Omega, C).$$

The problem of finding such optimal clustering is called *WCA CLUSTERING* and is defined as:

Given $k, X, \Omega, \mathcal{A}, K$, find $C^* \in \mathcal{C}_K(k, X, \Omega)$ that minimizes $\text{cost}_{\mathcal{A},K}(X, \Omega, C)$.

As the centroids depend on the clusterings, the WCA CLUSTERING objective function is highly non-linear in its variables ξ_{ij} . It is this property that makes the problem NP-hard and already renders unconstrained least-squares clustering difficult, [Alo+09; Das07]. The latter is shown to be NP-hard even in dimension $d = 2$, [MNV12]. Moreover, [Awa+15] shows that no PTAS exists in both, the number of clusters k , and the dimension of the data d .

These hardness results apply to WCA CLUSTERING, as the unconstrained least-squares clustering problem is contained as the special case $\mathcal{A} = \mathcal{E}$ and $K = K_\infty$. In this classic situation, we drop K and $\mathcal{A}$ from the above cost terms and simply write $\mathcal{C}(k, X, \Omega) = \mathcal{C}_{K_\infty}(k, X, \Omega)$ and

$$\begin{aligned} \text{cost}(X, \Omega, C, S) &= \text{cost}_{\mathcal{E},K_\infty}(X, \Omega, C, S), & \text{cost}(X, \Omega, S) &= \text{cost}_{\mathcal{E},K_\infty}(X, \Omega, S), \\ \text{cost}(X, \Omega, C) &= \text{cost}_{\mathcal{E},K_\infty}(X, \Omega, C), & \text{OPT}(X, \Omega) &= \text{OPT}_{\mathcal{E},K_\infty}(X, \Omega). \end{aligned}$$

If we only drop K but keep $\mathcal{A}$, we are in the anisotropic but unconstrained case, and if we drop $\mathcal{A}$ but keep K we are dealing with the constrained Euclidean case and write, for instance:

$$\begin{aligned} \text{cost}_{\mathcal{A}}(X, \Omega, C, S) &= \text{cost}_{\mathcal{A},K_\infty}(X, \Omega, C, S), & \text{OPT}_{\mathcal{A}}(X, \Omega) &= \text{OPT}_{\mathcal{A},K_\infty}(X, \Omega) \\ \text{cost}_K(X, \Omega, C, S) &= \text{cost}_{\mathcal{E},K}(X, \Omega, C, S), & \text{OPT}_K(X, \Omega) &= \text{OPT}_{\mathcal{E},K}(X, \Omega). \end{aligned}$$

Coresets: A popular method of handling the high computational complexity of unconstrained least-squares clustering is to approximate the data set by much smaller sets that still capture the relevant properties of the data, [Bra+16; Che09; Coh+15; FL11; HPK07; MMR19]. The decisive property of such a *coreset* is that the cost of a clustering of its points is comparable to the cost of clusterings of the original data set. Hence, time-consuming computations can be performed on much smaller data sets and subsequently extended to good approximations of optimal solutions for the original massive data.

We will now define coresets more formally, in a manner that is suitable for the weight-constrained anisotropic case.

Definition 2.1

Let $k \in \mathbb{N}$, let (X, Ω) and $(\tilde{X}, \tilde{\Omega})$ be two weighted data sets, and $\epsilon \in (0, 1/2]$, $\delta \in [1, \infty)$.

Then $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -coreset for $(k, X, \Omega, \mathcal{A})$ if for all K there exists a mapping $f : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, X, \Omega)$, called an *extension*, and real constants Δ^+, Δ^- , referred to as Δ -terms or *offsets*, with $0 \leq \Delta^+ \leq \delta \cdot \Delta^-$, such that the following two conditions hold for all sets S of k sites, and clusterings $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$:

- (a) $(1 - \epsilon) \text{cost}_{\mathcal{A}, K}(X, \Omega, f(\tilde{C}), S) \leq \text{cost}_{\mathcal{A}, K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta^+,$
- (b) $\text{cost}_{\mathcal{A}, K}(\tilde{X}, \tilde{\Omega}, S) + \Delta^- \leq (1 + \epsilon) \text{cost}_{\mathcal{A}, K}(X, \Omega, S).$

If the context is clear, we refer to $(\tilde{X}, \tilde{\Omega})$ as an (ϵ, δ) -coreset. If $\delta = 1$, we speak of an ϵ -coreset. If $\Delta^+ = \Delta^- = 0$, we set $\delta = 1$ and speak of a 0-offset ϵ -coreset.

Note that the conditions in Definition 2.1 are required for *every set* S of sites and *each* possible K . Hence, $(\tilde{X}, \tilde{\Omega})$ can also be regarded as a coreset for *each instance* $(k, X, \Omega, \mathcal{A}, K)$ of WCA CLUSTERING and *each instance* $(k, X, \Omega, \mathcal{A}, K, S)$ of WCA ASSIGNMENT, and we will sometimes refer to it this way. Let us mention in passing, that hence our coresets are *composable*, i.e., coresets for subsets of the data can be combined to result in a single coreset for the full data set.

For easy distinction, we will universally signify coresets by means of a tilde or (if additionally needed) a bar, as in $(\tilde{X}, \tilde{\Omega})$ or $(\bar{X}, \bar{\Omega})$. Parameters or objects on the different sets will also be marked this way. For instances, clusterings on (X, Ω) will usually be named C and their components ξ_{ij} , while clusterings and their components on coresets $(\tilde{X}, \tilde{\Omega})$ are signified by a tilde, i.e., they are called $\tilde{C}$ and $\tilde{\xi}_{ij}$.

Before we state our main result, let us briefly comment on the rationale behind the above coreset definition. First, note that $\text{cost}_{\mathcal{A}, K}(X, \Omega, S) \leq \text{cost}_{\mathcal{A}, K}(X, \Omega, f(\tilde{C}), S)$. Hence, in the special situation that $\tilde{C}$ attains $\text{cost}(\tilde{X}, \tilde{\Omega}, S)$, $\Delta^+ = \Delta^- = 0$, and $\delta = 1$, the above condition for a 0-offset coreset implies that

$$(1 - \epsilon) \text{cost}_{\mathcal{A}, K}(X, \Omega, S) \leq \text{cost}_{\mathcal{A}, K}(\tilde{X}, \tilde{\Omega}, S) \leq (1 + \epsilon) \text{cost}_{\mathcal{A}, K}(X, \Omega, S),$$

which coincides with the coreset definition of [HPK07] for standard unconstrained least-squares clustering, i.e., $\mathcal{A} = \mathcal{E}$, $K = K_\infty$. Further, note that the two Δ -terms generalize the one additive offset parameter in [FSS13; FSS20]. They will be used to show that (ϵ, δ) -coresets preserve δ -approximations up a factor ϵ . Finally, and most importantly, let us point out that, while rather weak, Definition 2.1 still captures the main motivation for the concept that an approximate solution of the constrained clustering problem on a coreset yields an approximate solution on the original data set; see Theorem 2.7.

As the general goal is to improve the tractability of the underlying optimization problem we are aiming at coresets $(\tilde{X}, \tilde{\Omega})$ of weighted sets (X, Ω) that are small enough to enhance the performance of available algorithms significantly but still preserve enough of the structure of the original instance to yield solutions which can be extended to good approximations for the original instance. In the unconstrained least-squares case, clusterings of the coreset can easily be converted into clusterings of the original data set. In fact, it suffices to assign the points of X to one of the closest determined sites, and this is efficiently facilitated by means of Voronoi diagrams. However, such a procedure does not respect the weight constraints and, hence, does not produce WCA clusterings of the full data set in general.

In Definition 2.1, this conversion of a *coreset clustering* $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ into a clustering $C \in \mathcal{C}_K(k, X, \Omega)$ of the full data set is facilitated by the explicitly introduced extension

$$f : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, X, \Omega).$$

In fact, f captures any way of “extending” $\tilde{C}$ to a feasible clustering C of the full data set. Specifically, condition (a) of Definition 2.1 can be interpreted as saying that for each weight-constrained coreset clustering $\tilde{C}$, we obtain a weight-constrained clustering $f(\tilde{C})$ on the full data set of cost that is not much worse. Condition (b) expresses the property that an optimal WCA ASSIGNMENT for sites S on the full data set is not much better than one on the coreset. Hence, good approximations of the latter lead to good approximations of the former.

Main result: Let us now state the main result of this chapter.

Theorem 2.2

For any instance $(k, X, \Omega, \mathcal{A}, K)$ of WCA CLUSTERING and for any $\epsilon \in (0, 1/2]$ and

$$\delta \geq \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})},$$

there is an (ϵ, δ) -coreset of size $\tilde{n}$ with

$$\tilde{n} \in \mathcal{O} \left(\frac{k^2}{\epsilon^{d+1}} \right).$$

The proof of Theorem 2.2 follows the basic construction of [HPK07] but generalizes it to weight-constrained anisotropic clustering and, by suitably exploiting the underlying geometric structure, even reduces the asymptotics by a factor of k . Our improvement over [HPK07] is a result of two elements. First, we provide a conceptually simpler coresset construction on the line that only uses half the points by exploiting the properties of the center of gravity for anisotropic least-squares clustering. Second, our analysis benefits from bounding the multiple errors of the coresset only once instead of summing-up the individual errors.

Note that, the uniform “condition number bound” $\delta \geq \lambda^+(\mathcal{A})/\lambda^-(\mathcal{A})$ becomes 1 for $\mathcal{A} = \mathcal{E}$. Hence, we obtain the following corollary for unconstrained least-squares clustering.

Corollary 2.3

There is an ϵ -coreset of size $\mathcal{O}\left(\frac{k^2}{\epsilon^{d+1}}\right)$ for least-squares clustering.

2.3 Estimates, approximation, neighborhood mergers, and other ingredients

Estimates: Let (X_0, Ω_0) be a weighted data set, let c_0 be its centroid, and let $A \in \mathbb{R}^{d \times d}$ be a positive definite symmetric matrix. (The index 0 is used to indicate that we are not necessarily dealing with the full data set but with clusters or batches of points.) Further, let ω_x denote the weight of a point $x \in X_0$, and define the cluster *variation* or batch *error* $\mathcal{V}_A(X_0)$ of X_0 for A by

$$\mathcal{V}_A(X_0) = \sum_{x \in X_0} \omega_x \|x - c_0\|_A^2.$$

Note that $\mathcal{V}_A(X_0)$ is bounded by $\mathcal{V}_E(X_0)$ as follows.

Remark 2.4

$$\lambda_{\min}(A)\mathcal{V}_E(X_0) \leq \mathcal{V}_A(X_0) \leq \lambda_{\max}(A)\mathcal{V}_E(X_0).$$

In accordance with our previous notation, $\omega(X_0) = \sum_{x \in X_0} \omega_x$ denotes the weight of X_0 . The following elementary lemma is a slight generalization of well-known facts. The proof uses the notion of the *support* $\text{supp}(C_i)$ of the cluster C_i , defined by

$$\text{supp}(C_i) = \{x_j : j \in [n] \wedge \xi_{ij} > 0\}.$$

Lemma 2.5

(a) *Let (X_0, Ω_0) be a weighted data set, c_0 its centroid, $A \in \mathbb{R}^{d \times d}$ a positive definite symmetric matrix, and $s \in \mathbb{R}^d$. Then,*

$$\sum_{x \in X_0} \omega_x \|x - s\|_A^2 = \omega(X_0) \cdot \|c_0 - s\|_A^2 + \mathcal{V}_A(X_0).$$

- (b) Let $(k, X, \Omega, \mathcal{A}, K, S)$ be an instance of WCA ASSIGNMENT and let $C = (C_1, \dots, C_k)$ be a clustering. Then,

$$\text{cost}_{\mathcal{A},K}(X, \Omega, C) \leq \text{cost}_{\mathcal{A},K}(X, \Omega, C, S)$$

with equality if and only if $s_i = c(C_i)$ for every non void cluster C_i .

Proof. See Section 2.6.1 □

The next result shows that the conditions of Definition 2.1 imply the corresponding inequalities involving the centroids.

Lemma 2.6

Let $(\tilde{X}, \tilde{\Omega})$ be an (ϵ, δ) -coreset for $(k, X, \Omega, \mathcal{A}, K)$. Then,

- (a') $(1 - \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C})) \leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}) + \Delta^+,$
 (b') $\text{OPT}_{\mathcal{A},K}(\tilde{X}) + \Delta^- \leq (1 + \epsilon) \text{OPT}_{\mathcal{A},K}(X).$

Proof. See Section 2.6.2. □

The following result shows that our definition “lives up to its expectation” that good clusterings on coresets yield good clusterings on the full data set. As usual, the quality of approximation will be measured in terms of the relative error. Hence, we speak of a γ -approximation if the objective function value of the produced solution is bounded from above by γ times its minimum.

Theorem 2.7

Let $I = (k, X, \Omega, \mathcal{A}, K)$ be an instance of WCA CLUSTERING, $\epsilon \in (0, 1/2]$, $\delta \in [1, \infty)$, let $(\tilde{X}, \tilde{\Omega})$ be an $(\epsilon/3, \delta)$ -coreset for I , and let f be its extension.

Further, let $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$, and $\gamma \in \mathbb{R}$ with $\gamma \geq \delta$.

- (a) If $\tilde{C}$ is a γ -approximation for $\tilde{I}_S$, then $f(\tilde{C})$ is a $(1 + \epsilon)\gamma$ -approximation for I_S .
 (b) If $\tilde{C}$ is a γ -approximation for $\tilde{I}$, then $f(\tilde{C})$ is a $(1 + \epsilon)\gamma$ -approximation for I .

Proof. See Section 2.6.3. □

Approximation for unconstrained least-squares clustering: The following well-known result is an important ingredient as the initial step for our coreset constructions for WCA CLUSTERING. See [ADK09; Bra+16; FSS13; FSS20; HPK07; Kan+04] for a non-exhaustive list of corresponding algorithms.

Proposition 2.8

There exist constants $\alpha \geq 1$, $\beta \in \mathbb{N}$ and a polynomial-time algorithm with the following property: Given an instance (k, X, Ω) of unconstrained least-squares clustering, the algorithm computes a clustering with at most βk clusters with a cost $\text{ALG}_{(\alpha, \beta)}$ satisfying

$$\text{ALG}_{(\alpha, \beta)} \leq \alpha \text{OPT}(X).$$

Algorithms according to Proposition 2.8 are usually called (α, β) -approximations, as they compute an α -approximation but use $k\beta$ rather than k clusters. While, in terms of their approximation properties, smaller constants α and β are preferable, the running times typically increase significantly with decreasing constants. Hence, in practice, these effects require some fine-tuning. As pointed out by [ADK09], the parameter β does not affect the coreset's ability to approximate unconstrained least-squares k -clusterings.

Coresets based on local neighborhood mergers: A well-known standard method for constructing coresets for unconstrained least-squares clustering is to merge neighboring data points into a single point that carries their total weight. The merging can be represented by a function that assigns each point of the original data set (X, Ω) to a point of the new set $(\tilde{X}, \tilde{\Omega})$. More formally, with $n = |X|$ and $\tilde{n} = |\tilde{X}|$ as before, let

$$p : [n] \rightarrow [\tilde{n}]$$

be surjective such that

$$\tilde{\omega}_{\tilde{j}} = \sum_{j \in p^{-1}(\tilde{j})} \omega_j \quad \forall \tilde{j} \in [\tilde{n}].$$

Then, we call p a *merging function*.

In order to bound the cost of optimal solutions for a given WCA CLUSTERING instance, recall first that $\lambda^+(\mathcal{A})$ and $\lambda^-(\mathcal{A})$ denote the largest and smallest eigenvalue of all matrices in $\mathcal{A}$, respectively, and note that the optima in the Euclidean and anisotropic unconstrained situation are related via

$$\text{OPT}(X) \leq \frac{1}{\lambda^-(\mathcal{A})} \text{OPT}_{\mathcal{A}}(X) \leq \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})} \text{OPT}(X).$$

This shows, in particular, that, for (α, β) -approximations of least-squares clustering we have the estimate

$$\text{ALG}_{(\alpha, \beta)} \leq \frac{\alpha}{\lambda^-(\mathcal{A})} \text{OPT}_{\mathcal{A}}(X) \leq \frac{\alpha \lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})} \text{OPT}(X)$$

for the corresponding anisotropic problem. Also, the former observation implies the following useful consequence.

Lemma 2.9

Let $\eta \in (0, \infty)$, and suppose that

$$\sum_{j=1}^n \omega_j \|\tilde{x}_{p(j)} - x_j\|_2^2 \leq \frac{\eta \cdot \lambda^-(\mathcal{A})}{\lambda^+(\mathcal{A})} \text{OPT}(X).$$

Then, whenever $\sum_{i=1}^k \xi_{ij} = 1$ for each $j \in [n]$,

$$\sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|\tilde{x}_{p(j)} - x_j\|_{A_i}^2 \leq \eta \cdot \text{OPT}_{\mathcal{A}}(X).$$

Proof. See Lemma 2.9 □

The next technical result, which we will also need later, considers the effect of merging data points on the cost of clustering. So, let p be a merging function for (X, Ω) , let $(\tilde{X}, \tilde{\Omega})$ be the obtained data set and let $C \in \mathcal{C}_K(k, X, \Omega)$. Then, p gives rise to a clustering $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ defined by

$$\tilde{\xi}_{i\tilde{j}} = \frac{1}{\tilde{\omega}_{\tilde{j}}} \sum_{j \in p^{-1}(\tilde{j})} \xi_{ij} \omega_j \quad \forall \tilde{j} \in [\tilde{n}], i \in [k].$$

Slightly abusing notation, we will write $p(X) = \tilde{X}$, $p(X, \Omega) = (\tilde{X}, \tilde{\Omega})$ and $p(C) = \tilde{C}$. Then, we obtain the following inequalities.

Lemma 2.10

Let p be a merging function for (X, Ω) , $C \in \mathcal{C}_K(k, X, \Omega)$ and set

$$D = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|\tilde{x}_{p(j)} - x_j\|_{A_i}^2.$$

Then,

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X, \Omega, C, S) &- 2\sqrt{D \cdot \text{cost}_{\mathcal{A},K}(X, \Omega, C, S)} \\ &\leq \text{cost}_{\mathcal{A},K}(p(X, \Omega), p(C), S) = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|\tilde{x}_{p(j)} - s_i\|_{A_i}^2 \\ &\leq \text{cost}_{\mathcal{A},K}(X, \Omega, C, S) + 2\sqrt{D \cdot \text{cost}_{\mathcal{A},K}(X, \Omega, C, S)} + D. \end{aligned}$$

Proof. See Section 2.6.5. □

With the aid of these lemmas and some additional arguments it can be shown that small changes in the position of the data points result in ϵ -coresets. This is a generalization of a result of [SSS20] to the weighted and anisotropic case. The permitted deviation is quantified in terms of the cost $\text{OPT}(X)$ of an optimal unconstrained least-squares clustering on the data set (X, Ω) .

Theorem 2.11

Let $(k, X, \Omega, \mathcal{A}, K)$ be an instance of WCA CLUSTERING. Further, let $(\tilde{X}, \tilde{\Omega})$ be a weighted data set, $\tilde{n} = |\tilde{X}|$, $\epsilon \in (0, 1/2]$, and let $p : [n] \rightarrow [\tilde{n}]$ be a merging function. If

$$\sum_{j=1}^n \omega_j \|x_j - \tilde{x}_{p(j)}\|_2^2 \leq \frac{\epsilon^2 \lambda^-(\mathcal{A})}{16 \lambda^+(\mathcal{A})} \text{OPT}(X),$$

then $(\tilde{X}, \tilde{\Omega})$ is a 0-offset ϵ -coreset.

Proof. The proof is based on new but rather technical results and on tedious but straight forward generalizations of results, previously known only for the Euclidean case; it is given in Section 2.6.6. $\square$

Diagrams: A major ingredient for obtaining the asserted coreset sizes is the strong relation between clustering and diagrams. We will describe it now in a concise manner, which is strongly focused towards its use in Section 2.4. For a detailed general account, additional pointers to the literature, and further background information, see [BGK17; GK17].

An *anisotropic diagram* $\mathcal{P} = \mathcal{P}(T, \Sigma, \mathcal{A})$ is specified by a set $T = \{t_1, \dots, t_k\} \subset \mathbb{R}^d$ of k different sites, corresponding *sizes* $\Sigma = \{\sigma_1, \dots, \sigma_k\} \in \mathbb{R}^k$, and a set of positive definite symmetric matrices $\mathcal{A} = \{A_1, \dots, A_k\}$. It is then defined as the collection $\mathcal{P} = \{P_1, \dots, P_k\}$ of subsets P_i of $\mathbb{R}^n$, called *cells*, given by

$$P_i = \{x \in \mathbb{R}^d : \forall \ell \in [k] : \|x - t_i\|_{A_i}^2 + \sigma_i \leq \|x - t_\ell\|_{A_\ell}^2 + \sigma_\ell\}.$$

Since the ellipsoidal norms $\|\cdot\|_{A_i}$ are strictly convex, cells of different sites do not have any interior points in common; see [BGK17]. Note that, in the special case of $\Sigma = \{0, \dots, 0\}$ and $\mathcal{A} = \mathcal{E}$, we obtain the classic *Voronoi diagram* of the points in T .

Now, let $C = (C_1, \dots, C_k) \in \mathcal{C}_K(k, X, \Omega)$. Then, we say that $\mathcal{P}$ and C are *compatible* if $\text{supp}(C_i) \subset P_i \cap X$ for all $i \in [k]$. More strongly, if $\text{supp}(C_i) = P_i \cap X$, the diagram $\mathcal{P}$ and the clustering C are called *strongly compatible*. Note that two cells of different sites can only have boundary points in common anyway. As we will see later, the bound on the coreset quality relies on the fact that for relevant clusterings, the number of points that are fractionally assigned to different clusters can be further controlled. More precisely, C and $\mathcal{P}$ are called *strictly compatible* if $\mathcal{P}$ and C are strongly compatible and, for each $i, \ell \in [k]$ with $i \neq \ell$ the intersection $(P_i \cap P_\ell) \cap X$ contains at most one point. Later, we will use the following result, which has been shown for the Euclidean case in [BG12, Corollary 2.2] and which follows from [BGK17] in the general case:

Proposition 2.12

Given an instance $(k, X, \Omega, \mathcal{A}, K, S)$ of WCA ASSIGNMENT, there is a clustering C that attains $\text{cost}_{\mathcal{A}, K}(X, \Omega, S)$ and admits a strictly compatible anisotropic diagram. Further, such a diagram can be computed in polynomial time.

2.4 Small coresets

We will now construct small (ϵ, δ) -coresets, analyze their properties and prove Theorem 2.2. In Section 2.4.1, we will first provide the construction and bound the sizes of the resulting coresets $(\tilde{X}, \tilde{\Omega})$. In Section 2.4.2, we will derive a structural property for clusterings on $(\tilde{X}, \tilde{\Omega})$ that admit a strictly compatible anisotropic diagram. Finally, the coreset properties according to Definition 2.1 will be verified in Section 2.4.3.

2.4.1 The construction

The construction of (ϵ, δ) -coresets for WCA CLUSTERING follows the basic scheme of [HPK07] for producing coresets for unconstrained least-squares clustering. It can be thought of as a two-ingredient recipe. First, points are projected onto a set of lines to generate a favorable structure. Second, points on each line are grouped into so-called batches.

The generalization of the first ingredient is technical, and we have moved the proofs to the appendix. The generalization of the second ingredient, however, requires more advanced arguments. Hence, we focus on the new construction of batch points as this allows to reduce the dependence on k from cubic to quadratic. As we will see later the structural properties of diagrams render it possible to derive the improved bounds even in the anisotropic case.

First ingredient: The initial step of the first ingredient is to compute an (α, β) -approximation for least-squares clustering using Proposition 2.8. This approximation results in an integer assignment of the points of X to $\hat{k}$ clusters $\hat{C}_i$ with $\hat{k} \leq \beta \cdot k$ of cost

$$\text{ALG}_{(\alpha, \beta)}(X) \leq \alpha \cdot \text{OPT}(X).$$

For $i \in [\hat{k}]$, let $\hat{c}_i$ denote the centroid of $\hat{C}_i$. (The following exposition is written with the general situation in mind that none of the clusters is void. Note, however, that void clusters do not cause any problems but actually reduce the size of the constructed coreset.)

Next, the obtained cluster centroids $\hat{c}_i$ are used as *vertices*, from which suitable pencils of lines issue. In order to construct such pencils, ϵ -nets are employed, i.e., point sets Q with properties explained in the following lemma.

Proposition 2.13 (see [HPK07; Mat02])

Let, as usual, $\mathbb{S}^{d-1}$ denote the Euclidean unit sphere of $\mathbb{R}^d$. Given $\epsilon_0 \in (0, 1/2)$, there exists a point set $Q = Q(\epsilon_0) \subset \mathbb{Q}^d \setminus \{0\}$ with the following properties:

- (a) Q contains $\mathcal{O}(\epsilon_0^{-(d-1)})$ points.
- (b) For each $p \in \mathbb{S}^{d-1}$ there exists a point $q \in Q$ such that $\|p - q\|_2 \leq \epsilon_0$, and

(c) Q can be computed in time $\mathcal{O}(\epsilon_0^{-(d-1)})$.

Using Proposition 2.13, we compute such a set $Q(\epsilon_0)$ for

$$\epsilon_0 = \frac{\epsilon}{4} \sqrt{\frac{\lambda^-(\mathcal{A})}{\alpha \cdot \lambda^+(\mathcal{A})}}.$$

This leads to $\hat{k}$ pencils $\hat{c}_i + \mathbb{R}Q$ of the lines

$$L(\hat{c}_i, q) = \hat{c}_i + \mathbb{R}q, \quad \forall i \in [\hat{k}], q \in Q(\epsilon_0).$$

Note that, as $0 \notin Q$, all sets $L(\hat{c}_i, q)$ are indeed lines. Let $\mathcal{L}$ denote the set of all such lines. Then,

$$|\mathcal{L}| = \hat{k} \cdot |Q| \in \mathcal{O}(k\epsilon^{-(d-1)}).$$

Next, for each $i \in [\hat{k}]$, we project each point of $x \in X \cap \hat{C}_i$ orthogonally on a closest ray $L(\hat{c}_i, q)$; ties are resolved arbitrarily. The resulting point will be denoted by $\bar{x}$. Note that, unless Q is chosen in an appropriately general position, it may happen that two points of X are projected onto the same point $\bar{x}$. We do not merge such points, and we treat them as separate entities in the family $\bar{X}$. The weights then remain unchanged, and we obtain the weighted data set $(\bar{X}, \Omega)$ of the same cardinality n , i.e., the merging function is the identity. As the following lemma shows, $(\bar{X}, \Omega)$ is a 0-offset ϵ -coreset for WCA CLUSTERING, with the extension $f_1 : \mathcal{C}_K(k, \bar{X}, \Omega) \rightarrow \mathcal{C}_K(k, X, \Omega)$ defined by

$$\xi_{ij} = \bar{\xi}_{ij} \quad \forall j \in [n], i \in [\hat{k}].$$

Lemma 2.14

$(\bar{X}, \Omega)$ is a 0-offset ϵ -coreset for WCA CLUSTERING.

Proof. See Section 2.6.7 □

In conjunction with Lemma 2.14, the next lemma shows that it indeed suffices to design coresets that live on the union of the constructed pencils, i.e. on $\mathcal{L}$.

Lemma 2.15

Let $\epsilon_1, \epsilon_2 \in (0, 1/2]$, $\delta_1, \delta_2 \in [1, \infty)$, and set

$$\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2 \quad \text{and} \quad \delta = \max\{\delta_1, \delta_2\}.$$

Further, let $I = (k, X, \Omega, \mathcal{A}, K)$ be an instance of WCA CLUSTERING, let $(\bar{X}, \bar{\Omega})$ be an (ϵ_1, δ_1) -coreset for I , and let $(\tilde{X}, \tilde{\Omega})$ be an (ϵ_2, δ_2) -coreset for $\bar{I} = (k, \bar{X}, \bar{\Omega}, \mathcal{A}, K)$. Then, $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -coreset for I .

Proof. See Section 2.6.8 □

Second ingredient: To obtain the desired data set $(\tilde{X}, \tilde{\Omega})$ we merge the points on each line into batches by replacing them with their centroids and adjusting their weights. More precisely, let $L \in \mathcal{L}$. Starting from the leftmost point, we successively add, from left to right, points of $\tilde{X} \cap L$ to form the first batch $B = B_1$ until

$$\mathcal{V}_E(B) = \mathcal{V}_E = \frac{\epsilon^2}{32\alpha k \lambda^+(\mathcal{A})|\mathcal{L}|} \text{ALG}_{(\alpha, \beta)}(X),$$

and continue with the next batch. The above condition can always be achieved by assigning, if needed, the last point fractionally to the batch. In such a case, we may assume that all points are completely assigned to a single batch by splitting this point into two with the appropriate weights. While this assumption simplifies the exposition, it does not effect on the results and can be made without loss of generality.

The process is continued until all points are assigned to batches. Note that all batches have the same error, apart from the last one, whose error may be smaller. Let $\mathcal{B}(L)$ denote the set of batches on the line L , and let $\mathcal{B}$ signify the total set of all batches, i.e., the union of all $\mathcal{B}(L)$ with $L \in \mathcal{L}$.

Finally, we merge the points of each batch. So, let $B \in \mathcal{B}$, and let

$$\omega_B = \sum_{\bar{x}_j \in \tilde{X} \cap B} \omega_j, \quad x_B = \frac{1}{\omega_B} \sum_{\bar{x}_j \in \tilde{X} \cap B} \omega_j \bar{x}_j$$

be the set's $\tilde{X} \cap B$ total weight and centroid, respectively. Then we replace the points of $\tilde{X} \cap B$ by their centroid x_B and assign ω_B as its weight.

The data set that results from applying the merger to all lines of the pencils will be called (X_B, Ω_B) . This is actually the desired set $(\tilde{X}, \tilde{\Omega})$, and we will use both notations interchangeably, depending on which one leads to the more intuitive exposition in the subsequent proofs. In particular, the number of points will still be denoted by $\tilde{n}$. If we index the batches as $B_{\tilde{j}}$ for $\tilde{j} \in [\tilde{n}]$ and set $x_{\tilde{j}} = x_{B_{\tilde{j}}}$, the corresponding merging function $p : [n] \rightarrow [\tilde{n}]$ is again specified by

$$p(j) = \tilde{j} \iff \bar{x}_j \in B_{\tilde{j}},$$

and the extension $f_2 : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, \tilde{X}, \Omega)$ is again defined by

$$\bar{\xi}_{ij} = \tilde{\xi}_{ip(j)} \quad \forall j \in [n], i \in [k].$$

Hence, the extension $f : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, X, \Omega)$ is simply given by

$$f = f_1 \circ f_2, \quad \text{i.e.,} \quad \xi_{ij} = \tilde{\xi}_{ip(j)} \quad \forall j \in [n], i \in [k].$$

This completes the construction.

Let us point out that our line construction differs from that of [HPK07]. By using the center of gravity for each batch together, as we will see, with a new proof which allows us to bound the errors of all batches at once – instead of summing over the individual batch errors – we reduce the dependency of the construction on k from cubic to quadratic. Further, our analysis is more general as it also works for anisotropic power diagrams. While ellipsoidal norms create additional technicalities our proof mainly relies on geometrical and structural properties of anisotropic power diagrams.

Before we prove that $(X_{\mathcal{B}}, \Omega_{\mathcal{B}})$ is an (ϵ, δ) -coreset, we bound its cardinality $\tilde{n}$ in this section. The basic idea for deriving a bound is due to [HPK07]. However, we improve on their arguments to reduce the dependency of the estimate by a factor of k .

Lemma 2.16

$$|\mathcal{B}| \in \mathcal{O}\left(\frac{k^2}{\epsilon^{d+1}}\right),$$

i.e., the number of points in $X_{\mathcal{B}}$ is of the order k^2/ϵ^{d+1} .

Proof. To establish the asserted bound, let $\bar{c}_1, \dots, \bar{c}_k$ be the centroids of an optimal least-squares clustering of $(\bar{X}, \Omega)$. In this unconstrained Euclidean case, the clustering is compatible with a Voronoi diagram. As we need not respect any weight constraints for the clusters, we may assume that no point of $\bar{X}$ lies on the boundary of any Voronoi cell. Since L intersects each Voronoi cell in a (possibly empty or one-element) interval, all points in the same interval lie in the same cluster. This implies that for at most $k - 1$ batches in L , the points are assigned to more than one cluster, i.e., at least $|\mathcal{B}(L)| - k + 1$ such batches are fully assigned to a single cluster.

Now, let $B \in \mathcal{B}(L)$ be contained in the i th cluster, but let B not be the last batch on L . As the error $\mathcal{V}_E(B)$ is known from the construction, we obtain from Lemma 2.5 (with $X_0 = B$, $c_0 = c_B$, $\omega(X_0) = \omega_B$ and $A = E$) that

$$\sum_{\bar{x} \in B} \omega_{\bar{x}} \|\bar{x} - \bar{c}_i\|_2^2 = \omega_B \|c_B - \bar{c}_i\|_2^2 + \mathcal{V}_E(B) \geq \mathcal{V}_E(B) = \mathcal{V}_E.$$

As at most $(k - 1) + 1 = k$ of the batches $B \in \mathcal{B}(L)$ can contribute less than $\mathcal{V}_E$ to the cost $\text{OPT}(\bar{X})$ of the clustering, we see that

$$|X_{\mathcal{B}}| \leq \frac{\text{OPT}(\bar{X})}{\mathcal{V}_E} + k \cdot |\mathcal{L}|.$$

Since $(\bar{X}, \Omega)$ is a 0-offset ϵ -coreset for (X, Ω) by Lemma 2.14, we can use Lemma 2.6 (b') to obtain

$$\text{OPT}(\bar{X}) \leq (1 + \epsilon) \cdot \text{OPT}(X) \leq 2 \cdot \text{ALG}_{(\alpha, \beta)}(X).$$

All in all,

$$|X_{\mathcal{B}}| \leq 2 \cdot \frac{\text{ALG}_{(\alpha, \beta)}(X)}{\mathcal{V}_E} + k \cdot |\mathcal{L}| = \left(64 \cdot \alpha \cdot \lambda^+(\mathcal{A}) \cdot \frac{k}{\epsilon^2} + k\right) \cdot |\mathcal{L}|.$$

As the right term is in $\mathcal{O}\left(\frac{k^2}{\epsilon^{d+1}}\right)$, this concludes the proof. $\square$

2.4.2 Structural properties of clusterings admitting strictly compatible diagrams

We will now show that each clustering that admits a strictly compatible diagram gives rise to a partition of each line L into at most $2k - 1$ intervals such that no data point in the relative interior of each interval is fractionally assigned, i.e., belongs to more than one cluster. Since by Proposition 2.12 it suffices to consider such diagrams on the coresets, this will allow us to verify coreset condition (b) of Definition 2.1. Condition (a), on the other hand, does not require additional assumptions on the structure of the clustering.

Given an instance $(k, \tilde{X}, \tilde{\Omega}, \mathcal{A}, K, S)$ of WCA ASSIGNMENT, suppose that $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ attains $\text{cost}(\tilde{X}, \tilde{\Omega}, S)$ and admits a strictly compatible diagram $\mathcal{P} = \mathcal{P}(T, \Sigma, \mathcal{A}) = \{P_1, \dots, P_k\}$. Note that $\tilde{C}$ is available by Proposition 2.12.

We use $\mathcal{P}$ to bound the “structural complexity” of $\tilde{C}$ ’s intersections with lines L . Later we apply the results to those lines that constitute the pencils on which $(\tilde{X}, \tilde{\Omega})$ “lives”.

The first lemma considers the “generic situation” and could be stated within the realm of *Davenport-Schinzel Sequences*; see, e.g., [Sha+95]. We will, however, give a direct, self-contained proof of the relevant result.

So, let $c \in \mathbb{R}$, $q \in \mathbb{R}^d \setminus \{0\}$, $L = L(c, q)$, and, for $i \in [k]$,

$$g_i : \mathbb{R}^d \rightarrow \mathbb{R}, \quad g_i(x) = \|x - s_i\|_{A_i}^2 + \sigma.$$

Further, let

$$h : \mathbb{R}^d \rightarrow \mathbb{R}, \quad h(x) = \min\{g_i(x) : i \in [k]\}$$

be the lower envelope of $g_1, \dots, g_k$. Then, of course, the diagram $\mathcal{P}$ is induced by $g_1, \dots, g_k$, and

$$L \cap P_i = \{x \in L : g_i(x) = h(x)\}$$

consists of finitely many connected components, all closed intervals and *proper*, i.e., 1-dimensional or possibly singletons. Let γ_i denote their number and let

$$g_i|_L : L \rightarrow \mathbb{R}, \quad g_i|_L(x) = g_i(x)$$

be the *restriction of g_i to the line L* . Then, we have the following bound.

Lemma 2.17

Suppose that no two of the restrictions $g_i|_L$ coincide. Then,

$$\sum_{i=1}^k \gamma_i \leq 2k - 1,$$

and this bound is the best possible.

Proof. The proof of the inequality is by induction with respect to k . As the assertion is trivial for $k_0 = 1$, suppose that it has been verified for some $k_0 \in \mathbb{N}$, and let $k = k_0 + 1$.

Suppose, first, that there exists an index $i^* \in [k]$ such that $\gamma_{i^*} = 1$. By the induction hypothesis, we have $\sum_{i \in [k] \setminus \{i^*\}} \gamma_i \leq 2(k-1) - 1$, and adding g_{i^*} back in, the corresponding interval can split at most one other interval by the assumption that no two of the restrictions $g_i|_L$ coincide. Hence, the assertion holds.

So, suppose in the following, that $\gamma_i \geq 2$ for all $i \in [k]$. We will show that this leads to a contradiction, thereby proving the theorem.

For $i \in [k]$, let $y_j^{(i)}$ be a relative interior point in the j th component of $L \cap P_i$, ordered such that $y_1^{(i)} < \dots < y_{\gamma_i}^{(i)}$. Further, let

$$i_0 \in M_1 = \operatorname{argmax}\{y_1^{(i)} : i \in [k]\}.$$

If the index i_0 is unique, set $i^* = i_0$. Otherwise let $i^* \in M_2 = \operatorname{argmin}\{y_2^{(i)} : i \in M_1\}$. Note that M_2 still does not need to be a singleton, but the different choices do not affect the subsequent arguments.

Since $\gamma_{i^*} \geq 2$, there must be $i \in [k] \setminus \{i^*\}$ and $j \in [\gamma_i]$ such that

$$y_1^{(i^*)} < y_j^{(i)} < y_2^{(i^*)}.$$

As $i^* \in M_1$, we know that $j \geq 2$ and $y_1^{(i)} \leq y_1^{(i^*)}$. Equality, however, is excluded by the choice of $i^* \in M_2$. Hence,

$$y_1^{(i)} < y_1^{(i^*)} < y_j^{(i)} < y_2^{(i^*)}.$$

This implies that $g_i - g_{i^*}$ has at least three roots in L , i.e., $g_i|_L = g_{i^*}|_L$, which contradicts our general assumption.

The bound is tight for examples of suitably nested parabolas. $\square$

As a consequence, we obtain the following result.

Corollary 2.18

Suppose again that no two of the restrictions $g_i|_L$ coincide. Then, there is a dissection of L into at most $2k - 1$ proper intervals such that the relative interior of each interval is contained in exactly one of the cells of $\mathcal{P}$.

Proof. For $i \in [k]$, let $\mathcal{I}_i = \mathcal{I}_i^{(0)}$ denote the set of the relative interiors of all proper intervals that occur as connected components of $L \cap P_i$, and set $\mathcal{I} = \mathcal{I}^{(0)} = \mathcal{I}_1 \cup \dots \cup \mathcal{I}_k$. Since, by assumption, all univariate polynomials $g_i|_L$ are different, any two intervals in $\mathcal{I}$ are disjoint.

For $i \in [k]$, let $\gamma_i^{(0)}$ denote the number of singletons that occur as connected components of $L \cap P_i$, and set $\gamma_i^{(1)} = |\mathcal{I}_i|$. Then, of course,

$$\gamma_i = \gamma_i^{(0)} + \gamma_i^{(1)} \quad \forall i \in [k], \quad \text{and} \quad |\mathcal{I}| = \sum_{i=1}^k \gamma_i^{(1)}.$$

Starting with $\mathcal{I}^{(0)}$, we will now construct refinements $\mathcal{I}^{(\ell)}$ by successively splitting intervals that violate the assertion as long as such intervals exist.

So, let $i_0 \in [k]$, let $I_0 \in \mathcal{I}_{i_0}$, and suppose that there exists a point $y \in I_0$ that also belongs to some other cell P_{i_1} . Then, y is the unique point of $P_{i_0} \cap P_{i_1}$ in $\text{relint}(I_0)$ since, otherwise, $g_{i_0}|_L - g_{i_1}|_L$ would have at least three local extrema, contradicting the assumption that $g_{i_0}|_L \neq g_{i_1}|_L$. For the same reason, no other interval $I \in \mathcal{I}_{i_0}$ can contain a point of P_{i_1} in its relative interior.

Now, we split I_0 at y into two intervals and obtain a new set $\mathcal{I}^{(1)}$ by replacing I_0 in $\mathcal{I}_{i_0}$ by the corresponding two relatively open intervals $I_-^{(0)}$ and $I_+^{(0)}$. Note that neither $I_-^{(0)}$ nor $I_+^{(0)}$ contains points of P_{i_1} anymore, and that $\{y\}$ is a connected component of $L \cap P_{i_1}$, which counted towards $\gamma_{i_1}^{(0)}$.

We continue by successively splitting intervals that violate the assertion and finally obtain a set $\mathcal{I}^*$ of intervals that have the desired property. The total number of necessary splits is bounded by $\sum_{i=1}^k \gamma_i^{(0)}$. Hence, with the aid of Lemma 2.17,

$$|\mathcal{I}^*| \leq \sum_{i=1}^k \gamma_i^{(1)} + \sum_{i=1}^k \gamma_i^{(0)} = \sum_{i=1}^k \gamma_i \leq 2k - 1,$$

which is the asserted bound. $\square$

As an immediate consequence, the corollary yields a structural result when $\tilde{C}$ and $\mathcal{P}$ are compatible.

Corollary 2.19

Let $g_1|_L, \dots, g_k|_L$ be all different, and let $\tilde{C}$ and $\mathcal{P}$ be compatible. Then, there exists a partition of L into at most $2k - 1$ proper intervals such that all data points in the relative interior of each interval are fully assigned to the same cluster.

Note, however, that the “genericity assumption” that the k univariate polynomials $g_1|_L, \dots, g_k|_L$ are all different for each line L can generally not be guaranteed, even if all multivariate polynomials $g_1, \dots, g_k$ are different. For instance, when $\mathcal{A} = \mathcal{E}$, the

cells of $\mathcal{P}$ are polyhedra that might all have a line (or even some higher-dimensional affine subspace) in common. In this case, it may happen that all points of $\tilde{X}$ in the relative interior of a component I of $L \cap P_i$ are fractionally assigned to C_i the number of splitting operations will generally not be bounded by a function of k only, and Corollary 2.19 does not hold anymore; see Figure 2.1 for an illustration.

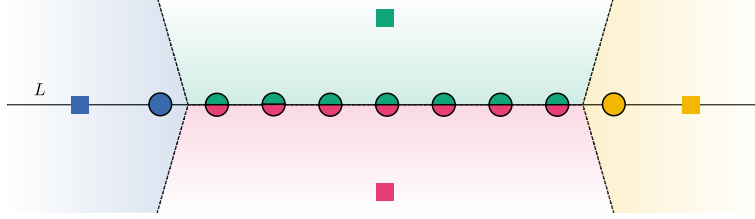


Figure 2.1: A clustering and diagram that are strongly compatible. Sites are depicted as squares. The clustering assigns all points on the boundary between the green and red cluster with a proportion of $1/2 : 1/2$ to the two clusters. Consequently, a dissection of the horizontal line L with the required properties of Corollary 2.19 requires more than $2k - 1 = 2 \cdot 4 - 1 = 7$ intervals.

In fact, a clustering that admits a feasible or even strongly compatible diagram does not suffice to provide the bound required later, and we need the additional properties of strictly compatible diagrams to prove the following theorem.

Theorem 2.20

Let $\bar{C} \in \mathcal{C}_K(k, \bar{X}, \Omega)$ admit a strictly compatible diagram, and let L be a line in $\mathbb{R}^d$. Then, there is a partition of L into at most $2k - 1$ proper intervals such that all data points in the relative interior of each interval are fully assigned to the same cluster.

Proof. Let $\mathcal{P} = \mathcal{P}(T, \Sigma, \mathcal{A}) = \{P_1, \dots, P_k\}$ be a diagram such that $\bar{C}$ and $\mathcal{P}$ are strictly compatible. Suppose that exactly k' of the univariate polynomials $g_i|_L$ are different, let $N^+ \subset [k]$ be the set of the corresponding indices, and set $N^- = [k] \setminus N^+$.

If $k' = k$, the assertion follows from Corollary 2.19. So, suppose that $k' < k$. By Corollary 2.19, we obtain a partition $\mathcal{I}$ of L into at most $2k' - 1$ proper intervals with the following property: If $I \in \mathcal{I}$ and $\tilde{x} \in \tilde{X} \cap \text{relint}(I)$, then $\tilde{x} \notin P_i$ for any $i \in N^+$.

Since $\bar{C}$ and $\mathcal{P}$ are strictly compatible, $P_i \cap P_\ell$ contains at most one point for $i \neq \ell$. Since, for each $i \in N^-$, each $g_i|_L$ coincides with one of the $g_\ell|_L$ with $\ell \in N^+$, splitting intervals for each such points will result in an additional number of at most $|N^-|$ intervals in order to guarantee the desired property. Hence we obtain a total of at most

$$2|N^+| - 1 + |N^-| = 2k' - 1 + k - k' = k + k' - 1 \leq 2k - 1$$

intervals that have the required properties. $\square$

2.4.3 Proof of the coreset properties

We will now prove various lemmas that will subsequently be used to show that $(\tilde{X}, \tilde{\Omega}) = (X_{\mathcal{B}}, \Omega_{\mathcal{B}})$ is a coreset for $(k, X, \Omega, \mathcal{A}, K)$, thereby proving Theorem 2.2. Note that, by Lemma 2.14 and Lemma 2.15, it suffices to show that $(X_{\mathcal{B}}, \Omega_{\mathcal{B}})$ is a coreset for $(k, \bar{X}, \Omega, \mathcal{A}, K)$.

We begin with a simple observation that relates the batch errors in the anisotropic case to those in the Euclidean case.

Lemma 2.21

Let $\mathcal{B}' \subset \mathcal{B}$. Then,

$$\lambda^-(\mathcal{A}) \cdot \sum_{B \in \mathcal{B}'} \nu_E(B) \leq \sum_{i=1}^k \sum_{B \in \mathcal{B}'} \xi_{iB} \nu_{A_i}(B) \leq \lambda^+(\mathcal{A}) \cdot \sum_{B \in \mathcal{B}'} \nu_E(B).$$

Proof. See Section 2.6.9 □

The following lemma essentially handles condition (a) of Definition 2.1 for each set S of sites which is the simple part of showing the coreset property.

Lemma 2.22

Let $C_{\mathcal{B}} \in \mathcal{C}_K(k, X_{\mathcal{B}}, \Omega_{\mathcal{B}})$. Then,

$$\text{cost}_{\mathcal{A}, K}(\bar{X}, \Omega, f_2(C_{\mathcal{B}}), S) \leq \text{cost}_{\mathcal{A}, K}(X_{\mathcal{B}}, \Omega_{\mathcal{B}}, C_{\mathcal{B}}, S) + \lambda^+(\mathcal{A}) \cdot \sum_{B \in \mathcal{B}} \nu_E(B).$$

Proof. See Section 2.6.10. □

In order to verify that $(X_{\mathcal{B}}, \Omega_{\mathcal{B}})$ is a coreset for $(k, \bar{X}, \Omega, \mathcal{A}, K)$, we will address now condition (b) of Definition 2.1. With the aid of Proposition 2.12, let $\bar{C}^* \in \mathcal{C}_K(k, \bar{X}, \Omega)$ such that it attains $\text{cost}_{\mathcal{A}, K}(\bar{X}, \Omega, S)$ and admits a strictly compatible anisotropic diagram $\mathcal{P} = \mathcal{P}(T, \Sigma, \mathcal{A})$. We will show that, for $C_{\mathcal{B}} = p(\bar{C}^*)$,

$$\text{cost}_{\mathcal{A}, K}(X_{\mathcal{B}}, \Omega_{\mathcal{B}}, C_{\mathcal{B}}, S) + \Delta^- \leq (1 + \epsilon) \text{cost}_{\mathcal{A}, K}(\bar{X}, \Omega, \bar{C}^*, S),$$

as this yields the inequality

$$\begin{aligned} \text{cost}_{\mathcal{A}, K}(X_{\mathcal{B}}, \Omega_{\mathcal{B}}, S) + \Delta^- &\leq \text{cost}_{\mathcal{A}, K}(X_{\mathcal{B}}, \Omega_{\mathcal{B}}, C_{\mathcal{B}}, S) + \Delta^- \\ &\leq (1 + \epsilon) \text{cost}_{\mathcal{A}, K}(\bar{X}, \Omega, \bar{C}^*, S) = (1 + \epsilon) \text{cost}_{\mathcal{A}, K}(\bar{X}, \Omega, S), \end{aligned}$$

needed for condition (b). The next two lemmas will prepare the ground to verify the former inequality.

The clustering $\bar{C}^*$ partitions $\mathcal{B}$ into the set $\mathcal{B}_+$ of those batches B whose points are all assigned to the same cluster, and its complement $\mathcal{B}_- = \mathcal{B} \setminus \mathcal{B}_+$. Similarly, $(\bar{X}, \Omega)$

and $(X_{\mathcal{B}}, \Omega_{\mathcal{B}})$ are split into $(\bar{X}|_{\pm}, \Omega|_{\pm})$ and $(X_{\mathcal{B}}|_{\pm}, \Omega_{\mathcal{B}}|_{\pm})$, and the clusterings $\bar{C}^*$ and $C_{\mathcal{B}}$ are split into their restrictions $\bar{C}^*|_{\pm}$ and $C_{\mathcal{B}}|_{\pm}$ of $(\bar{X}|_{\pm}, \Omega|_{\pm})$ and $(X_{\mathcal{B}}|_{\pm}, \Omega_{\mathcal{B}}|_{\pm})$, respectively. As the clustering cost is additive in the contributions of each point, we can consider the costs

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(\bar{X}|_{\pm}, \bar{C}^*|_{\pm}, S) &= \sum_{i=1}^k \sum_{B \in \mathcal{B}_{\pm}} \sum_{\bar{x}_j \in B} \bar{\xi}_{ij} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2, \\ \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_{\pm}, \Omega_{\mathcal{B}}|_{\pm}, C_{\mathcal{B}}|_{\pm}, S) &= \sum_{i=1}^k \sum_{B \in \mathcal{B}_{\pm}} \xi_{iB} \omega_B \|x_B - s_i\|_{A_i}^2 \end{aligned}$$

incurred by points in batches from $\mathcal{B}_+$ and $\mathcal{B}_-$ separately. (Let us point out that the subscript K is used here to indicate that we are still in the constrained case. The bounds in K, however, only apply to the clusterings on the unsplit data sets.)

Note that, for each $B \in \mathcal{B}_+$,

$$\xi_{iB} = \bar{\xi}_{ij} = \begin{cases} 1 & \text{for } \bar{x}_j \in B; \\ 0 & \text{for } \bar{x}_j \notin B. \end{cases}$$

This property is used in the next lemma to show that batches in $\mathcal{B}_+$ behave nicely.

Lemma 2.23

$$\text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_+, \Omega_{\mathcal{B}}|_+, \tilde{X}|_+, C_{\mathcal{B}}|_+, S) + \lambda^-(\mathcal{A}) \sum_{B \in \mathcal{B}_+} \mathcal{V}_E(B) \leq \text{cost}_{\mathcal{A},K}(\bar{X}|_+, \Omega|_+, \bar{C}^*|_+, S).$$

Proof. Since all points of each batch $B \in \mathcal{B}_+$ have been assigned to the same cluster, we see with the aid of Lemma 2.5 (a) (with $X_0 = B$, $c_0 = x_B$) that

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X|_+, \Omega|_+, \bar{C}^*|_+, S) &= \sum_{i=1}^k \sum_{B \in \mathcal{B}_+} \sum_{\bar{x}_j \in B} \bar{\xi}_{ij} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2 \\ &= \sum_{i=1}^k \sum_{B \in \mathcal{B}_+} \xi_{iB} \sum_{\bar{x}_j \in B} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2 = \sum_{i=1}^k \sum_{B \in \mathcal{B}_+} \xi_{iB} \left(\omega_B \|x_B - s_i\|_{A_i}^2 + \mathcal{V}_{A_i}(B) \right) \\ &= \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_+, \Omega_{\mathcal{B}}|_+, \tilde{X}|_+, C_{\mathcal{B}}|_+, S) + \sum_{i=1}^k \sum_{B \in \mathcal{B}_+} \xi_{iB} \mathcal{V}_{A_i}(B). \end{aligned}$$

Now, the asserted inequality follows with the aid of Lemma 2.21. $\square$

While the proof of Lemma 2.23 did not make any use of the special choice of the optimal clustering, the next lemma, Lemma 2.24, which deals with $\mathcal{B}_-$, relies heavily

on the property that $\bar{C}^*$ admits a strictly compatible diagram. (It can also be used to fix what seems to be a flaw in the proof [HPK07, Theorem 3.7] where an equation is asserted while only an inequality is clear.)

Lemma 2.24

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_{-}, \Omega_{\mathcal{B}}|_{-}, C_{\mathcal{B}}|_{-}, S) + \lambda^{-}(\mathcal{A}) \sum_{B \in \mathcal{B}_{-}} \nu_E(B) \\ \leq \text{cost}_{\mathcal{A},K}(\bar{X}|_{-}, \Omega|_{-}, \bar{C}^*|_{-}, S) + \epsilon \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S). \end{aligned}$$

Proof. First, by the definition of the merging function p ,

$$\xi_{iB}\omega_B = \sum_{\bar{x}_j \in B} \bar{\xi}_{ij}\omega_j.$$

Hence,

$$\text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_{-}, \Omega_{\mathcal{B}}|_{-}, C_{\mathcal{B}}|_{-}, S) = \sum_{i=1}^k \sum_{B \in \mathcal{B}_{-}} \sum_{\bar{x}_j \in B} \bar{\xi}_{ij}\omega_j \|x_B - s_i\|_{A_i}^2.$$

Next, we apply Lemma 2.10 to $(\bar{X}|_{-}, \Omega|_{-})$ and $\bar{C}^*|_{-}$. In this setting, the number D becomes

$$D = \sum_{i=1}^k \sum_{B \in \mathcal{B}_{-}} \sum_{\bar{x}_j \in B} \bar{\xi}_{ij}\omega_j \|x_B - \bar{x}_j\|_{A_i}^2,$$

and Lemma 2.10 yields

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_{-}, \Omega_{\mathcal{B}}|_{-}, C_{\mathcal{B}}|_{-}, S) \\ \leq \text{cost}_{\mathcal{A},K}(\bar{X}|_{-}, \Omega|_{-}, \bar{C}^*|_{-}, S) + 2\sqrt{D \cdot \text{cost}_{\mathcal{A},K}(\bar{X}|_{-}, \Omega|_{-}, \bar{C}^*|_{-}, S)} + D. \end{aligned}$$

We will now apply Lemma 2.9 to derive an upper bound for D . In order to do so, let us first consider the lemma's assumption. In fact, we have

$$\sum_{B \in \mathcal{B}_{-}} \sum_{x_j \in B} \omega_j \|x_B - \bar{x}_j\|_2^2 = \sum_{B \in \mathcal{B}_{-}} \nu_E(B) \leq |\mathcal{B}_{-}| \cdot \nu_E = \frac{|\mathcal{B}_{-}| \cdot \epsilon^2}{32\alpha k \lambda^{+}(\mathcal{A}) |\mathcal{L}|} \text{ALG}_{(\alpha, \beta)}(X).$$

Hence we need to bound the number of batches in $\mathcal{B}_{-}$. We consider the different lines of the pencils separately.

So, let $L \in \mathcal{L}$. Since $\bar{C}^*$ admits a strictly compatible diagram we can apply Theorem 2.20. There is thus a partition of L into at most $2k - 1$ proper intervals such that all data points in the relative interior of each interval are fully assigned to the same

cluster. Consequently, only batches that contain a boundary point of at least one such interval can contain points of more than one cluster. Therefore, $|\mathcal{B}_- \cap \mathcal{B}(L)| \leq 2(k-1)$, and thus

$$|\mathcal{B}_-| \leq 2(k-1) \cdot |\mathcal{L}|.$$

Since $(\bar{X}, \Omega)$ is a 0-offset ϵ -coreset for (X, Ω) and $\epsilon \in (0, 1/2]$, we obtain

$$\sum_{B \in \mathcal{B}_-} \sum_{x_j \in B} \omega_j \|x_B - \bar{x}_j\|_2^2 \leq \frac{\epsilon^2}{16\lambda^+(\mathcal{A})} \text{OPT}(X) \leq \frac{\epsilon^2}{8\lambda^+(\mathcal{A})} \text{OPT}(\bar{X}).$$

Lemma 2.9, applied to $(\bar{X}, \Omega)$ with $\eta = \epsilon^2/8$ yields

$$D = \sum_{i=1}^k \sum_{B \in \mathcal{B}_-} \sum_{\bar{x}_j \in B} \bar{\xi}_{ij} \omega_j \|x_B - \bar{x}_j\|_{A_i}^2 \leq \frac{\epsilon^2}{8} \text{OPT}_{\mathcal{A}}(\bar{X}) \leq \frac{\epsilon^2}{8} \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S).$$

Since

$$\text{cost}_{\mathcal{A},K}(\bar{X}|_-, \Omega|_-, \bar{C}^*|_-, S) \leq \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S),$$

we obtain

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_-, \Omega_{\mathcal{B}}|_-, C_{\mathcal{B}}|_-, S) &\leq \text{cost}_{\mathcal{A},K}(\bar{X}|_-, \Omega|_-, \bar{C}^*|_-, S) \\ &\quad + \left(2\sqrt{\frac{\epsilon^2}{8}} + \frac{\epsilon^2}{8} \right) \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S). \end{aligned}$$

Finally, with the aid of Lemma 2.21, the previous estimates also imply that

$$\begin{aligned} \sum_{i=1}^k \sum_{B \in \mathcal{B}_-} \xi_{iB} \mathcal{V}_{A_i}(B) &\leq \lambda^+(\mathcal{A}) \cdot \sum_{B \in \mathcal{B}_-} \mathcal{V}_E(B) = \lambda^+(\mathcal{A}) \cdot \sum_{B \in \mathcal{B}_-} \sum_{x_j \in B} \omega_j \|x_B - \bar{x}_j\|_2^2 \\ &\leq \frac{\epsilon^2}{8} \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S). \end{aligned}$$

Therefore, using Lemma 2.21 again,

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_-, \Omega_{\mathcal{B}}|_-, C_{\mathcal{B}}|_-, S) &+ \lambda^-(\mathcal{A}) \sum_{B \in \mathcal{B}_-} \mathcal{V}_E(B) \\ &\leq \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}|_-, \Omega_{\mathcal{B}}|_-, C_{\mathcal{B}}|_-, S) + \sum_{i=1}^k \sum_{B \in \mathcal{B}_-} \xi_{iB} \mathcal{V}_{A_i}(B) \\ &\leq \text{cost}_{\mathcal{A},K}(\bar{X}|_-, \Omega|_-, \bar{C}^*|_-, S) + \left(2\sqrt{\frac{\epsilon^2}{8}} + 2 \cdot \frac{\epsilon^2}{8} \right) \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S) \\ &\leq \text{cost}_{\mathcal{A},K}(\bar{X}|_-, \Omega|_-, \bar{C}^*|_-, S) + \epsilon \cdot \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S), \end{aligned}$$

as claimed. $\square$

Now, we are ready to verify that (X_B, Ω_B) is a coreset for $(k, \bar{X}, \Omega, \mathcal{A}, K)$.

Theorem 2.25

(X_B, Ω_B) is an $\left(\epsilon, \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})}\right)$ -coreset for $(k, \bar{X}, \Omega, \mathcal{A}, K)$.

Proof. Let

$$\Delta^+ = \lambda^+(\mathcal{A}) \cdot \sum_{B \in \mathcal{B}} \mathcal{V}_E(B), \quad \text{and} \quad \Delta^- = \lambda^-(\mathcal{A}) \cdot \sum_{B \in \mathcal{B}} \mathcal{V}_E(B).$$

Then,

$$\Delta^+ = \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})} \cdot \Delta^-.$$

Further, Lemma 2.22 immediately yields

$$\text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, f_2(C_B), S) \leq \text{cost}_{\mathcal{A},K}(X_B, \Omega_B, C_B, S) + \Delta^+,$$

which shows condition (a) of Definition 2.1.

In order to verify (b), remember that it suffices to show:

$$\text{cost}_{\mathcal{A},K}(X_B, \Omega_B, C_B, S) + \Delta^- \leq (1 + \epsilon) \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S).$$

We combine Lemmata 2.23 and 2.24, using

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X_B, \Omega_B, C_B, S) + \Delta^- &= \text{cost}_{\mathcal{A},K}(X_B|_+, \Omega_B|_+, C_B|_+, S) + \lambda^-(\mathcal{A}) \sum_{B \in \mathcal{B}_+} \mathcal{V}_E(B) \\ &\quad + \text{cost}_{\mathcal{A},K}(X_B|_-, \Omega_B|_-, C_B|_-, S) + \lambda^-(\mathcal{A}) \sum_{B \in \mathcal{B}_-} \mathcal{V}_E(B). \end{aligned}$$

This yields

$$\begin{aligned} &\text{cost}_{\mathcal{A},K}(X_B, \Omega_B, C_B, S) + \Delta^- \\ &\leq \text{cost}_{\mathcal{A},K}(\bar{X}|_+, \Omega|_+, \bar{C}^*|_+, S) + \text{cost}_{\mathcal{A},K}(\bar{X}|_-, \Omega|_-, \bar{C}^*|_-, S) + \epsilon \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S) \\ &\leq (1 + \epsilon) \text{cost}_{\mathcal{A},K}(\bar{X}, \Omega, \bar{C}^*, S), \end{aligned}$$

which implies condition (b) of Definition 2.1. $\square$

Finally, we have all the ingredients with which to prove our main theorem.

Proof (Proof of Theorem 2.2). The assertion that $(\tilde{X}, \tilde{\Omega}) = (X_B, \Omega_B)$ is a coreset for the instance $(k, X, \Omega, \mathcal{A}, K)$ follows immediately from our previous results by means of Lemma 2.15.

More precisely, we apply Lemma 2.14 for $\epsilon_1 = \epsilon/3$ to obtain a 0-offset coreset, i.e., $\delta_1 = 1$, and Theorem 2.25 for $\epsilon_2 = \epsilon/3$ and $\delta_2 = \lambda^+(\mathcal{A})/\lambda^-(\mathcal{A})$. Since

$$\epsilon \geq \epsilon_1 + \epsilon_2 + \epsilon_1 \epsilon_2 \quad \text{and} \quad \delta = \max\{\delta_1, \delta_2\} = \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})},$$

the assertion follows from Lemma 2.15. $\square$

2.5 Final remarks

Our results provide small coresets for WCA CLUSTERING, i.e., in the weight-constrained anisotropic case. This allows to compute good clusterings for the generally much smaller sets $(\tilde{X}, \tilde{\Omega})$ and to subsequently convert them to clusterings of the original data sets (X, Ω) . While the corresponding extensions f preserve the cluster weights and guarantee the feasibility of the obtained clusterings, we might, however, lose the favorable properties of compatibility with diagrams. Hence, the question arises as to whether any efficiently computable extensions exist that map clusterings that admit a (strictly, strongly, or just) compatible diagram on the coresets to clusterings of the same type on the full data set. Alternatively, we might ask if we can find a diagram on the coreset such that the induced clustering on the original data set only slightly violates the cluster size constraints. This leads to a number of relevant stability issues.

One may also wonder whether, in our setting, the dependence on k can be further reduced.

2.6 Technical results

Here we add all proofs which were omitted in the main part of the chapter.

2.6.1 Proof of Lemma 2.5

(a) Since $\sum_{x \in X_0} \omega_x x = \omega(X_0)c_0$, we have

$$\begin{aligned} \sum_{x \in X_0} \omega_x \|x - s\|_A^2 &= \sum_{x \in X_0} \omega_x (\|x - c_0\|_A^2 + 2(x - c_0)^\top A(c_0 - s) + \|c_0 - s\|_A^2) \\ &= \omega(X_0) \cdot \|c_0 - s\|_A^2 + \mathcal{V}_A(X_0) + 2 \left(\sum_{x \in X_0} \omega_x x - \omega(X_0)c_0 \right)^\top A(c_0 - s) \\ &= \omega(X_0) \cdot \|c_0 - s\|_A^2 + \mathcal{V}_A(X_0). \end{aligned}$$

(b) Let, as usual, $C = (C_1, \dots, C_k)$ and define the weighted data set (X_i, Ω_i) for $i \in [k]$, by

$$X_i = \text{supp}(C_i) \quad \text{and} \quad \Omega_i = \{\xi_{ij}\omega_j : x_j \in X_i\}.$$

Then, by applying (a) to $(X_0, \Omega_0) = (X_i, \Omega_i)$ and $A = A_i$, for $i \in [k]$, we obtain

$$\begin{aligned} \text{cost}_{A,K}(X, \Omega, C, S) &= \sum_{i=1}^k \sum_{j=1}^n \xi_{ij}\omega_j \|x_j - s_i\|_{A_i}^2 = \sum_{i=1}^k \left(\omega(C_i) \cdot \|c(C_i) - s_i\|_{A_i}^2 + \mathcal{V}_{A_i}(X_i) \right) \\ &\geq \sum_{i=1}^k \mathcal{V}_{A_i}(X_i) = \text{cost}_{A,K}(X, \Omega, C). \end{aligned}$$

Thus, $\text{cost}_{A,K}(X, \Omega, C, S) \geq \text{cost}_{A,K}(X, \Omega, C)$ with equality if and only if, for each $i \in [k]$, either $\omega(C_i) = 0$ or $s_i = c(C_i)$.

2.6.2 Proof of Lemma 2.6

(a'): Let $S = c(\tilde{C})$. Then, by Lemma 2.5 (b)

$$\text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C})) = \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), c(f(\tilde{C})) \leq \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S),$$

and inequality (a) of Definition 2.1 yields

$$(1 - \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C})) \leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta^+ = \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}) + \Delta^+.$$

(b'): Let C^* be a clustering attaining $\text{OPT}_{\mathcal{A},K}(X)$ and set $S = c(C^*)$. Then, by Lemma 2.5 (b)

$$\text{OPT}_{\mathcal{A},K}(\tilde{X}) \leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S),$$

and, hence, by inequality (b) of Definition 2.1

$$\text{OPT}_{\mathcal{A},K}(\tilde{X}) + \Delta^- \leq (1 + \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, S) = (1 + \epsilon) \text{OPT}_{\mathcal{A},K}(X),$$

which concludes the proof.

2.6.3 Proof of Theorem 2.7

We need the following additional lemma.

Lemma 2.26

Let $I = (k, X, \Omega, \mathcal{A}, K)$ be an instance of WCA CLUSTERING, $\epsilon \in (0, 1/2]$, $\delta \in [1, \infty)$, let $(\tilde{X}, \tilde{\Omega})$ be an $(\epsilon/3, \delta)$ -coreset for I , and let f denote its extension. Further, let $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$, let S_1, S_2 be two sets of k sites, and $\gamma \in \mathbb{R}$ with $\gamma \geq \delta$. Further, suppose that

$$\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S_1) \leq \gamma \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S_2).$$

Then,

$$\text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S_1) \leq (1 + \epsilon)\gamma \text{cost}_{\mathcal{A},K}(X, \Omega, S_2).$$

Proof. By the coreset properties, Definition 2.1 (a) and (b), we have

$$\left(1 - \frac{\epsilon}{3}\right) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S_1) - \Delta^+ \leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S_1)$$

and

$$\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S_2) \leq \left(1 + \frac{\epsilon}{3}\right) \text{cost}_{\mathcal{A},K}(X, \Omega, S_2) - \Delta^-.$$

Hence, it follows from the assumption that the right hand side of the former inequality is bounded from above by γ times the left hand side of the latter:

$$\left(1 - \frac{\epsilon}{3}\right) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S_1) - \Delta^+ \leq \left(1 + \frac{\epsilon}{3}\right) \gamma \text{cost}_{\mathcal{A},K}(X, \Omega, S_2) - \gamma \Delta^-.$$

Since $\gamma \geq \delta$, and $\Delta^+ \leq \delta \Delta^-$, we have $\Delta^+ \leq \delta \Delta^- \leq \gamma \Delta^-$. Hence,

$$\left(1 - \frac{\epsilon}{3}\right) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S_1) \leq \left(1 + \frac{\epsilon}{3}\right) \gamma \text{cost}_{\mathcal{A},K}(X, \Omega, S_2).$$

As

$$\left(1 + \frac{\epsilon}{3}\right) = 1 + \frac{2\epsilon}{3} - \frac{\epsilon}{3} \leq 1 + \frac{2\epsilon}{3} - \frac{\epsilon^2}{3} = (1 + \epsilon) \left(1 - \frac{\epsilon}{3}\right),$$

we obtain

$$\text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S_1) \leq (1 + \epsilon)\gamma \text{cost}_{\mathcal{A},K}(X, \Omega, S_2),$$

as asserted. $\square$

Proof (Proof of Theorem 2.7). (a) Since $\tilde{C}$ is a γ -approximation for $\tilde{I}_S$, we have

$$\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) \leq \gamma \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S).$$

Thus, Lemma 2.26, applied with $S_1 = S_2 = S$, implies that

$$\text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S) \leq (1 + \epsilon)\gamma \text{cost}_{\mathcal{A},K}(X, \Omega, S).$$

Hence $f(\tilde{C})$ is a $(1 + \epsilon)\gamma$ -approximation for I_S .

(b) Let $C^* \in \mathcal{C}_K(k, X, \Omega)$ be optimal and set $S^* = c(C^*)$. By Lemma 2.5 (b),

$$\text{OPT}_{\mathcal{A},K}(\tilde{X}) \leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S^*).$$

Hence, together with the assumption that $\tilde{C}$ is a γ -approximation for $\tilde{I}$, this yields

$$\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, c(\tilde{C})) = \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}) \leq \gamma \text{OPT}_{\mathcal{A},K}(\tilde{X}) \leq \gamma \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S^*).$$

Thus, Lemma 2.26, applied with $S_1 = c(\tilde{C})$ and $S_2 = S^*$, implies that

$$\text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), c(\tilde{C})) \leq (1 + \epsilon)\gamma \text{cost}_{\mathcal{A},K}(X, \Omega, S^*) = (1 + \epsilon)\gamma \text{OPT}_{\mathcal{A},K}(X).$$

With the aid of Lemma 2.5 (b), we conclude that

$$\text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C})) \leq \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), c(\tilde{C})) \leq (1 + \epsilon)\gamma \text{OPT}_{\mathcal{A},K}(X),$$

i.e., $f(\tilde{C})$ is a $(1 + \epsilon)\gamma$ -approximation for I . $\square$

2.6.4 Proof of Lemma 2.9

Just note that

$$\begin{aligned} \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|\tilde{x}_{p(j)} - x_j\|_{A_i}^2 &\leq \lambda^+(\mathcal{A}) \sum_{j=1}^n \left(\omega_j \|\tilde{x}_{p(j)} - x_j\|_2^2 \cdot \sum_{i=1}^k \xi_{ij} \right) \\ &= \lambda^+(\mathcal{A}) \sum_{j=1}^n \omega_j \|\tilde{x}_{p(j)} - x_j\|_2^2 \leq \eta \lambda^-(\mathcal{A}) \text{OPT}(X) \leq \eta \text{OPT}_{\mathcal{A}}(X). \end{aligned}$$

2.6.5 Proof of Lemma 2.10

First, we explain the equality using the definition of p :

$$\begin{aligned}
 \text{cost}_{\mathcal{A},K}(p(X, \Omega), p(C), S) &= \sum_{i=1}^k \sum_{\tilde{j}=1}^{\tilde{n}} \tilde{\xi}_{i\tilde{j}} \tilde{\omega}_{\tilde{j}} \|\tilde{x}_{\tilde{j}} - s_i\|_{A_i}^2 \\
 &= \sum_{i=1}^k \sum_{\tilde{j}=1}^{\tilde{n}} \left(\frac{1}{\tilde{\omega}_{\tilde{j}}} \sum_{j \in p^{-1}(\tilde{j})} \xi_{ij} \omega_j \right) \tilde{\omega}_{\tilde{j}} \|\tilde{x}_{\tilde{j}} - s_i\|_{A_i}^2 = \sum_{i=1}^k \sum_{\tilde{j}=1}^{\tilde{n}} \sum_{j \in p^{-1}(\tilde{j})} \xi_{ij} \omega_j \|\tilde{x}_{\tilde{j}} - s_i\|_{A_i}^2 \\
 &= \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|\tilde{x}_{p(j)} - s_i\|_{A_i}^2.
 \end{aligned}$$

Now, note that the latter norm decomposes as follows:

$$\begin{aligned}
 \|\tilde{x}_{p(j)} - s_i\|_{A_i}^2 &= \|\tilde{x}_{p(j)} - x_j + x_j - s_i\|_{A_i}^2 \\
 &= \|\tilde{x}_{p(j)} - x_j\|_{A_i}^2 + 2(\tilde{x}_{p(j)} - x_j)^\top A_i(x_j - s_i) + \|x_j - s_i\|_{A_i}^2.
 \end{aligned}$$

Hence, it suffices to bound the three terms

$$D, \quad \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j (\tilde{x}_{p(j)} - x_j)^\top A_i(x_j - s_i), \quad \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{A_i}^2$$

separately. The first term appears in the bound from above and, as it is nonnegative, it can be omitted for the estimate from below. Also, the third sum is exactly $\text{cost}_{\mathcal{A},K}(X, \Omega, C, S)$. This suffices to suitably bound the absolute value of the second sum. To do so, we evoke the Cauchy-Schwarz inequality twice and obtain

$$\begin{aligned}
 \left| \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j (\tilde{x}_{p(j)} - x_j)^\top A_i(x_j - s_i) \right| &\leq \left| \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|\tilde{x}_{p(j)} - x_j\|_{A_i} \|x_j - s_i\|_{A_i} \right| \\
 &= \left| \sum_{i=1}^k \sum_{j=1}^n \sqrt{\xi_{ij} \omega_j} \|\tilde{x}_{p(j)} - x_j\|_{A_i} \sqrt{\xi_{ij} \omega_j} \|x_j - s_i\|_{A_i} \right| \leq \sqrt{D} \cdot \sqrt{\text{cost}_{\mathcal{A},K}(X, \Omega, C, S)}.
 \end{aligned}$$

This concludes the proof.

2.6.6 Proof of Theorem 2.11

We show that, under the assumptions of the theorem, the coreset properties, Definition 2.1 (a), (b), hold with offset 0, i.e., for $\Delta^+ = \Delta^- = 0$.

Since $\text{OPT}_{\mathcal{A}}(X) \leq \text{cost}_{\mathcal{A},K}(X, \Omega, C, S)$ and, by assumption,

$$\sum_{j=1}^n \omega_j \|x_j - \tilde{x}_{p(j)}\|_2^2 \leq \frac{\epsilon^2 \lambda^-(\mathcal{A})}{16 \lambda^+(\mathcal{A})} \text{OPT}(X),$$

Lemma 2.9, applied with $\eta = \epsilon^2/16$, provides

$$D = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|\tilde{x}_{p(j)} - x_j\|_{A_i}^2 \leq \frac{\epsilon^2}{16} \text{OPT}_{\mathcal{A}}(X) \leq \frac{\epsilon^2}{16} \text{cost}_{\mathcal{A},K}(X, \Omega, C, S).$$

Thus, Lemma 2.10 yields for $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ and $C = f(\tilde{C})$

$$\begin{aligned} (1 - \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, C, S) &\leq \text{cost}_{\mathcal{A},K}(X, \Omega, C, S) - 2\sqrt{D \cdot \text{cost}_{\mathcal{A},K}(X, \Omega, C, S)} \\ &\leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S), \end{aligned}$$

which shows coreset property (a).

To verify coreset property (b), let $C \in \mathcal{C}_K(k, X, \Omega)$ be optimal for the given sites S , and set $\tilde{C} = p(C)$. Then, by Lemma 2.10

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S) &\leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) \\ &\leq \text{cost}_{\mathcal{A},K}(X, \Omega, C, S) + 2\sqrt{D \cdot \text{cost}_{\mathcal{A},K}(X, \Omega, C, S)} + D \\ &\leq \left(1 + \frac{\epsilon}{2} + \frac{\epsilon^2}{16}\right) \text{cost}_{\mathcal{A},K}(X, \Omega, S) \leq (1 + \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, S). \end{aligned}$$

This concludes the proof.

2.6.7 Proof of Lemma 2.14

Let $j \in [n]$ and $i(j) \in [\hat{k}]$ such that $x_j \in X \cap \hat{C}_{i(j)}$. Further, let $q \in Q$ such that $\bar{x}_j \in L(\hat{c}_{i(j)}, q)$. Using the intercept theorem, we see that with $i = i(j)$,

$$\|x_j - \bar{x}_j\|_2 \leq \left\| \left(\hat{c}_i + \frac{x_j - \hat{c}_i}{\|x_j - \hat{c}_i\|_2} \right) - (\hat{c}_i + q) \right\|_2 \cdot \|x_j - \hat{c}_i\|_2 \leq \epsilon_0 \cdot \|x_j - \hat{c}_i\|_2.$$

Since the $\hat{k}$ clusters $\hat{C}_i$ constitute an (α, β) -approximation for least-squares clustering, this implies

$$\begin{aligned} \sum_{j=1}^n \omega_j \|x_j - \bar{x}_j\|_2^2 &\leq \epsilon_0^2 \cdot \sum_{j=1}^n \omega_j \|x_j - \hat{c}_{i(j)}\|_2^2 = \epsilon_0^2 \cdot \text{ALG}_{(\alpha, \beta)}(X) \leq \epsilon_0^2 \cdot \alpha \cdot \text{OPT}(X) \\ &= \frac{\epsilon^2 \lambda^-(\mathcal{A})}{16 \lambda^+(\mathcal{A})} \text{OPT}(X). \end{aligned}$$

Hence, it follows from Theorem 2.11 that $(\bar{X}, \Omega)$ is an ϵ -coreset for WCA CLUSTERING.

2.6.8 Proof of Lemma 2.15

Let f_1 and f_2 denote the extensions for $(\bar{X}, \bar{\Omega})$ and $(\tilde{X}, \tilde{\Omega})$, and set $f = f_1 \circ f_2$. Further, for $\ell \in [2]$, let $\Delta_\ell^\pm$ denote the corresponding Δ -terms, and set $\Delta^\pm = \Delta_1^\pm + \Delta_2^\pm$. Since $0 \leq \Delta_\ell^+ \leq \delta_\ell \Delta_\ell^-$ for $\ell \in [2]$, we have

$$0 \leq \Delta^+ = \Delta_1^+ + \Delta_2^+ \leq \delta_1 \Delta_1^- + \delta_2 \Delta_2^- \leq \delta \Delta^-.$$

Also, note that,

$$1 - \epsilon = 1 - (\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2) \leq 1 - (\epsilon_1 + \epsilon_2 - \epsilon_1\epsilon_2) = (1 - \epsilon_1)(1 - \epsilon_2).$$

We will now verify the two conditions of Definition 2.1 (a) and (b). So, let S be a set of k sites, and let $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$. Then, by the coreset conditions for $(\tilde{X}, \tilde{\Omega})$ and $(\tilde{X}, \tilde{\Omega})$, we have

$$\begin{aligned} (1 - \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S) &\leq (1 - \epsilon_1)(1 - \epsilon_2) \text{cost}_{\mathcal{A},K}(X, \Omega, f_1(f_2(\tilde{C})), S) \\ &\leq (1 - \epsilon_2) (\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, f_2(\tilde{C}), S) + \Delta_1^+) \\ &\leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta_2^+ + (1 - \epsilon_2)\Delta_1^+ \\ &\leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta^+. \end{aligned}$$

Further,

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S) + \Delta^- &\leq (1 + \epsilon_2) \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S) + \Delta_1^- \\ &\leq (1 + \epsilon_2) (\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S) + \Delta_1^-) \\ &\leq (1 + \epsilon_2)(1 + \epsilon_1) \text{cost}_{\mathcal{A},K}(X, \Omega, S) = (1 + \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, S). \end{aligned}$$

Hence, $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -coreset for I .

2.6.9 Proof of Lemma 2.21

By Remark 2.4 and since $\sum_{i=1}^k \xi_{iB} = 1$ for each $B \in \mathcal{B}'$, we obtain

$$\sum_{i=1}^k \sum_{B \in \mathcal{B}'} \xi_{iB} \mathcal{V}_{A_i}(B) \leq \lambda^+(\mathcal{A}) \sum_{i=1}^k \sum_{B \in \mathcal{B}'} \xi_{iB} \mathcal{V}_E(B) = \lambda^+(\mathcal{A}) \sum_{B \in \mathcal{B}'} \mathcal{V}_E(B)$$

and

$$\sum_{i=1}^k \sum_{B \in \mathcal{B}'} \xi_{iB} \mathcal{V}_{A_i}(B) \geq \lambda^-(\mathcal{A}) \sum_{i=1}^k \sum_{B \in \mathcal{B}'} \xi_{iB} \mathcal{V}_E(B) = \lambda^-(\mathcal{A}) \sum_{B \in \mathcal{B}'} \mathcal{V}_E(B).$$

2.6.10 Proof of Lemma 2.22

First note that, for each $B \in \mathcal{B}$ we have by Lemma 2.5 (a) (with $X_0 = B$, $c_0 = c_B$)

$$\omega_B \|x_B - s_i\|_{A_i}^2 = \sum_{\bar{x}_j \in B} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2 - \mathcal{V}_{A_i}(B).$$

Therefore,

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}, \Omega_{\mathcal{B}}, C_{\mathcal{B}}, S) &= \sum_{i=1}^k \sum_{B \in \mathcal{B}} \xi_{iB} \omega_B \|x_B - s_i\|_{A_i}^2 \\ &= \sum_{i=1}^k \sum_{B \in \mathcal{B}} \left(\sum_{\bar{x}_j \in B} \xi_{iB} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2 - \xi_{iB} \mathcal{V}_{A_i}(B) \right). \end{aligned}$$

Since $\bar{\xi}_{ij} = \xi_{iB}$ for each $\bar{x}_j \in B$ we have

$$\begin{aligned} \sum_{i=1}^k \sum_{B \in \mathcal{B}} \sum_{\bar{x}_j \in B} \xi_{iB} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2 &= \sum_{i=1}^k \sum_{B \in \mathcal{B}} \sum_{\bar{x}_j \in B} \bar{\xi}_{ij} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2 \\ &= \sum_{i=1}^k \sum_{j=1}^n \bar{\xi}_{ij} \omega_j \|\bar{x}_j - s_i\|_{A_i}^2 = \text{cost}_{\mathcal{A},K}(X, \Omega, f_2(C_{\mathcal{B}}), S). \end{aligned}$$

Hence,

$$\text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}, \Omega_{\mathcal{B}}, , C_{\mathcal{B}}, S) = \text{cost}_{\mathcal{A},K}(X, \Omega, f_2(C_{\mathcal{B}}), S) - \sum_{i=1}^k \sum_{B \in \mathcal{B}} \xi_{iB} \mathcal{V}_{A_i}(B),$$

and, by Lemma 2.21, this yields

$$\text{cost}_{\mathcal{A},K}(X, \Omega, f_2(C_{\mathcal{B}}), S) \leq \text{cost}_{\mathcal{A},K}(X_{\mathcal{B}}, \Omega_{\mathcal{B}}, , C_{\mathcal{B}}, S) + \lambda^+(\mathcal{A}) \sum_{B \in \mathcal{B}} \mathcal{V}_E(B),$$

which is the asserted inequality.

Chapter 3

Resolution Coresets

Specific data compression techniques, formalized by the concept of coresets, proved to be powerful for many optimization problems. In fact, while tightly controlling the approximation error, coresets may lead to significant speed up of the computations and hence allow to extend algorithms to much larger problem sizes. The chapter deals with a weight-balanced clustering problem, and is specifically motivated by an application in materials science where a voxel-based image is to be processed into a diagram representation. Here, the class of desired coresets is naturally confined to those which can be viewed as lowering the resolution of the input data. While one might expect that such resolution coresets are inferior to unrestricted coreset we prove bounds for resolution coresets which improve known bounds in the relevant dimensions and also lead to significantly faster algorithms in practice.

3.1 Introduction

Many challenging practical problems require the solution of weight-balanced clustering problems on huge data sets. Fields of applications include agriculture (consolidation of farmland, see e.g. [BBG14]), transportation (air cargo prediction, [BG20]), political sciences (electoral districting, [BGK17]) or material sciences [Alp+15]. An imaging problem in the latter domain motivates this chapter. The goal of solving this problem is to obtain concise representations of polycrystalline materials in terms of “best-fitting” anisotropic power diagrams. More specifically, [Alp+15] suggested to compute a best clustering of a set of points which correspond to the voxels of a sample at a given resolution according to certain constraints and a suitable objective function and use a structural duality result from [BG12; Kle20] to obtain a desired diagram. As various independent studies, including [Šed+16; Šed+17; Spe+16], indicate the so called GBPD-algorithm of [Alp+15] seems to capture the physical principles of forming polycrystals quite well. It is, however, rather slow, so that dynamic simulations of grain growth at a small temporal resolution quickly drive the method over the limit.

As experimentally verified in [Alp+23b], significant acceleration can be obtained by heavily compressing the original data set without losing too much accuracy. This is made precise by the notion of coresets which have been developed for various kinds of imaging and clustering problems; see, in particular, [BLK17; FSS13; Fic+13; FG22a; HPK07; SW18]. Such “general-purpose coresets” work on arbitrary finite data sets but typically ignore the special voxel-based Cartesian structure which is present in many image processing applications.

In the imaging problem from materials science such a structure can be viewed as encoding the given image at a lower resolution. The basic underlying theoretical question is just how “sparse” such a Cartesian data set can be made without losing too much accuracy. This is

highly relevant in practice since polycrystalline images at lower resolution are much easier to obtain than those of high resolution. It is, however not clear what to expect from such *resolution coresets*. On the one hand, the strong restriction to a Cartesian structure might render such resolution coresets inferior to unrestricted coresets. On the other hand, the specific structure of the original data set might allow for stronger bounds.

We prove that resolution coresets are indeed smaller in the relevant dimensions. In particular, if k denotes the number of grains and ϵ is the allowed error, then an order of

$$\frac{k^2}{\epsilon^{4/3}}, \quad \text{and} \quad \frac{k^3}{\epsilon^2}$$

points suffices to guarantee the desired accuracy in dimensions 2 and 3, respectively. Phrased differently, this means that even when applied to data whose size is independent of the original image resolution, the clustering techniques will still produce near-optimal weight-constrained clusterings. Precise formulations of our main result (for general dimension) and a comparison to known coreset bounds will be given in Section 3.2. Let us only mention here that our bounds on the size of resolution coresets prove that the given data can be safely thinned out beyond what has been known before, and hence, put experimental observations of [Alp+23b] on a firm theoretical ground.

This text is organized as follows. First, Section 3.2 formally introduces the problems and states our main results. Section 3.3 describes relevant concepts and provides some required preliminaries. Sections 3.4, 3.5, and 3.6 then study resolution coresets in detail and prove our main results. Section 3.7 concludes with some final remarks.

3.2 Motivation, notation, and main results

We will now provide the relevant general notation, introduce the problems and state our main results. While the developed techniques can be applied to a variety of imaging problems, we begin with a brief description of our main “motivating application” of grain mapping. Further details will follow later.

Grain map representation

Polycrystalline materials can be viewed as dissections of the occupied spatial region in $\mathbb{R}^3$ into monocrystalline cells called *grains*. (For some such applications 2D-imaging is relevant too.) While such materials can be accessed by *grain maps* or scans, i.e., voxel-based discrete images at some resolution, their analysis and processing is much more easily facilitated by *diagram representations* which are governed by only a few parameters. In the underlying *anisotropic model* these parameters correspond to information about the location, shape and size of each of the k grains $G_1, \dots, G_k$.

More precisely, for $i \in [k] = \{1, \dots, k\}$, let $s_i \in \mathbb{R}^3$ be a given *site*, let $A_i \in \mathbb{R}^{3 \times 3}$ be a positive definite symmetric matrix, and let $\gamma_i \in \mathbb{R}$. The site s_i can be viewed as the “nuclei” of the i th grain G_i , whereas γ_i is a *size* parameter. The matrix A_i encodes information on the “growth directions” of G_i in terms of the main axes of the ellipsoid $\{x \in \mathbb{R}^3 : x^T A_i x \leq 1\}$. Then, the i th cell P_i of the *diagram* is given by

$$P_i = \{x \in \mathbb{R}^3 : (x - s_i)^T A_i (x - s_i) + \gamma_i \leq (x - s_\ell)^T A_\ell (x - s_\ell) + \gamma_\ell\}.$$

Naturally, one is interested in diagrams $\mathcal{P} = (P_1, \dots, P_k)$ whose cells are “close” to the grains G_i .

Suppose now that a grain scan image is given. If we identify the voxels by their centroids, we can view it as an assignment of the corresponding grid points to the grains. Appropriate sites s_i and matrices A_i can be computed from grain scans, and also the volume of G_i is available in terms of the number of its grid points. (These “ingredients” are often already directly available from tomographic measurements; see [Alp+15; Alp+23b].) Hence the task reduces to the computation of the sizes $\gamma_1, \dots, \gamma_k$. Using linear programming duality, [Alp+15] compute the sizes by means of a constrained clustering problem on the grid points. A natural question is how low an initial resolution can be chosen while still keeping a high accuracy of representation. As the underlying resolution determines the dimension of the clustering problem, this is of utmost importance for practical computations as it may determine whether desired clusterings and hence diagram representations can actually be computed in practice.

Weight-constrained clustering

Next, we introduce the constrained clustering problem more formally. While we are specifically interested in (unweighted) Cartesian point sets, the following description is more general both, in terms of the ground set, and the associated weights. The reason is that we will later aggregate batches of points to single points in various ways which carry the number of aggregated points as their weights.

Given $x_1, \dots, x_n \in \mathbb{R}^d$ and associated weights $\omega_1, \dots, \omega_n \in (0, \infty)$, let (X, Ω) denote the weighted data set of the tuples (x_j, ω_j) for $j \in [n]$. Further, let $k \in \mathbb{N} \setminus \{1\}$.

A k -clustering (or, if k is clear, simply a clustering) C of (X, Ω) is a vector

$$C = (C_1, \dots, C_k) = (\xi_{11}, \dots, \xi_{1n}, \dots, \xi_{k1}, \dots, \xi_{kn}) \in [0, 1]^{kn}$$

whose components ξ_{ij} specify the fraction of x_j that is assigned to the i th cluster $C_i = (\xi_{i1}, \dots, \xi_{in})$. The set of all such clusterings will be denoted by $\mathcal{C}(k, X, \Omega)$. If all ξ_{ij} are integer, i.e., $\xi_{ij} \in \{0, 1\}$, the clustering is called *integer*.

Suppose that approximations $\kappa_1, \dots, \kappa_k$ of the desired volumes of the cells are given; collected in the family $K = \{\kappa_1, \dots, \kappa_k\}$. Then, we also demand that the *weight*

$$\omega(C_i) = \sum_{j=1}^n \xi_{ij} \omega_j$$

of each cluster C_i is close to κ_i . More precisely, for given $\epsilon_i \in (0, \infty)$, we require that the weak weight-constraints

$$(1 - \epsilon_i) \kappa_i \leq \omega(C_i) \leq (1 + \epsilon_i) \kappa_i \quad (i \in [k])$$

hold. In the specific situation of grain mapping on the original grid all point weights are identical, and the cluster weights of all integer clusterings and also the target values κ_i are integer multiples of that weight ν . Hence with measurement errors $\epsilon_i < \nu/2\kappa_i$ the weak weight-constraints are equivalent to the strong *weight-constraints*

$$\omega(C_i) = \kappa_i \quad (i \in [k])$$

As the measurement errors are generally determined by the initial resolution anyway we will in the following (just for the simplicity of exposition) always require this stronger condition.

The set of all such *weight-constrained clusterings* will be denoted by $\mathcal{C}_K(k, X, \Omega)$. Note that, $\mathcal{C}_K(k, X, \Omega) \neq \emptyset$ if and only if

$$\sum_{i=1}^k \kappa_i = \sum_{j=1}^n \omega_j,$$

and we will always assume that this is the case. As it is well-known (see e.g. [BG12]), when all weights are 1 or, equivalently, are identical, there always exist optimal clusterings $C \in \mathcal{C}_K(k, X, \Omega)$ which are integer.

The quality of a clustering will be measured in terms of deviations from k different *sites* $s_1, \dots, s_k$, with respect to *ellipsoidal norms*. More precisely, for $i \in [k]$, let $A_i \in \mathbb{R}^{d \times d}$ be a positive definite symmetric matrix, and let $\|\cdot\|_{A_i}$ denote the associated norm, i.e. $\|x\|_{A_i} = \sqrt{x^T A_i x}$ for any $x \in \mathbb{R}^d$. Further, we set $S = \{s_1, \dots, s_k\}$ and $\mathcal{A} = \{A_1, \dots, A_k\}$. Let us point out that $\mathcal{A}$ is a family, i.e., repetitions are allowed.

The *cost* of a clustering C with respect to S and $\mathcal{A}$ is defined as

$$\text{cost}_{\mathcal{A}, K}(X, \Omega, C, S) = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{A_i}^2.$$

A case of special relevance is that with all matrices being the unit matrix E_d . Then $\mathcal{E} = \{E_d, \dots, E_d\}$ refers to the classic Euclidean *least-squares* objective, in which each cluster norm is $\|x\|_2 = \|x\|_{E_d}$. In the realm of grain mapping, this is called the *isotropic case* while the general situation models the *anisotropic case*. In the isotropic situation we drop the subscript and simply write $\text{cost}(X, \Omega, C, S)$ rather than $\text{cost}_{\mathcal{E}}(X, \Omega, C, S)$.

If no sites S are given, we measure the cost of a clustering C with respect to the “best sites” (see Remark 3.9), i.e., the cluster *centroids*

$$c_i = \frac{1}{\omega(C_i)} \sum_{j=1}^n \xi_{ij} \omega_j x_j,$$

and define

$$\text{cost}_{\mathcal{A}, K}(X, \Omega, C) = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - c_i\|_{A_i}^2.$$

Grain volumes K , sites S and matrices $\mathcal{A}$ can often be measured directly (with sites approximating the centroids). Different norms then reflect measured knowledge about the moments of the grains in materials, see [Alp+15]. In the case that grain scans are explicitly available, the volumes K can be derived directly, sites can be computed as centroids, and the matrices can be estimated as the inverse of covariance matrices. However, both, S and $\mathcal{A}$ may also be subject to optimization to reduce the classification error; see [Alp+23b]. Here we are dealing with the most relevant situation, which leads to the mathematical problems of finding optimal weight-constrained anisotropic assignments or clusterings:

WCA ASSIGNMENT:

Given $k, X, \Omega, \mathcal{A}, K, S$, find $C^* \in \mathcal{C}_K(k, X, \Omega)$ that minimizes $\text{cost}_{\mathcal{A}, K}(X, \Omega, C, S)$.

WCA CLUSTERING:

Given $k, X, \Omega, \mathcal{A}, K$, find $C^* \in \mathcal{C}_K(k, X, \Omega)$ that minimizes $\text{cost}_{\mathcal{A}, K}(X, \Omega, C)$.

We say that $(k, X, \Omega, \mathcal{A}, K, S)$ or $(k, X, \Omega, \mathcal{A}, K)$ is an instance of WCA ASSIGNMENT or WCA CLUSTERING, respectively. If we are in the Euclidean case, i.e., when $\mathcal{A} = \mathcal{E} = \{E_d, \dots, E_d\}$, we just speak of *weight-constrained clustering* or *assignment* and drop $\mathcal{A}$ from the instance.

Coresets

The notion of a coreset is used to classify the effect of data reductions on the quality of solutions of hard optimization problems. As it is well-known, even unconstrained least-squares clustering is NP-hard, [Alo+09; Das07], and the NP-hardness persists even if the dimension is fixed to $d = 2$, [MNV12]. Also, by [Awa+15], no PTAS in both, d and k , exists. Of course, when S is fixed, the computation of

$$\text{cost}_{\mathcal{A},K}(X, \Omega, S) = \min_{C \in \mathcal{C}_K(k, X, \Omega)} \text{cost}_{\mathcal{A},K}(X, \Omega, C, S)$$

boils down to a linear program and can hence be carried out in polynomial time. But even then, typical real-world instances in materials science are so large that they can be solved only on very sparse grids. Therefore, one is aiming at a significant speed up by resorting to a “compressed” set $(\tilde{X}, \tilde{\Omega})$ without sacrificing much of the clustering quality.

For this reason (various) notions of coresets have been introduced, see e.g. [BLK17; FSS13; FSS20; Fic+13; HPK07; SW18]. In order to be able to state Proposition 3.2 precisely, we provide a quite weak notion introduced in [FG22a] which is still sufficient for deriving good weight-constrained anisotropic clusterings on the original data set from those on coresets. We will, however, also strengthen the definition as our main result will yield coresets according to the stronger notion.

Definition 3.1

Let $\epsilon \in (0, 1/2]$ and $\delta \in [1, \infty)$. Then $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -coreset for (k, X, Ω, K) if there exists a mapping $f : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, X, \Omega)$, called *extension*, and real constants Δ^+, Δ^- , referred to as Δ -terms or *offsets*, with $0 \leq \Delta^+ \leq \delta \cdot \Delta^-$ such that the following two conditions hold for all sets S of k sites and clusterings $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$:

- (a) $(1 - \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}), S) \leq \text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta^+,$
- (b) $\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, S) + \Delta^- \leq (1 + \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, S).$

If the two Δ -terms coincide, i.e. $0 \leq \Delta^+ = \Delta^-$, we can choose $\delta = 1$, and we speak of an ϵ -coreset then.

Note that, in the latter situation, for $\tilde{C}^* \in \text{argmin}\{\text{cost}_{\mathcal{A},K}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) : \tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})\}$,

$$(1 - \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, S) \leq (1 - \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, f(\tilde{C}^*), S) \leq (1 + \epsilon) \text{cost}_{\mathcal{A},K}(X, \Omega, S).$$

Hence, if the extension g can be computed efficiently, solving the assignment problem on the coreset $(\tilde{X}, \tilde{\Omega})$ leads to an approximate solution on the original data set (X, Ω) . It is less obvious but still true that the above general definition of coresets indeed captures the intuition behind the concept that an approximate solution of the clustering problem on the coresets leads to an approximate solution on the original data set (X, Ω) ; see [FG22a, Thm. 3.5] and Proposition 3.28.

Coresets for unconstrained least-squares clustering have been studied intensively, see [BLK17; Bra+16; Coh+15; FSS13; FSS20; Fic+13; HPK07; SW18]. In particular, [HPK07] constructed coresets of size $\mathcal{O}\left(\frac{k^3}{\epsilon^{d+1}}\right)$ for unconstrained least-squares clustering which “live” on $\mathcal{O}(k)$ pencils of lines through well-chosen points. Each pencil has a single point as its vertex and consists of the union of finitely many lines through this point. The lines are chosen in such a way that their points of intersection with the unit sphere centered at the vertex are “evenly” and (depending on the desired error term ϵ) “sufficiently densely distributed”.

Later, [FG22a] improved and generalized [HPK07]’s construction and analysis, leading to *pencil coresets* of size only $\mathcal{O}\left(\frac{k^2}{\epsilon^{d+1}}\right)$ even for weight-constrained anisotropic clustering. More precisely, let $\lambda^+(\mathcal{A})$ and $\lambda^-(\mathcal{A})$ denote the largest and smallest eigenvalue of all matrices in $\mathcal{A}$, respectively. Then the following result holds.

Proposition 3.2 ([FG22a, Thm. 2.3])

For any instance $(k, X, \Omega, K, \mathcal{A})$ of WCA CLUSTERING, $\epsilon \in (0, 1/2]$ and

$$\delta \geq \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})},$$

there is an (ϵ, δ) -coreset $(\tilde{X}, \tilde{\Omega})$ with

$$|\tilde{X}| \in \mathcal{O}\left(\frac{k^2}{\epsilon^{d+1}}\right).$$

Note that the coresets according to Proposition 3.2 are independent of the number of points in the original data set but depend on the dimension d . Let us point out, however, that [FSS13] provides a method which often allows to drop the dependency of coreset sizes from the dimension of space. Corresponding techniques for clustering, [Bra+16; Coh+15; FSS13; FSS20; SSS20] yield coresets for unconstrained least-squares clustering which only depend on k and ϵ . While some of these techniques generalize to the weight-constrained case, it would be theoretically intriguing to extend such results also to anisotropic clustering. However, this will be largely irrelevant for the imaging applications we have in mind, particularly when they come with higher order dependencies on k and ϵ . In fact, our motivating imaging problems “live” in dimensions 2 and 3, anyway, and our focus is on “absolutely small” coresets rather than on favorable asymptotic behavior in d .

While Proposition 3.2 gives the currently best bound for (deterministic) coresets for WCA CLUSTERING, in terms of k and ϵ it has, however, still some severe drawbacks which limit its use for image segmentation like grain mapping. First, the available coresets may still be very large. For instance, for $k = 100$, $d = 3$, and $\epsilon = 0.001$, we have

$$\frac{k^2}{\epsilon^{d+1}} = \frac{10^4}{10^{-12}} = 10^{16}.$$

Also, note that the asymptotic $\mathcal{O}$ -statement for the size of $\tilde{X}$ conceals the constant which is also relevant. As another crucial downside, the constructed pencil coresets completely ignore the grid structure that is present in many image segmentation problems including our motivating application of grain mapping.

Resolution coresets

We will now provide the relevant notions for stating and proving our main result on the existence of small coresets which carry a grid structure.

While we are mainly interested in pixel- or voxel-based images, i.e., in the dimensions $d = 2, 3$, we will formulate our results in arbitrary dimensions. Hence we will use *voxel* as generic term, independently of the dimension $d \in \mathbb{N}$ of space. Typically, we consider an image in $[0, 1]^d$. Each axis is partitioned into intervals of equal length, and each voxel is the Cartesian product of one such interval on each axis.

Formally, let $\boldsymbol{\rho} = (\rho_1, \dots, \rho_d) \subset \mathbb{N}^d$ be a vector. We consider the partitioning of the d -dimensional cube $[0, 1]^d$ into uniform boxes of size $2^{-\rho_1} \times \dots \times 2^{-\rho_d}$. Then, we obtain a data set $X(\boldsymbol{\rho})$ (which will later be clustered into sets which represent the gains) by replacing each box by its centroid, i.e., $X(\boldsymbol{\rho})$ consists of all points

$$x = x_{j_1, \dots, j_d} = \begin{pmatrix} \frac{1}{2^{\rho_1+1}} + \frac{j_1-1}{2^{\rho_1}} \\ \vdots \\ \frac{1}{2^{\rho_d+1}} + \frac{j_d-1}{2^{\rho_d}} \end{pmatrix} = \begin{pmatrix} \frac{-1}{2^{\rho_1+1}} + \frac{j_1}{2^{\rho_1}} \\ \vdots \\ \frac{-1}{2^{\rho_d+1}} + \frac{j_d}{2^{\rho_d}} \end{pmatrix} \quad (j_1 \in [2^{\rho_1}], \dots, j_d \in [2^{\rho_d}]).$$

Note that

$$|X(\boldsymbol{\rho})| = 2^{\rho_1} \cdot \dots \cdot 2^{\rho_d}.$$

Of course, the voxels of the image are the boxes

$$x_{j_1, \dots, j_d} + \frac{1}{2^{\rho_1+1}}[-1, 1] \times \dots \times \frac{1}{2^{\rho_d+1}}[-1, 1]$$

of volume

$$\nu = \nu(\boldsymbol{\rho}) = \frac{1}{2^{\rho_1}} \cdot \dots \cdot \frac{1}{2^{\rho_d}}.$$

The spatial resolution of such a discretization of $[0, 1]^d$ in each coordinate i is $2^{-\rho_i}$. For simplicity, we speak of the resolution $\boldsymbol{\rho}$. In the following, we will refer to point sets of the form $X(\boldsymbol{\rho})$ as *Cartesian point sets*. Further, let

$$\omega_1 = \dots = \omega_n = \nu(\boldsymbol{\rho}) = \nu, \quad \Omega(\boldsymbol{\rho}) = \{\omega_1, \dots, \omega_n\}.$$

Then $(X(\boldsymbol{\rho}), \Omega(\boldsymbol{\rho}))$ will be called *resolution set*.

We will now adopt the following convention: $\boldsymbol{\rho}$ is the given fixed *benchmark resolution*, and

$$X = X(\boldsymbol{\rho}) = \{x_1, \dots, x_n\}, \quad \Omega = \Omega(\boldsymbol{\rho}), \quad \kappa_1, \dots, \kappa_k \in \nu\mathbb{N}, \quad n\nu = \sum_{i=1}^k \kappa_i$$

is the given *resolution setting*. We are then interested in coresets $(\tilde{X}, \tilde{\Omega})$ which carry the same structure, i.e.,

$$\tilde{X} = X(\boldsymbol{\tau}) = \{\tilde{x}_1, \dots, \tilde{x}_{\tilde{n}}\}, \quad \tilde{\Omega} = \tilde{\Omega}(\boldsymbol{\tau}) = \{\tilde{\omega}_1, \dots, \tilde{\omega}_{\tilde{n}}\}$$

with

$$\tilde{\omega}_1 = \dots = \tilde{\omega}_{\tilde{n}} = \nu(\boldsymbol{\tau}) = 2^{-\tau_1} \cdot \dots \cdot 2^{-\tau_d}$$

for some coarser resolution τ with $\tau \leq \rho$ (inequalities involving vectors are always meant componentwise).

Also, due to the underlying Cartesian product structure, we need to restrict our considerations to the Euclidean case, i.e., $\mathcal{A} = \mathcal{E}$. Theorem 3.5 will, however, later show that results for this isotropic setting also yield approximation results for the anisotropic case.

The question how far we can reduce τ while maintaining the coreset properties leads to the definition of a resolution coreset.

Definition 3.3

Let $\epsilon \in (0, 1/2]$ and $\delta \in [1, \infty)$. Then $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -resolution coreset if $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -coreset and a resolution set, i.e., there exists a resolution τ such that $\tilde{X} = \tilde{X}(\tau)$ and $\tilde{\Omega} = \tilde{\Omega}(\tau)$. If the resolution set $(\tilde{X}, \tilde{\Omega})$ is an ϵ -coreset we speak of an ϵ -resolution coreset.

Main results

We show that the drawbacks of the more general result of Proposition 3.2 can be avoided in the resolution setting at least in small dimensions (those which are relevant for grain mapping and other related image segmentation problems).

Theorem 3.4

Let $\rho \in \mathbb{N}^d$ $X = X(\rho)$, $\Omega = \Omega(\rho)$, and $\epsilon \in (0, 1/2]$. Then there is an ϵ -resolution coreset $(\tilde{X}, \tilde{\Omega})$ for (k, X, Ω, K) of size

$$|\tilde{X}| \leq 2^{8d/3} \left(\frac{k}{\epsilon^{2/3}} \right)^d.$$

Note that the size of the resolution coreset in Theorem 3.4 is independent of ρ . Thus the result shows that a higher resolution is often not needed to achieve a good clustering of image voxels. In fact, using a grid with less than $7k/\epsilon^{2/3}$ points on each coordinate axis results in a clustering whose cost is at most off the optimum by a factor of $1 + \epsilon$. In the realm of our motivating application from materials science, Theorem 3.4 allows to scan grains at a relatively low resolution without sacrificing much of the resulting quality and can hence be used to speed up scanning based imaging series of grain growth dynamics in practice. Even more importantly, Theorem 3.4 can also be used for computing grain maps from few of their characteristic measurements in practise even in large 3D settings which are far beyond the reach of [Alp+15]’s original GBPD-algorithm.

Let us now compare the size of the resolution coreset to the bound given in Proposition 3.2. Beginning with the asymptotics (and ignoring multiplicative constants), the bounds read

$$\frac{k^2}{\epsilon^{d+1}} \quad (\text{Proposition 3.2}), \quad \frac{k^d}{\epsilon^{2d/3}} \quad (\text{Theorem 3.4}).$$

Hence, the former is better in k for $d \geq 3$ but always worse in ϵ .

In dimension $d = 3$, the latter is better by a factor of $\epsilon^2 k$. Specifically, for values $k = 100$ and $\epsilon = 0.001$, and when compared to the evaluation after Proposition 3.2, we save a factor of 10^4 . Note that the bound for resolution coresets involves an additional constant of 256, while the $\mathcal{O}$ -notation in Proposition 3.2 hides the constant present in the construction of the required number of lines; see [HPK07; Mat02] for the relevant notion and result on ϵ -nets. Such

constructions typically involve large constants (which grow exponential in d); see e.g. [Bri+98, Lemma 5.3].

Let us point out that, as we will see, the construction of Theorem 3.4 is explicit, including the extension g . Hence, from a constrained clustering $\tilde{C}$ on the low resolution data set we obtain quickly a clustering $g(\tilde{C})$ on the original data set. Let us finally point out that this is not just useful in the isotropic case but also for anisotropic clusterings. The following theorem uses the standard notion of a γ -approximation for some $\gamma \geq 1$ which means that the cost of the produced clustering is off by at most a factor of γ from the optimum, i.e., is bounded from above by γ times the cost of an optimal clustering.

Theorem 3.5

Let $I(\mathcal{A}) = (k, X(\rho), \Omega(\rho), \mathcal{A}, K, S)$ be an instance of WCA ASSIGNMENT, $\epsilon \in (0, 1/2]$, and $I(\mathcal{E}) = (k, X(\rho), \Omega(\rho), \mathcal{E}, K, S)$. Further, let $(\tilde{X}, \tilde{\Omega})$ be an $\epsilon/3$ -resolution coreset for $I(\mathcal{E})$, f its extension, $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$, $C = g(\tilde{C})$ and $\gamma \in [0, \infty)$.

If $\tilde{C}$ is a γ -approximation for $\tilde{I}(\mathcal{E}) = (k, \tilde{X}, \tilde{\Omega}, \mathcal{E}, K, S)$, then C is a $(1 + \epsilon)\gamma^{\frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})}}$ -approximation for $I(\mathcal{A})$.

3.3 Some basics for the analysis of resolution coresets

In the following we will prepare the proof of our main result. We will outline its structure, collect some prerequisites, and derive some first basic estimates.

Structure of the proof of Theorem 3.4

The proof of Theorem 3.4 will rely heavily on the specific underlying resolution setting. In particular, all three main features will be crucial, the Cartesian structure of the point set X , the uniform weights, and the fact that, in the isotropic case, the objective function is separable with respect to the coordinates. While the former allow us to use local arguments, the latter permits, in effect, to reduce the estimates to the 1-dimensional case. In fact, indicating, as usual, the t th coordinate of a vector y by y_t , we have

$$\begin{aligned} \text{cost}_{\mathcal{A}, K}(X, \Omega, C, S) &= \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_2^2 = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \sum_{t=1}^d ((x_j)_t - (s_i)_t)^2 \\ &= \sum_{t=1}^d \left(\sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j ((x_j)_t - (s_i)_t)^2 \right). \end{aligned}$$

Next, we will provide some preparatory results. Then, in Section 3.4, we settle the 1-dimensional case, and subsequently extend the result to arbitrary dimensions in Section 3.5. Naturally, certain ingredients of our proof resemble the rationale behind the analysis of general coresets in [FG22a]. Due to the need to utilize the specifics of the resolution setting to the extreme most parts of the following proof go, however, substantially beyond general estimates.

Merging function

Let $\boldsymbol{\rho} = (\rho_1, \dots, \rho_d) \in \mathbb{N}^d$, and $\boldsymbol{\tau} = (\tau_1, \dots, \tau_d) \in \mathbb{N}^d$ with $\boldsymbol{\tau} \leq \boldsymbol{\rho}$. While $\boldsymbol{\rho}$ is the fixed benchmark resolution, the parameter vector $\boldsymbol{\tau}$ will be adjusted later to specify the resolution of the desired coreset. As before,

$$X = X(\boldsymbol{\rho}), \quad n = |X|, \quad \Omega = \Omega(\boldsymbol{\rho}), \quad \tilde{X} = X(\boldsymbol{\tau}), \quad \tilde{n} = |\tilde{X}|, \quad \tilde{\Omega} = \Omega(\boldsymbol{\tau}),$$

with weights

$$\omega_j = \nu = 2^{-\rho_1} \cdot \dots \cdot 2^{-\rho_d} \quad (j \in [n]), \quad \tilde{\omega}_q = 2^{-\tau_1} \cdot \dots \cdot 2^{-\tau_d} \quad (q \in [\tilde{n}]).$$

We will interpret the coarsening of the resolution from $\boldsymbol{\rho}$ to $\boldsymbol{\tau}$ in terms of a *uniform neighborhood merger*. It is facilitated by the *merging function*

$$p : X \rightarrow \tilde{X}$$

defined by

$$\begin{pmatrix} \frac{1}{2^{\rho_1+1}} + \frac{j_1-1}{2^{\rho_1}} \\ \vdots \\ \frac{1}{2^{\rho_d+1}} + \frac{j_d-1}{2^{\rho_d}} \end{pmatrix} \mapsto \begin{pmatrix} \frac{1}{2^{\tau_1+1}} + \frac{q_1-1}{2^{\tau_1}} \\ \vdots \\ \frac{1}{2^{\tau_d+1}} + \frac{q_d-1}{2^{\tau_d}} \end{pmatrix}$$

for

$$j_t = r_t + (q_t - 1)2^{\rho_t - \tau_t}, \quad r_t \in [2^{\rho_t - \tau_t}], \quad q_t \in [2^{\tau_t}], \quad t \in [d].$$

Whenever this seems more convenient, we will regard p as a function of the index set $[n]$, i.e.,

$$p : [n] \rightarrow [\tilde{n}].$$

Note that, indeed,

$$\tilde{\omega}_q = \sum_{j \in p^{-1}(q)} \omega_j \quad (q \in [\tilde{n}]).$$

In effect, we replace all points of X in each *batch*

$$B_q = \{x_j : j \in p^{-1}(q)\}$$

by $\tilde{x}_q$ which is carrying the total weight of all corresponding batch points. Observe that

$$|B_q| = \frac{\nu(\boldsymbol{\tau})}{\nu(\boldsymbol{\rho})} = 2^{\rho_1 - \tau_1} \cdot \dots \cdot 2^{\rho_d - \tau_d} = \frac{\tilde{\omega}_q}{\omega_j}.$$

Given a clustering $C \in \mathcal{C}_K(k, X, \Omega)$, the merging function p gives rise to a clustering $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ defined by

$$\tilde{\xi}_{iq} = \frac{1}{\tilde{\omega}_q} \sum_{j \in p^{-1}(q)} \xi_{ij} \omega_j = \frac{\nu(\boldsymbol{\rho})}{\nu(\boldsymbol{\tau})} \sum_{j \in B_q} \xi_{ij} \quad (q \in [\tilde{n}], i \in [k]).$$

Conversely, given $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$, we obtain a clustering $C \in \mathcal{C}_K(k, X, \Omega)$ by setting

$$\xi_{ij} = \tilde{\xi}_{ip(j)} \quad (j \in [n], i \in [k]).$$

This defines an extension

$$f : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, X, \Omega)$$

that converts clusterings of $(\tilde{X}, \tilde{\Omega})$ into clusterings of (X, Ω) . Note that the extension f assigns each point x_j of X to clusters in the same way its representative $x_{p(j)}$ in $\tilde{X}$ is assigned to clusters. As we will prove later, the induced approximation error can be tightly controlled.

Although this is not an issue in our present context, let us point out that coarsening the resolution generally excludes the integrality of clusterings. Of course, the constraint matrix is still totally unimodular but, unless the prescribed cluster weights κ_i are divisible by $\nu(\tau)$, the set $\mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ will not contain any integer clustering. While, by [BG12, Cor. 2.3], the number of fractional variables $\tilde{\xi}_{i,q}$ that need to be accepted in optimal coreset clusterings is at most $2(k-1)$ the extension g “spreads” any such non-integrality to all points in the corresponding batches B_q .

Power diagrams

As pointed out before, anisotropic power diagrams are desired representations for polycrystals in our motivating application. Diagrams will, however, also play a crucial role in the proof of our main theorem. As, in the isotropic setting, $\mathcal{A} = \mathcal{E}$ the *power diagram*

$$\mathcal{P} = \mathcal{P}(S, \Gamma) = (P_1, \dots, P_k)$$

is specified only by sites $S = \{s_1, \dots, s_k\}$ and sizes $\Gamma = \{\gamma_1, \dots, \gamma_k\}$, and provides a dissection of $\mathbb{R}^d$ into *polyhedral* cells

$$P_i = \{x \in \mathbb{R}^d : \|x - s_i\|_2^2 + \gamma_i \leq \|x - s_\ell\|_2^2 + \gamma_\ell \quad \forall \ell \in [k] \setminus \{i\}\}.$$

We say that a clustering $C \in \mathcal{C}_K(k, X, \Omega)$ *admits* a power diagram if there exists a power diagram $\mathcal{P}$ such that

$$\text{supp}(C_i) = \{x_j \in X : \xi_{ij} > 0\} \subset P_i \quad (i \in [k]).$$

In this situation $\mathcal{P}$ and C are called *compatible*. If, actually

$$\text{supp}(C_i) = P_i \cap X \quad (i \in [k])$$

then $\mathcal{P}$ and C are *strongly compatible*. Note that if, in addition, C is integer then $\text{supp}(C_i)$ lies in the interior of P_i for each i , i.e., none of the point in X lies on the boundary of any of the cells of $\mathcal{P}$.

The proofs of the subsequent Theorem 3.10 and Remark 3.13 will make use of the following result.

Proposition 3.6 ([BG12, Cor. 2.2])

Any $C \in \mathcal{C}_K(k, X, \Omega)$ which attains $\text{cost}(X, \Omega, S)$ admits a power diagram. Further, there exists an optimal clustering $C^* \in \mathcal{C}_K(k, X, \Omega)$ and a power diagram $\mathcal{P}$ which are strongly compatible.

See [BGK17] and the handbook article [GK17] for generalizations and additional pointers to the literature.

Batch error

For a nonempty subset Y of X , let

$$c(Y) = \frac{1}{|Y|} \sum_{x \in Y} x$$

denote its *centroid*, and set

$$\mathcal{V}(Y) = \sum_{x \in Y} \nu(\rho) \|x - c(Y)\|_2^2.$$

We will mostly (but not exclusively) apply this notion to batches B_q with respect to the resolution τ , then use the notation $\mathcal{V}_\tau(B_q)$ and speak of the *batch error*.

Let us point out that the centroid $c(B_q)$ of a batch B_q lies in $\tilde{X}$, and is actually the point resulting from the merging process.

Remark 3.7

Let $q \in [\tilde{n}]$, then $c(B_q) = \tilde{x}_q$.

Proof. For $t \in [d]$, let $q_t \in [2^{\tau_t}]$ such that $\tilde{x}_q = \tilde{x}_{q_1, \dots, q_d}$. Then, using the abbreviation $\alpha_t = 2^{\rho_t - \tau_t}$, we have for the t th coordinate $c(B_q)_t$ of the centroid $c(B_q)$

$$\begin{aligned} c(B_q)_t &= \frac{1}{|B_q|} \sum_{x \in B_q} c(B_q)_t = \frac{\nu(\rho)}{\nu(\tau)} \sum_{x \in B_q} c(B_q)_t = \frac{1}{\alpha_t} \sum_{r_t=1}^{\alpha_t} \left(\frac{-1}{2^{\rho_t+1}} + \frac{r_t + (q_t - 1)\alpha_t}{2^{\rho_t}} \right) \\ &= \frac{1}{\alpha_t} \left(\frac{-\alpha_t}{2^{\rho_t+1}} + \frac{\alpha_t(\alpha_t + 1)}{2^{\rho_t+1}} \right) + \frac{\alpha_t^2(q_t - 1)}{2^{\rho_t}} \\ &= \frac{\alpha_t(2q_t - 1)}{2^{\rho_t+1}} = \frac{2q_t - 1}{2^{\tau_t+1}} = \frac{1}{2^{\tau_t+1}} + \frac{q_t - 1}{2^{\tau_t}} = (\tilde{x}_q)_t. \end{aligned} \quad \square$$

Remark 3.7 shows that the reduction of the resolution from ρ to τ can be viewed as batching. Accordingly, the batch error $\mathcal{V}_\tau(B_q)$ quantifies the effect of the merging process. Note that, due to the uniform batching process, the batch error is independent of q and we will therefore omit B_q and simply write $\mathcal{V}(\tau) = \mathcal{V}_\tau(B_q)$.

Lemma 3.8

We have

$$\mathcal{V}(\tau) = \frac{1}{12} \frac{1}{2^{\tau_1}} \cdots \frac{1}{2^{\tau_d}} \sum_{t=1}^d \left(\frac{1}{2^{2\tau_t}} - \frac{1}{2^{2\rho_t}} \right).$$

Proof. Let $B = B_1$ denote the batch with centroid $(2^{-(\tau_1+1)}, \dots, 2^{-(\tau_d+1)})$. Then

$$\begin{aligned} \mathcal{V}(\tau) &= \mathcal{V}(B) = \sum_{x \in B} \nu(\rho) \|x - c(B)\|_2^2 = \nu(\rho) \sum_{t=1}^d \sum_{x \in B} (x_t - c(B)_t)^2 \\ &= \nu(\rho) \sum_{t=1}^d \left(\prod_{\ell \in [d] \setminus \{t\}} 2^{\rho_\ell - \tau_\ell} \right) \sum_{r=1}^{2^{\rho_t - \tau_t}} \left(\frac{-1}{2^{\rho_t+1}} + \frac{r}{2^{\rho_t}} - \frac{1}{2^{\tau_t+1}} \right)^2 \\ &= \nu(\tau) \sum_{t=1}^d \frac{1}{2^{\rho_t - \tau_t}} \sum_{r=1}^{2^{\rho_t - \tau_t}} \left(\frac{-1}{2^{\rho_t+1}} + \frac{r}{2^{\rho_t}} - \frac{1}{2^{\tau_t+1}} \right)^2. \end{aligned}$$

Now, for $\rho = \rho_t$, $\tau = \tau_t$ and with $\alpha = 2^{\rho-\tau}$,

$$\begin{aligned} \sum_{r=1}^{\alpha} \left(\frac{-1}{2^{\rho+1}} + \frac{r}{2^{\rho}} - \frac{1}{2^{\tau+1}} \right)^2 &= \frac{1}{2^{2\rho+2}} \sum_{r=1}^{\alpha} \left(2r - (\alpha + 1) \right)^2 \\ &= \frac{1}{2^{2\rho+2}} \left(4 \frac{\alpha(\alpha+1)(2\alpha+1)}{6} - 4(\alpha+1) \frac{\alpha(\alpha+1)}{2} + \alpha(\alpha+1)^2 \right) \\ &= \frac{1}{3} \frac{1}{2^{2\rho+2}} \alpha(\alpha^2 - 1) = \frac{1}{3} \frac{1}{2^{\rho+\tau+2}} (2^{2(\rho-\tau)} - 1). \end{aligned}$$

Hence,

$$\begin{aligned} \mathcal{V}(\tau) &= \frac{1}{3} \nu(\tau) \sum_{t=1}^d \frac{1}{2^{\rho_t-\tau_t}} \frac{1}{2^{\rho_t+\tau_t+2}} (2^{2(\rho_t-\tau_t)} - 1) = \frac{1}{3} \nu(\tau) \sum_{t=1}^d \frac{1}{2^{2(\rho_t+1)}} (2^{2(\rho_t-\tau_t)} - 1) \\ &= \frac{1}{12} \nu(\tau) \sum_{t=1}^d \left(\frac{1}{2^{2\tau_t}} - \frac{1}{2^{2\rho_t}} \right), \end{aligned}$$

as claimed. $\square$

In the next section we will also consider the error with respect to an arbitrary site. This can be handled with the aid of the following remark (which is folklore, can be found in [BBG17], and actually holds in greater generality than needed here, see [FG22a, Lemma 3.2 (a)]).

Remark 3.9

Let Y be a nonempty subset of X , $s \in \mathbb{R}$. Then,

$$\sum_{x \in Y} \nu(\rho) \|x - s\|_2^2 = \mathcal{V}(Y) + \nu(\rho) \cdot |Y| \cdot \|c(Y) - s\|_2^2,$$

hence, in particular, the error on the left is minimized for $s = c(Y)$.

Proof. Since $\|x - s\|_2^2 = \|x - c\|_2^2 + 2(x - c)^T(c - s) + \|c - s\|_2^2$ and

$$\sum_{x \in Y} (x - c)^T(c - s) = \left(\left(\sum_{x \in Y} x \right) - |Y|c \right)^T (c - s) = 0,$$

we have

$$\begin{aligned} \sum_{x \in Y} \nu(\rho) \|x - s\|_2^2 &= \left(\sum_{x \in Y} \nu(\rho) \|x - c\|_2^2 \right) + \nu(\rho) \cdot |Y| \cdot \|c - s\|_2^2 \\ &= \mathcal{V}(Y) + \nu(\rho) \cdot |Y| \cdot \|c - s\|_2^2, \end{aligned}$$

which also implies the second assertion. $\square$

3.4 Proof of Theorem 3.4: The case $d = 1$

While up to now all computations were exact for an arbitrary dimension d we will next derive estimates for the costs of clusterings in dimension 1. In Section 3.5 these results will, in effect, be applied for each line parallel to any coordinate axis through points of $\tilde{X}$. We will formulate them already with a view towards their later use.

Since, in this section, we are considering exclusively the 1-dimensional situation we will adopt the simplified notation

$$d = 1, \quad \rho = \boldsymbol{\rho}, \quad \tau = \boldsymbol{\tau},$$

and do not distinguish between the 1-dimensional vectors x_j , $\tilde{x}_q$ and their coordinates $(x_j)_1$, $(\tilde{x}_j)_1$, respectively.

Best unconstrained least squares clustering

We begin by computing the cost $\text{OPT}_{\hat{k}}(X, \Omega)$ of a *best unconstrained least squares clustering* of the 1-dimensional resolution set (X, Ω) into

$$\hat{k} = 2^{\lceil \log(k) \rceil}$$

clusters. The following result will later be used as a lower bound for $\text{cost}(X, \Omega, C, S)$ for any choice of C and S .

Theorem 3.10

Let $\gamma = \lceil \log(k) \rceil$, $\hat{k} = 2^\gamma$, and let $\hat{C} = (C_1, \dots, C_{\hat{k}})$ be an optimal unconstrained least-squares clustering of X . Then, each cluster contains the same number $2^{\rho-\gamma}$ of points of X , the centroids $c_i = c(C_i)$ of the $\hat{k}$ clusters are the equidistant points

$$c_i = \frac{1}{2\hat{k}} + \frac{i-1}{\hat{k}} = \frac{1}{2^{\gamma+1}} + \frac{i-1}{2^\gamma} \quad (i \in [\hat{k}]),$$

and

$$\text{OPT}_{\hat{k}}(X, \Omega) = \frac{1}{3} \frac{1}{2^{2(\rho+1)}} (2^{2(\rho-\gamma)} - 1) = \frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(\frac{2^{2\rho}}{\hat{k}^2} - 1 \right).$$

Proof. Since, by Proposition 3.6, $\hat{C}$ admits a power diagram, the points in each cluster are consecutive. Without loss of generality we may also assume that $\hat{C}$ is integer. So, with $\beta_{\hat{k}+1} = n$,

$$\alpha_i = |C_i|, \quad \beta_i = \sum_{\ell=1}^{i-1} \alpha_\ell, \quad I_i = [\beta_{i+1}] \setminus [\beta_i] \quad (i \in [\hat{k}])$$

(and the standard interpretation of empty sums and $[0] = \emptyset$) we have for $j \in I_i$

$$x_j = \frac{1}{2^{\rho+1}} + \frac{j-1}{2^\rho} = \frac{\beta_i}{2^\rho} + \frac{1}{2^{\rho+1}} + \frac{j-1-\beta_i}{2^\rho} = \frac{\beta_i}{2^\rho} + x_{j-\beta_i}.$$

Hence

$$\begin{aligned} c_i &= \frac{1}{\alpha_i} \sum_{j \in I_i} x_j = \frac{\beta_i}{2^\rho} + \frac{1}{\alpha_i} \sum_{j=1}^{\alpha_i} x_j = \frac{\beta_i}{2^\rho} + \frac{1}{\alpha_i} \sum_{j=1}^{\alpha_i} \left(\frac{1}{2^{\rho+1}} + \frac{j-1}{2^\rho} \right) \\ &= \frac{\beta_i}{2^\rho} + \frac{1}{\alpha_i} \left(\frac{\alpha_i}{2^{\rho+1}} + \frac{1}{2^\rho} \sum_{j=1}^{\alpha_i} (j-1) \right) = \frac{\beta_i}{2^\rho} + \frac{\alpha_i}{2^{\rho+1}}. \end{aligned}$$

Therefore we obtain as C_i 's contribution to the cost of the clustering

$$\begin{aligned} \mathcal{V}(C_i) &= \sum_{j=1}^{\alpha_i} \nu(\rho) \left(x_j - c_i \right)^2 = \nu(\rho) \sum_{j=1}^{\alpha_i} \left(\frac{1}{2^{\rho+1}} + \frac{j-1}{2^\rho} - \frac{\alpha_i}{2^{\rho+1}} \right)^2 \\ &= \frac{1}{2^{3\rho+2}} \sum_{j=1}^{\alpha_i} \left((\alpha_i + 1) - 2j \right)^2 = \frac{1}{2^{3\rho+2}} \sum_{j=1}^{\alpha_i} \left((\alpha_i + 1)^2 - 4(\alpha_i + 1)j + 4j^2 \right) \\ &= \frac{1}{2^{3\rho+2}} \left(-\alpha_i(\alpha_i + 1)^2 + \frac{2}{3} \alpha_i(\alpha_i + 1)(2\alpha_i + 1) \right) = \frac{1}{3} \frac{1}{2^{3\rho+2}} \alpha_i(\alpha_i^2 - 1). \end{aligned}$$

Since $\hat{C}$ is optimal the cluster cardinalities minimize $\sum_{i=1}^{\hat{k}} \alpha_i(\alpha_i^2 - 1)$ (as a function of the α_i) under the constraints $\sum_{i=1}^{\hat{k}} \alpha_i = n$ and $\alpha_i \in \mathbb{N} \cup \{0\}$. Now note that the i th summand of the objective function is strictly increasing in α_i . Hence, in the minimum we must have $\alpha_i \in \mathbb{N}$ for all $i \in [\hat{k}]$.

Let us now discard the integrality condition and consider the constrained nonlinear program

$$\begin{aligned} \min \quad & \sum_{i=1}^{\hat{k}} \alpha_i(\alpha_i^2 - 1) \\ \sum_{i=1}^{\hat{k}} \alpha_i &= n \\ \alpha_i &\geq 1 \quad (i \in [\hat{k}]) \end{aligned}$$

in the real variables $\alpha_1, \dots, \alpha_k$. As all occurring functions are continuously differentiable and the objective function is convex in the nonnegative orthant we can apply the Karush-Kuhn-Tucker Theorem and obtain the conditions

$$\begin{aligned} (3\alpha_1^2 - 1, \dots, 3\alpha_k^2 - 1) + \sum_{i=1}^{\hat{k}} \mu_i(-1, \dots, -1) + \lambda(1, \dots, 1) &= 0, \\ \sum_{i=1}^{\hat{k}} \alpha_i &= n, \quad \alpha_1, \dots, \alpha_{\hat{k}} \geq 1, \quad \mu_1, \dots, \mu_{\hat{k}} \geq 0, \quad \sum_{i=1}^{\hat{k}} \mu_i \alpha_i = 0 \end{aligned}$$

involving the Lagrange parameters μ_i and λ . Since $\alpha_i \geq 1$ and $\mu_i \geq 0$ for all $i \in [\hat{k}]$, the complementary slackness condition yields $\mu_1 = \dots = \mu_{\hat{k}} = 0$, and the first condition reads

$$(3\alpha_1^2 - 1 + \lambda, \dots, 3\alpha_k^2 - 1 + \lambda) = 0.$$

This implies that all α_i coincide. Thus we obtain

$$\alpha_1 = \dots = \alpha_{\hat{k}} = n/\hat{k} = 2^{\rho-\gamma} \in \mathbb{N}, \quad \beta_i = (i-1)2^{\rho-\gamma} \quad (i \in [\hat{k}])$$

and

$$c_i = \frac{\beta_i}{2^\rho} + \frac{\alpha_i}{2^{\rho+1}} = \frac{1}{2^{\gamma+1}} + \frac{i-1}{2^\gamma} \quad (i \in [\hat{k}]),$$

as claimed. Also, $\mathcal{V}(C_1) = \dots = \mathcal{V}(C_{\hat{k}}) = \mathcal{V}(\gamma)$ and hence

$$\text{OPT}_{\hat{k}}(X, \Omega) = 2^\gamma \mathcal{V}(\gamma) = 2^\gamma \cdot \frac{1}{3} \frac{1}{2^{3\rho+2}} 2^{\rho-\gamma} (2^{2(\rho-\gamma)} - 1) = \frac{1}{3} \frac{1}{2^{2(\rho+1)}} (2^{2(\rho-\gamma)} - 1),$$

which completes the proof. $\square$

Note that the centroids (and hence, by Remark 3.9 best choices of sites) are exactly the points of the data set $X(\gamma)$ and thus the cluster error is the batch error $\mathcal{V}(\gamma)$. As a corollary we obtain the desired lower bound.

Corollary 3.11

Let S be a set of k sites and $C \in \mathcal{C}_K(k, X, \Omega)$. Then

$$\text{cost}(X, \Omega, C, S) \geq \text{OPT}_k(X, \Omega) \geq \frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(\frac{2^{2(\rho-1)}}{k^2} - 1 \right).$$

Proof. Since

$$k \leq \hat{k} = 2^{\lceil \log(k) \rceil} < 2^{\log(k)+1} \leq 2k$$

and

$$\text{OPT}_{\hat{k}}(X, \Omega) \leq \text{OPT}_k(X, \Omega) \leq \text{cost}(X, \Omega, C, S),$$

Theorem 3.10 implies

$$\frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(\frac{2^{2(\rho-1)}}{k^2} - 1 \right) < \frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(\frac{2^{2\rho}}{\hat{k}^2} - 1 \right) \leq \text{cost}(X, \Omega, C, S),$$

as asserted. $\square$

Coreset property (a)

Next, we prove an even stronger version of coreset property (a).

Lemma 3.12

Let $\Delta = 2^\tau \mathcal{V}(\tau)$, $S \subset \mathbb{R}$ a set of k sites, $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$, and $C = g(\tilde{C})$, where $f : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, X, \Omega)$ is the standard extension defined in the previous section. Then

$$\text{cost}(X, \Omega, C, S) = \text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta$$

Proof. For $i \in [k]$, $j \in [n]$ and $q \in [\tilde{n}]$ let, as before, ξ_{ij} and $\tilde{\xi}_{iq}$ denote the components of C and $\tilde{C}$, respectively. Further recall, that the merging function maps each point of a batch B_q to its centroid. Then, using Remark 3.9, we have

$$\begin{aligned}
 \text{cost}(X, \Omega, C, S) &= \sum_{i=1}^k \sum_{j=1}^n \nu(\rho) \cdot \xi_{ij} \cdot (x_j - s_i)^2 = \sum_{i=1}^k \sum_{q=1}^{\tilde{n}} \tilde{\xi}_{iq} \sum_{x_j \in B_q} \nu(\rho) \cdot (x_j - s_i)^2 \\
 &= \sum_{i=1}^k \sum_{q=1}^{\tilde{n}} \tilde{\xi}_{iq} (\mathcal{V}(\tau) + \nu(\rho) \cdot |B_q| \cdot (x_q - s_i)^2) \\
 &= \tilde{n} \mathcal{V}(\tau) + \sum_{i=1}^k \sum_{q=1}^{\tilde{n}} \tilde{\xi}_{iq} \frac{2^{\rho-\tau}}{2^\rho} (x_q - s_i)^2 = 2^\tau \mathcal{V}(\tau) + \sum_{i=1}^k \sum_{q=1}^{\tilde{n}} \nu(\tau) \tilde{\xi}_{iq} (x_q - s_i)^2 \\
 &= \Delta + \text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S),
 \end{aligned}$$

which proves the assertion. $\square$

Coreset property (b): Setup and good indices

Not surprisingly, the proof of coreset property (b) is much more involved, and the rest of this section will be devoted to that. First, we observe, that it follows from a more general inequality.

Remark 3.13

Suppose that there exists a constant Δ such that the inequality

$$(b') \quad \text{cost}(\tilde{X}, \tilde{\Omega}, p(C), S) + \Delta \leq (1 + \epsilon) \text{cost}(X, \Omega, C, S)$$

holds for every set S of k sites and any integer clustering $C \in \mathcal{C}_K(k, X, \Omega)$ which admits a strongly compatible power diagram. Then, for every such S ,

$$\text{cost}(\tilde{X}, \tilde{\Omega}, S) + \Delta \leq (1 + \epsilon) \text{cost}(X, \Omega, S).$$

Proof. First note that, by Proposition 3.6, there exists an integer clustering C^* which admits a strongly compatible power diagram and attains $\text{cost}(X, \Omega, S)$. Then, of course,

$$\text{cost}(\tilde{X}, \tilde{\Omega}, S) \leq \text{cost}(\tilde{X}, \tilde{\Omega}, p(C^*), S) \quad \text{and} \quad \text{cost}(X, \Omega, C^*, S) = \text{cost}(X, \Omega, S).$$

Hence, by assumption,

$$\begin{aligned}
 \text{cost}(\tilde{X}, \tilde{\Omega}, S) + \Delta &\leq \text{cost}(\tilde{X}, \tilde{\Omega}, p(C^*), S) + \Delta \leq (1 + \epsilon) \text{cost}(X, \Omega, C^*, S) \\
 &= (1 + \epsilon) \text{cost}(X, \Omega, S).
 \end{aligned}$$

$\square$

According to Remark 3.13, it suffices to prove (b'). So let, in the following, S be a set of k sites, let $C = (C_1, \dots, C_k) \in \mathcal{C}_K(k, X, \Omega)$ be an integer clustering which admits the strongly compatible power diagram $\mathcal{P} = (P_1, \dots, P_k)$, and set $\tilde{C} = p(C)$.

We split the cost of the clustering $\tilde{C}$ of the resolution coreset into two components which reflect whether batches are fully contained in one of the cells of $\mathcal{P}$ or split by the diagram.

More precisely, for each $q \in [\tilde{n}]$, let $i(q)$ denote the unique index $i \in [k]$ such that $x_q \in P_i$, and set

$$\tilde{I}^{(+)} = \{q \in [\tilde{n}] : B_q \subset P_{i(q)}\}, \quad \tilde{I}^{(-)} = [\tilde{n}] \setminus \tilde{I}^{(+)}.$$

The elements of $\tilde{I}^{(+)}$ and $\tilde{I}^{(-)}$ will be called *good* and *bad indices*, respectively, as the latter require significantly more care in terms of cost estimations. Note that for good indices q the clusterings C and $\tilde{C}$ “coincide” in the sense that

$$\tilde{\xi}_{iq} = \xi_{ij} \quad \text{for all } i \in [k], j \in [n], q \in \tilde{I}^{(+)} \text{ with } x_j \in B_q.$$

Also, as the points of X in, both, the batches B_q and the intervals P_i are consecutive, we obtain the following bound.

Remark 3.14

$$|\tilde{I}^{(-)}| \leq k - 1.$$

The cost of C can be expressed in terms of the good and bad indices. In fact,

$$\text{cost}(X, \Omega, C, S) = \sum_{i=1}^k \sum_{j=1}^n \nu(\rho) \cdot \xi_{ij} \cdot (x_j - s_i)^2 = \gamma(\tilde{I}^{(+)}) + \gamma(\tilde{I}^{(-)}),$$

where

$$\gamma(\tilde{I}^{(\pm)}) = \sum_{i=1}^k \sum_{q \in \tilde{I}^{(\pm)}} \sum_{x_j \in B_q} \nu(\rho) \cdot \xi_{ij} \cdot (x_j - s_i)^2.$$

Similarly, we use the abbreviations

$$\tilde{\gamma}(\tilde{I}^{(\pm)}) = \sum_{i=1}^k \sum_{q \in \tilde{I}^{(\pm)}} \nu(\tau) \cdot \tilde{\xi}_{iq} \cdot (\tilde{x}_q - s_i)^2$$

to denote the parts of $\text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S)$ associated with $\tilde{I}^{(\pm)}$.

The cost for the good indices relates directly to the cost incurred by the corresponding original data points. In fact, by Remark 3.9 we have

$$\begin{aligned} \gamma(\tilde{I}^{(+)}) &= \sum_{i=1}^k \sum_{q \in \tilde{I}^{(+)}} \sum_{x_j \in B_q \cap C_i} \nu(\rho) \cdot (x_j - s_i)^2 \\ &= \sum_{i=1}^k \sum_{q \in \tilde{I}^{(+)}} (\mathcal{V}(B_q \cap C_i) + \nu(\rho) \cdot |B_q \cap C_i| \cdot (\tilde{x}_q - s_i)^2) \\ &= \sum_{i=1}^k \sum_{q \in \tilde{I}^{(+)}} \tilde{\xi}_{iq} (\mathcal{V}(\tau) + \nu(\tau) \cdot (\tilde{x}_q - s_i)^2). \end{aligned}$$

Hence, we obtain for the good indices,

Remark 3.15

$$\gamma(\tilde{I}^{(+)}) = |\tilde{I}^{(+)}| \cdot \mathcal{V}(\tau) + \tilde{\gamma}(\tilde{I}^{(+)}).$$

Bad indices

Now we deal with the indices in $\tilde{I}^{(-)}$. Naturally, the clustering error for the bad indices depends more specifically on the batch sizes. In the following, we assume that $\tau = \tau^*$ is chosen appropriately. In view of the extension of our results to the d -dimensional case in Section 3.5 we show slightly more than what would otherwise be needed in this section. So, let $k^* \in \mathbb{N}$ such that $k \leq k^*$ and set

$$\tau^* = \left\lceil \log \left(\frac{2^{5/3} k^*}{\epsilon^{2/3}} \right) \right\rceil.$$

Of course, we also assume that $\rho \geq \tau^*$ as before.

Next we deal with the *total deviation*

$$\text{dev}(\tilde{I}^{(-)}) = \sum_{i=1}^k \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \xi_{ij} \cdot \nu(\rho) \cdot (\tilde{x}_q - x_j)^2$$

caused by the bad indices. Let us first point out how $\text{dev}(\tilde{I}^{(-)})$ relates to the batch error.

Remark 3.16

$$\text{dev}(\tilde{I}^{(-)}) = |\tilde{I}^{(-)}| \cdot \mathcal{V}(\tau^*).$$

Proof. By Remark 3.7 we have

$$\begin{aligned} \text{dev}(\tilde{I}^{(-)}) &= \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \sum_{i=1}^k \xi_{ij} \cdot \nu(\rho) \cdot (\tilde{x}_q - x_j)^2 = \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \nu(\rho) \cdot (\tilde{x}_q - x_j)^2 \\ &= \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \nu(\rho) \cdot (c(B_q) - x_j)^2 = \sum_{q \in \tilde{I}^{(-)}} \mathcal{V}(B_q) = |\tilde{I}^{(-)}| \cdot \mathcal{V}(\tau^*). \quad \square \end{aligned}$$

The next lemma provides an upper bound for the total deviation. It will employ Lemma 3.8 in its following 1-dimensional form.

Remark 3.17

$$\mathcal{V}(\tau) = \frac{1}{12} \frac{1}{2^\rho} \left(\frac{1}{2^{2\tau}} - \frac{1}{2^{2\rho}} \right) = \frac{1}{3} \frac{1}{2^\rho} \frac{2^{2(\rho-\tau)} - 1}{2^{\rho+\tau+2}}.$$

Lemma 3.18

$$\text{dev}(\tilde{I}^{(-)}) \leq \frac{\epsilon^2}{8} \text{OPT}_k(X, \Omega) < \frac{\epsilon}{6} \text{cost}(X, \Omega, C, S).$$

Proof. By Remarks 3.14, 3.16 and 3.17,

$$\text{dev}(\tilde{I}^{(-)}) = |\tilde{I}^{(-)}| \cdot \mathcal{V}(\tau^*) \leq (k-1) \cdot \frac{1}{3} \frac{1}{2^{2\rho+\tau^*+2}} (2^{2(\rho-\tau^*)} - 1),$$

and, by Corollary 3.11,

$$\text{OPT}_k(X, \Omega) \geq \frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(\frac{2^{2(\rho-1)}}{k^2} - 1 \right).$$

Thus

$$\text{dev}(\tilde{I}^{(-)}) - \frac{\epsilon^2}{8} \text{OPT}_k(X, \Omega) \leq \frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(\frac{k-1}{2^{\tau^*}} \left(2^{2(\rho-\tau^*)} - 1 \right) - \frac{\epsilon^2}{8} \left(\frac{2^{2(\rho-1)}}{k^2} - 1 \right) \right).$$

Setting

$$f(k) = \frac{k-1}{2^{\tau^*}} \left(2^{2(\rho-\tau^*)} - 1 \right) - \frac{\epsilon^2}{8} \left(\frac{2^{2(\rho-1)}}{k^2} - 1 \right)$$

we have for $k < k^*$

$$f(k+1) - f(k) = \frac{1}{2^{\tau^*}} \left(2^{2(\rho-\tau^*)} - 1 \right) + \frac{\epsilon^2 \cdot 2^{2(\rho-1)}}{8} \left(\frac{2k+1}{k^2(k+1)^2} \right) \geq 0.$$

Hence $f(k)$ is increasing in k , and we have

$$\begin{aligned} \text{dev}(\tilde{I}^{(-)}) - \frac{\epsilon^2}{8} \text{OPT}_k(X, \Omega) &\leq \frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(\frac{k^*-1}{2^{\tau^*}} \left(2^{2(\rho-\tau^*)} - 1 \right) - \frac{\epsilon^2}{8} \left(\frac{2^{2(\rho-1)}}{(k^*)^2} - 1 \right) \right) \\ &= \frac{1}{3} \frac{1}{2^{2(\rho+1)}} \left(2^{2\rho} \left(\frac{k^*-1}{2^{3\tau^*}} - \frac{\epsilon^2}{32(k^*)^2} \right) - \left(\frac{k^*-1}{2^{\tau^*}} - \frac{\epsilon^2}{8} \right) \right), \end{aligned}$$

and, to prove the first asserted inequality, it suffices to show that

$$\frac{k^*-1}{2^{3\tau^*}} \leq \frac{\epsilon^2}{32(k^*)^2}, \quad \frac{k^*-1}{2^{\tau^*}} \geq \frac{\epsilon^2}{8}.$$

To verify these inequalities, simply note that,

$$\frac{(k^*-1)(k^*)^2}{2^{3\tau^*}} \leq \frac{(k^*-1)\epsilon^2}{2^5 k^*} \leq \frac{\epsilon^2}{32},$$

and, since $k^* \geq 2$ and $\epsilon \in (0, 1/2]$,

$$\frac{k^*-1}{2^{\tau^*}} \geq \frac{k^*-1}{4k^*} \cdot \epsilon^{2/3} = \frac{\epsilon^2}{8} \left(\frac{k^*-1}{k^*} \frac{2}{\epsilon^{4/3}} \right) \geq \frac{\epsilon^2}{8}.$$

Finally note that $\epsilon^2/8 \leq \epsilon/16 < \epsilon/6$. □

The next lemma gives the desired bound for the bad indices.

Lemma 3.19

$$\tilde{\gamma}(\tilde{I}^{(-)}) \leq \gamma(\tilde{I}^{(-)}) + \frac{5\epsilon}{6} \text{cost}(X, \Omega, C, S).$$

Proof. Let us remark that the assertion can also be inferred from the more general result in [FG22a, Lemma 4.3]. As a service to the reader we give an explicit direct proof of what we need here. In fact,

$$\begin{aligned}
 \tilde{\gamma}(\tilde{I}^{(-)}) &= \sum_{i=1}^k \sum_{q \in \tilde{I}^{(-)}} \nu(\tau) \cdot \tilde{\xi}_{iq} \cdot (\tilde{x}_q - s_i)^2 \\
 &= \sum_{i=1}^k \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \nu(\rho) \cdot \xi_{ij} \cdot ((\tilde{x}_q - x_j) + (x_j - s_i))^2 \\
 &= \text{dev}(\tilde{I}^{(-)}) + \gamma(\tilde{I}^{(-)}) + 2 \sum_{i=1}^k \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \nu(\rho) \cdot \xi_{ij} \cdot (\tilde{x}_q - x_j)(x_j - s_i).
 \end{aligned}$$

Further, using the Cauchy-Schwartz inequality,

$$\begin{aligned}
 &\left(\sum_{i=1}^k \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \nu(\rho) \cdot \xi_{ij} \cdot (\tilde{x}_q - x_j)(x_j - s_i) \right)^2 \\
 &\leq \left(\sum_{i=1}^k \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \nu(\rho) \cdot \xi_{ij} \cdot (\tilde{x}_q - x_j)^2 \right) \cdot \left(\sum_{i=1}^k \sum_{q \in \tilde{I}^{(-)}} \sum_{x_j \in B_q} \nu(\rho) \cdot \xi_{ij} \cdot (x_j - s_i)^2 \right) \\
 &= \text{dev}(\tilde{I}^{(-)}) \cdot \gamma(\tilde{I}^{(-)}).
 \end{aligned}$$

Since $\epsilon \in (0, 1/2]$ and $1/16 + 1/\sqrt{2} \leq 5/6$, Lemma 3.18 hence yields

$$\begin{aligned}
 \tilde{\gamma}(\tilde{I}^{(-)}) &\leq \text{dev}(\tilde{I}^{(-)}) + \gamma(\tilde{I}^{(-)}) + 2 \left(\text{dev}(\tilde{I}^{(-)}) \cdot \gamma(\tilde{I}^{(-)}) \right)^{1/2} \\
 &\leq \gamma(\tilde{I}^{(-)}) + \frac{\epsilon^2}{8} \text{OPT}_k(X, \Omega) + 2 \left(\frac{\epsilon^2}{8} \text{OPT}_k(X, \Omega) \cdot \gamma(\tilde{I}^{(-)}) \right)^{1/2} \\
 &\leq \gamma(\tilde{I}^{(-)}) + \left(\frac{\epsilon^2}{8} + \frac{\epsilon}{\sqrt{2}} \right) \text{cost}(X, \Omega, C, S) \leq \gamma(\tilde{I}^{(-)}) + \frac{5\epsilon}{6} \text{cost}(X, \Omega, C, S),
 \end{aligned}$$

as claimed. $\square$

We can now combine the results of this subsection to show property (b').

Lemma 3.20

Let $\Delta = 2^{\tau^*} \mathcal{V}(\tau^*)$. Then

$$\text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta \leq (1 + \epsilon) \text{cost}(X, \Omega, C, S).$$

Proof. First, note that $2^{\tau^*} = |\tilde{I}^{(+)}| + |\tilde{I}^{(-)}|$. Using, in this order, Remarks 3.15 and 3.16, and Lemmata 3.18 and 3.19 we, hence, obtain

$$\begin{aligned}
 \text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta &= \tilde{\gamma}(\tilde{I}^{(+)}) + |\tilde{I}^{(+)}| \cdot \mathcal{V}(\tau^*) + \tilde{\gamma}(\tilde{I}^{(-)}) + |\tilde{I}^{(-)}| \cdot \mathcal{V}(\tau^*) \\
 &= \gamma(\tilde{I}^{(+)}) + \tilde{\gamma}(\tilde{I}^{(-)}) + \text{dev}(\tilde{I}^{(-)}) \\
 &\leq \gamma(\tilde{I}^{(+)}) + \gamma(\tilde{I}^{(-)}) + \frac{5\epsilon}{6} \text{cost}(X, \Omega, C, S) + \frac{\epsilon}{6} \text{cost}(X, \Omega, C, S) \\
 &= (1 + \epsilon) \text{cost}(X, \Omega, C, S),
 \end{aligned}$$

as claimed. $\square$

Note that Δ depends only on the resolutions ρ and τ but is independent of k and K .

Resolution coresets in $[0, 1]$

The main result of this sections follows now easily.

Theorem 3.21

$(X(\tau^*), \Omega(\tau^*))$ is an ϵ -coreset for $(k, X(\rho), \Omega(\rho), K)$ of size

$$|X(\tau^*)| \leq 2^{8/3} \frac{k}{\epsilon^{2/3}}.$$

Proof. By Lemmata 3.12 and 3.20, $(X(\tau^*), \Omega(\tau^*))$ is, indeed, an ϵ -coreset and, by the choice of τ^* , we have

$$|X(\tau^*)| = 2^{\tau^*} = 2^{\left\lceil \log\left(\frac{2k}{\epsilon^{2/3}}\right) \right\rceil} \leq 2 \cdot \frac{2^{5/3}k}{\epsilon^{2/3}} = \frac{2^{8/3}k}{\epsilon^{2/3}}.$$

This completes the proof of the theorem. $\square$

3.5 Proof of Theorem 3.4: The general case

We will now prove our main theorem by reducing it to the 1-dimensional case settled already in Section 3.4. So suppose that $d \geq 2$. The key observation is that, in the isotropic case, the costs of a given clustering $C = (\xi_{ij})_{i \in [k], j \in [n]} \in \mathcal{C}_K(k, X, \Omega)$ on $X = X(\rho)$ in $\mathbb{R}^d$ is a separable function with respect to the lines parallel to the coordinate axes.

In fact, for $t \in [d]$, let u_t denote the t th standard unit vector, and set

$$X^{(-t)}(\rho) = X(\rho_1) \times \cdots \times X(\rho_{t-1}) \times \{0\} \times X(\rho_{t+1}) \times \cdots \times X(\rho_d).$$

Note that

$$X(\rho) = \left(\frac{-1}{2^{\rho_t+1}} + [2^{\rho_t}] \cdot u_t \right) + X^{(-t)}(\rho),$$

i.e., $X^{(-t)}(\rho)$ is the corresponding $(d-1)$ -dimensional Cartesian point set located in the coordinate hyperplane perpendicular to u_t . Further,

$$\mathcal{L}^{(t)} = X^{(-t)}(\rho) + \mathbb{R} \cdot u_t,$$

denote the set of lines through the Cartesian point set and parallel to the t th coordinate axis, and set $\mathcal{L} = \bigcup_{t \in [d]} \mathcal{L}^{(t)}$. Then we have

$$\begin{aligned} \text{cost}(X, \Omega, C, S) &= \sum_{t=1}^d \left(\sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \left((x_j)_t - (s_i)_t \right)^2 \right) \\ &= \sum_{t=1}^d \sum_{i=1}^k \sum_{L \in \mathcal{L}^{(t)}} \sum_{x_j \in L} \xi_{ij} \omega_j \left((x_j)_t - (s_i)_t \right)^2 \\ &= \sum_{t=1}^d \sum_{L \in \mathcal{L}^{(t)}} \left(\sum_{i=1}^k \sum_{x_j \in L} \nu(\rho) \xi_{ij} \left((x_j)_t - (s_i)_t \right)^2 \right). \end{aligned}$$

Clusterings induced on the lines in $\mathcal{L}$

The double sum in parenthesis can be interpreted as the cost of a certain clustering of the points of $X(\boldsymbol{\rho}) \cap L$ on the line L . The reduction to the 1-dimensional case is, however, somewhat technical and requires some additional notation.

Before we introduce the details let us begin with some remarks concerning our notational conventions. Obviously, we need to distinguish between the 1- and d -dimensional objects and also bear in mind, that in the 1-dimensional setting points carry different weights than in the d -dimensional situation. In addition, we need to consider sets in the $(d-1)$ -dimensional coordinate hyperplanes. In particular, batches of points in $X^{(-t)}(\boldsymbol{\rho})$ are relevant. In order to keep the notation as intelligible as possible, we will use subscripts for components (as in $\boldsymbol{\rho} = (\rho_1, \dots, \rho_d)$) but also for “intrinsically 1-dimensional” objects. For instance, we write X_L for the point set $X(\boldsymbol{\rho}) \cap L$. Naturally, X_L can be identified with the Cartesian point set $X(\rho_t)$ in $\mathbb{R}$. This will be signified by Ω_L . Note, however, that each point in $X(\rho_t)$ carries the weight $\nu(\rho_t)$ (rather than $\nu(\boldsymbol{\rho})$ as it is the case for subsets of $X(\boldsymbol{\rho})$). Further, we use superscripts (in parenthesis) to indicate the dependence of sets in $\mathbb{R}^d$ on a parameter, as e.g. in $\mathcal{L}^{(t)}$. We indicate that a single component is missing by means of a minus sign, and write, for instances, $\boldsymbol{\rho}^{(-t)} = (\rho_1, \dots, \rho_{t-1}, \rho_{t+1}, \dots, \rho_d)$. The same notation is also used if a component is fixed. In $X^{(-t)}(\boldsymbol{\rho})$, for example, the minus sign indicates that the t th coordinate of each point in $X(\boldsymbol{\rho})$ is set to 0. In other situations, again indicated by the superscript $(-t)$, the t th coordinate may, however, be fixed differently. While these conventions are used to facilitate a more intuitive exposition, all objects will, of course, be defined precisely.

Let us now introduce the needed concepts in detail. We begin with the formal definition of clusterings on the lines $L \in \mathcal{L}^{(t)}$ induced by C . Let

$$N_L = \{j \in [n] : x_j \in L\}, \quad I_{C,L} = \{i \in [k] : \exists j \in N_L : \xi_{ij} \neq 0\}, \quad k_L = |I_{C,L}|,$$

and set

$$\begin{aligned} (\kappa_i)_L &= \sum_{j \in N_L} \nu(\rho_t) \xi_{ij} & \text{for } i \in I_{C,L}, & \quad K_L = \{(\kappa_i)_L : i \in I_{C,L}\}, \\ (\omega_j)_L &= \nu(\rho_t) & \text{for } j \in N_L, & \quad \Omega_L = \{(\omega_j)_L : j \in N_L\}. \end{aligned}$$

Then, of course,

$$C_L = ((C_i)_L : i \in I_{C,L}) = (\xi_{ij} : i \in I_{C,L}, j \in N_L) \in \mathcal{C}_{K_L}(k_L, X_L, \Omega_L).$$

Note that

$$\sum_{i \in I_{C,L}} (\kappa_i)_L = \sum_{i \in I_{C,L}} \sum_{j \in N_L} \xi_{ij} (\omega_j)_L = \sum_{x_j \in X_L} \nu(\rho_t) = \nu(\rho_t) \cdot |X_L| = 1.$$

Further, let $(s_i)_L$ denote the orthogonal projection of s_i on L , and set

$$S_L = \{(s_i)_L : i \in I_{C,L}\}.$$

Let us point out that $(s_i)_L = (s_\ell)_L$ whenever $(s_i)_t = (s_\ell)_t$, i.e. S_L may, in general, be a multiset. In such a situation it is straightforward to merge the corresponding cells of C_L , perform subsequent operations in the standard model and, finally, split-up the assignment again. We prefer, however, not to further complicate the notation and simply apply this additional

technicality tacitly. Note that, when C is integer and admits a strongly compatible power diagram, the sites $(s_i)_L$ for $i \in I_{C,L}$ are all different anyway.

With the introduced notation we see that

$$\begin{aligned} \sum_{i=1}^k \sum_{x_j \in L} \nu(\rho) \xi_{ij} \left((x_j)_t - (s_i)_t \right)^2 &= \sum_{i \in S_{C,L}} \sum_{j \in N_L} \nu(\rho) \xi_{ij} \|x_j - (s_i)_L\|^2 \\ &= \nu(\rho^{(-t)}) \sum_{i \in S_{C,L}} \sum_{j \in N_L} \nu(\rho_t) \xi_{ij} \|x_j - (s_i)_L\|^2 = \nu(\rho^{(-t)}) \text{cost}(X_L, \Omega_L, C_L, S_L). \end{aligned}$$

and hence have the following result.

Remark 3.22

$$\text{cost}(X, \Omega, C, S) = \sum_{t=1}^d \sum_{L \in \mathcal{L}^{(t)}} \nu(\rho^{(-t)}) \text{cost}(X_L, \Omega_L, C_L, S_L).$$

Remark 3.22 gives the desired split of the costs of C into the costs of the induced clusterings C_L for $L \in \mathcal{L}$. This opens up the possibility to apply results from Section 3.4.

Resolution coresets

As before we will use the simplified notation $X = X(\rho)$, $\tilde{X} = X(\tau)$, and generally signify objects related to $\tilde{X}$ by means of $\sim$ put on top. In particular, we set

$$\tilde{\mathcal{L}}^{(t)} = \tilde{X}^{(-t)} + \mathbb{R} \cdot u_t \quad \text{for } t \in [d], \quad \tilde{\mathcal{L}} = \bigcup_{t \in [d]} \tilde{\mathcal{L}}^{(t)},$$

and let $\mathcal{B}^{(-t)}$ denote the set of batches in $X^{(-t)}$ induced by $\tilde{X}^{(-t)}$ by. Further, for $B \in \mathcal{B}^{(-t)}$, let $\mathcal{L}^{(t)}(B) = B + \mathbb{R} \cdot u_t$. For future reference, the following remark collects the cardinalities of the different sets.

Remark 3.23

We have $\nu(\tau^{(-t)}) \cdot |\mathcal{B}^{(-t)}| = 1$ and, for $B \in \mathcal{B}^{(-t)}$,

$$|B| = \frac{\nu(\tau^{(-t)})}{\nu(\rho^{(-t)})} = \prod_{\ell \in [d] \setminus \{t\}} 2^{\rho_\ell - \tau_\ell} = |\mathcal{L}^{(t)}(B)|, \quad |\mathcal{L}^{(t)}| = |\mathcal{L}^{(t)}(B)| \cdot |\tilde{\mathcal{L}}^{(t)}|$$

In fact, the lines of $\tilde{\mathcal{L}}^{(t)}$ are in 1-to-1 correspondence with the batches in $\mathcal{B}^{(-t)}$ as each line $\tilde{L} \in \tilde{\mathcal{L}}^{(t)}$ intersects exactly one such batch. In the following, we use the notation $\tilde{L} = \tilde{L}(B)$ and $B = B(\tilde{L})$ to indicate this correspondence. Note that the point $\tilde{L} \cap X^{(-t)} \in B$ is the centroid of B . Also, recall from Section 3.3 that the merging function $p : X \rightarrow \tilde{X}$ has a Cartesian structure, too. More precisely, for $x \in X$, we have

$$p(x) = \left(p_1((x)_1), \dots, p_d((x)_d) \right).$$

Hence merging takes place in each coordinate independently.

In the following, we will use the specific values

$$\tau_1^* = \dots = \tau_d^* = \tau^* = \left\lceil \log \left(\frac{2^{5/3} k^*}{\epsilon^{2/3}} \right) \right\rceil, \quad \boldsymbol{\tau} = \boldsymbol{\tau}^* = (\tau_1^*, \dots, \tau_d^*), \quad \tilde{X} = X(\boldsymbol{\tau}^*),$$

and define

$$\Delta_t = 2^{\tau_t^*} \mathcal{V}(\tau_t^*) \quad (t \in [d]), \quad \Delta = \sum_{t=1}^d \sum_{L \in \mathcal{L}^{(t)}} \nu(\boldsymbol{\rho}^{(-t)}) \Delta_t.$$

Finally we set $k^* = k$ and note that $k_L \leq k^*$ for each $L \in \mathcal{L}$. Since

$$|\tilde{X}| = |X(\boldsymbol{\tau}^*)|^d \leq \left(\frac{2^{8/3} k^*}{\epsilon^{2/3}} \right)^d,$$

Theorem 3.4 is proved once the two coreset properties are established. This will be done in the next two subsections, more specifically in Lemmata 3.26 and 3.27. To (slightly) simplify the notation we will omit the asterisk and simply write $\boldsymbol{\tau} = (\tau_1, \dots, \tau_d)$ for the resolution of $\tilde{X}$.

Coreset property (a)

Let $\tilde{C} = (\tilde{\xi}_{iq})_{i \in [k], q \in [\tilde{n}]} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ be a clustering on $\tilde{X}$. Recall from Section 3.3 that $\tilde{C}$ can be extended to a clustering $C = g(\tilde{C}) = (\xi_{ij})_{i \in [k], j \in [n]} \in \mathcal{C}_K(k, X, \Omega)$ by means of the extension $f : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(k, X, \Omega)$ defined by

$$\xi_{ij} = \tilde{\xi}_{iq} \quad \text{for } j \in [n], i \in [k], q \in [\tilde{n}] \text{ with } p(x_j) = \tilde{x}_q.$$

In order to apply the established 1-dimensional results we will now take a closer look at the relation between extensions in $\mathbb{R}^d$ and on the lines. Of course, Remark 3.22 can be used to express $\text{cost}(\tilde{X}, \tilde{C}, S)$ in terms of the costs of the induced clusterings on the lines $\tilde{L} \in \tilde{\mathcal{L}}$. Explicitly, it reads as follows.

Remark 3.24

$$\text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) = \sum_{t=1}^d \sum_{\tilde{L} \in \tilde{\mathcal{L}}^{(t)}} \nu(\boldsymbol{\tau}^{(-t)}) \text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}).$$

Hence, using the coreset properties on the involved lines alone we can bound the term $\text{cost}(X_L, \Omega_L C_L, S_L)$ only for $L \in \tilde{\mathcal{L}}$. Indeed, the underlying point set is $X \cap \bigcup_{\tilde{L} \in \tilde{\mathcal{L}}} \tilde{L}$ but does not contain any point from $X \setminus \bigcup_{\tilde{L} \in \tilde{\mathcal{L}}} \tilde{L}$. Due to the definition of the extension, $\text{cost}(X_L, \Omega_L, C_L, S_L)$ can, however, be included for all lines in $\mathcal{L} \setminus \tilde{\mathcal{L}}$ as well. In fact, except for a translation of the point set, the clusterings C_{L_1} and C_{L_2} for lines $L_1, L_2 \in \mathcal{L}^{(t)}$ coincide whenever there is a batch $B \in \mathcal{B}^{(-t)}$ such that $L_1, L_2 \in \mathcal{L}^{(t)}(B)$. Hence, in particular, we have the following remark.

Remark 3.25

Let $\tilde{L} \in \tilde{\mathcal{L}}^{(t)}$, $B = B(\tilde{L}) \in \mathcal{B}^{(t)}$, and $L \in \mathcal{L}^{(t)}(B)$. Then

$$\text{cost}(X_L, \Omega_L, C_L, S_L) = \text{cost}(X_{\tilde{L}}, \Omega_{\tilde{L}}, C_{\tilde{L}}, S_{\tilde{L}}).$$

The following lemma addresses coreset property (a).

Lemma 3.26

Let $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ and $C = g(\tilde{C})$. Then

$$\text{cost}(X, \Omega, C, S) = \text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta.$$

Proof. By Remark 3.23 we have $\nu(\rho^{(-t)}) \cdot |\mathcal{L}^{(t)}| = \nu(\tau^{(-t)}) \cdot |\tilde{\mathcal{L}}^{(t)}|$, for each $t \in [d]$. Hence,

$$\Delta = \sum_{t=1}^d \sum_{L \in \mathcal{L}^{(t)}} \nu(\rho^{(-t)}) \Delta_t = \sum_{t=1}^d \sum_{\tilde{L} \in \tilde{\mathcal{L}}^{(t)}} \nu(\tau^{(-t)}) \Delta_t.$$

Thus with Remarks 3.23 and 3.24 we obtain

$$\begin{aligned} \text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta &= \sum_{t=1}^d \sum_{\tilde{L} \in \tilde{\mathcal{L}}^{(t)}} \nu(\tau^{(-t)}) (\text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}) + \Delta_t) \\ &= \sum_{t=1}^d \sum_{\tilde{L} \in \tilde{\mathcal{L}}^{(t)}} \frac{1}{|B(\tilde{L})|} \sum_{L \in \mathcal{L}^{(t)}(B(\tilde{L}))} \nu(\tau^{(-t)}) (\text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}) + \Delta_t) \\ &= \sum_{t=1}^d \sum_{\tilde{L} \in \tilde{\mathcal{L}}^{(t)}} \sum_{L \in \mathcal{L}^{(t)}(B(\tilde{L}))} \nu(\rho^{(-t)}) (\text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}) + \Delta_t). \end{aligned}$$

Now, with $\tilde{L} \in \tilde{\mathcal{L}}^{(t)}$, $B = B(\tilde{L}) \in \mathcal{B}^{(t)}$, and $L \in \mathcal{L}^{(t)}(B)$, we apply Lemma 3.12 to the clustering $\tilde{C}_{\tilde{L}} \in \mathcal{C}_{K_L}(k_{\tilde{L}}, \tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}})$. With the aid of Remark 3.25, we obtain

$$\text{cost}(X_L, \Omega_L, C_L, S_L) = \text{cost}(X_{\tilde{L}}, \Omega_{\tilde{L}}, C_{\tilde{L}}, S_{\tilde{L}}) = \text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}) + \Delta_t.$$

Hence, with Remark 3.22, we can now reverse the previous arguments to see that

$$\begin{aligned} &\sum_{t=1}^d \sum_{\tilde{L} \in \tilde{\mathcal{L}}^{(t)}(\tau)} \sum_{L \in \mathcal{L}^{(t)}(B(\tilde{L}))} \nu(\rho^{(-t)}) (\text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}) + \Delta_t) \\ &= \sum_{t=1}^d \sum_{\tilde{L} \in \tilde{\mathcal{L}}^{(t)}} \sum_{L \in \mathcal{L}^{(t)}(B(\tilde{L}))} \nu(\rho^{(-t)}) \text{cost}(X_L, \Omega_L, C_L, S_L) \\ &= \sum_{t=1}^d \sum_{L \in \mathcal{L}^{(t)}} \text{cost}(X_L, \Omega_L, C_L, S_L) = \text{cost}(X, \Omega, C, S). \end{aligned}$$

This completes the proof. $\square$

Coreset property (b')

Let us now turn to coreset property (b'). So, suppose that the clustering $C \in \mathcal{C}_K(k, X, \Omega)$ is integer and admits a strongly compatible power diagram. Then, for each line $L \in \mathcal{L}$ each clustering

$$C_L = (\xi_{ij} : i \in I_{C,L}, j \in N_L) \in \mathcal{C}_{K_L}(k_L, X_L, \Omega_L)$$

is also integer and admits a strongly compatible power diagram. We can hence apply Lemma 3.20 to relate its cost to that of the corresponding coresets clustering. Note that the latter lives on L while $\tilde{C}_{\tilde{L}}$ is only defined on the lines $\tilde{L} \in \tilde{\mathcal{L}}$. The following proof of coresets property (b') will therefore explicitly address the transition from coresets on each line $L \in \mathcal{L}$ to $\tilde{C}$.

Lemma 3.27

Let $C \in \mathcal{C}_K(k, X, \Omega)$ be integer and admit a strongly compatible power diagram, and set $\tilde{C} = p(C)$. Then

$$\text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S) + \Delta \leq (1 + \epsilon) \text{cost}(X, \Omega, C, S).$$

Proof. By Remark 3.22, $\text{cost}(X, \Omega, C, S)$ can be decomposed into the costs on the lines $L \in \mathcal{L}$, i.e., $\text{cost}(X_L, \Omega_L, C_L, S_L)$. For each $C_L \in \mathcal{C}_{K_L}(k_L, X_L, \Omega_L)$ we derive a coresets clustering with $k_l \leq k^*$ according to Lemma 3.20. We refrain from denoting it by $\tilde{C}_L$ (with a wide tilde) but use a different notation for better distinction. Let $t \in [d]$, $L \in \mathcal{L}^{(t)}$ and $x_L^{(-t)} \in X^{(-t)}$ such that $L = x_L^{(-t)} + \mathbb{R} \cdot u_t$. Then we set

$$\begin{aligned} Y_L &= x_L^{(-t)} + \{x_t u_t : x_t \in X(\tau_t)\}, & \tilde{\Omega}_L &= (\nu(\tau_t), \dots, \nu(\tau_t)) \\ Z_L &= (\zeta_{ir}^{(L)})_{i \in I_{C,L}, r \in [2^{\tau_t}]} \in \mathcal{C}_{K_L}(k_L, Y_L, \tilde{\Omega}_L). \end{aligned}$$

Note that

$$Y_L = x_L^{(-t)} + \{y_1, \dots, y_{2^{\tau_t}}\} \cdot u_t \quad \text{with} \quad y_r = 2^{-(\tau_t+1)} + (r-1)2^{-\tau_t} \quad (r \in [2^{\tau_t}]).$$

By Remark 3.22 and Lemma 3.20 we have

$$\begin{aligned} (1 + \epsilon) \text{cost}(X, \Omega, C, S) - \Delta &= \sum_{t=1}^d \sum_{L \in \mathcal{L}^{(t)}} \nu(\rho^{(-t)}) ((1 + \epsilon) \text{cost}(X_L, \Omega_L, C_L, S_L) - \Delta_t) \\ &\geq \sum_{t=1}^d \sum_{L \in \mathcal{L}^{(t)}} \nu(\rho^{(-t)}) \text{cost}(Y_L, \tilde{\Omega}_L, Z_L, S_L) \\ &= \sum_{t=1}^d \sum_{B \in \mathcal{B}^{(-t)}} \sum_{L \in \mathcal{L}^{(t)}(B)} \nu(\rho^{(-t)}) \text{cost}(Y_L, \tilde{\Omega}_L, Z_L, S_L). \end{aligned}$$

Now, let $q \in [\tilde{n}]$, $B = (B_q + \mathbb{R} \cdot u_t) \cap X^{(-t)} \in \mathcal{B}^{(t)}$, $\tilde{L} = \tilde{L}(B)$, and $r = r(q) \in [2^{\tau_t}]$ such that $(\tilde{x}_q)_t = (y_r)_t$. Then, as $\tilde{x}_q \in \tilde{L}$, we have

$$\|y_r - (s_i)_L\|_2^2 = \|\tilde{x}_q - (s_i)_{\tilde{L}}\|_2^2 \quad \text{for all } L \in \mathcal{L}^{(t)}(B)$$

and

$$\begin{aligned} \tilde{\xi}_{iq} &= \frac{1}{\nu(\tau)} \sum_{x_j \in B_q} \xi_{ij} \nu(\rho) = \frac{\nu(\rho)}{\nu(\tau)} \sum_{L \in \mathcal{L}^{(t)}(B)} \sum_{x_j \in L \cap B_q} \xi_{ij} \\ &= \frac{\nu(\rho)}{\nu(\tau)} \sum_{L \in \mathcal{L}^{(t)}(B)} \frac{\nu(\tau_t)}{\nu(\rho_t)} \left(\frac{1}{\nu(\tau_t)} \sum_{x_j \in L \cap B_q} \xi_{ij} \nu(\rho_t) \right) = \frac{\nu(\rho^{(-t)})}{\nu(\tau^{(-t)})} \sum_{L \in \mathcal{L}^{(t)}(B)} \zeta_{ir}. \end{aligned}$$

Hence, with the additional setting $\zeta_{ir}^{(L)} = 0$ for $i \notin I_{C,L}$, we obtain

$$\begin{aligned}
 & \sum_{L \in \mathcal{L}^{(t)}(B)} \nu(\rho^{(-t)}) \text{cost}(Y_L, \tilde{\Omega}_L, Z_L, S_L) \\
 &= \sum_{L \in \mathcal{L}^{(t)}(B)} \nu(\rho^{(-t)}) \sum_{i \in I_{C,L}} \sum_{r=1}^{2^{\tau_t}} \zeta_{ir}^{(L)} \nu(\tau_t) \|y_r - (s_i)_L\|_2^2 \\
 &= \sum_{i=1}^k \sum_{\tilde{x}_q \in \tilde{X}_{\tilde{L}}} \left(\nu(\rho^{(-t)}) \sum_{L \in \mathcal{L}^{(t)}(B)} \zeta_{ir(q)}^{(L)} \right) \nu(\tau_t) \|\tilde{x}_q - (s_i)_{\tilde{L}}\|_2^2 \\
 &= \nu(\tau^{(-t)}) \sum_{i=1}^k \sum_{\tilde{x}_q \in \tilde{X}_{\tilde{L}}} \tilde{\xi}_{iq} \nu(\tau_t) \|\tilde{x}_q - (s_i)_{\tilde{L}}\|_2^2 = \nu(\tau^{(-t)}) \text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}).
 \end{aligned}$$

Thus,

$$\begin{aligned}
 (1 + \epsilon) \text{cost}(X, \Omega, C, S) - \Delta &\geq \sum_{t=1}^d \sum_{B \in \mathcal{B}^{(-t)}} \sum_{L \in \mathcal{L}(B)} \nu(\rho^{(-t)}) \text{cost}(Y_L, \tilde{\Omega}_L, Z_L, S_L) \\
 &= \sum_{t=1}^d \sum_{\tilde{L} \in \mathcal{L}^{(-t)}} \nu(\tau^{(-t)}) \text{cost}(\tilde{X}_{\tilde{L}}, \tilde{\Omega}_{\tilde{L}}, \tilde{C}_{\tilde{L}}, S_{\tilde{L}}) = \text{cost}(\tilde{X}, \tilde{\Omega}, \tilde{C}, S),
 \end{aligned}$$

which concludes the proof. $\square$

3.6 Proof of Theorem 3.5

While Theorem 3.4 explicitly exploits the orthogonality of the Euclidean norm and hence produces resolution coresets only for the Euclidean case we will now show that it can still be used for general anisotropic objective functions defined by means of families $\mathcal{A} = \{A_1, \dots, A_k\}$ of positive definite symmetry $(d \times d)$ -matrices. The next well-known result specifies how good constrained clusterings on coresets yield good clusterings on the original data sets. In the proof of Theorem 3.5 we will need it only in the Euclidean case and for $\delta = 1$. We formulate it here for (ϵ, δ) -coresets in the anisotropic case since it provides the rationale behind the general coreset definition given in Section 3.2 which involves two Δ -terms.

Proposition 3.28 (*[FG22a, Thm. 3.5 (a)]*)

Let $I = (k, X, \Omega, \mathcal{A}, K, S)$ be an instance of WCA ASSIGNMENT, $\epsilon \in (0, 1/2]$, $\delta \in [1, \infty)$, and $\gamma \geq \delta$. Further, let $(\tilde{X}, \tilde{\Omega})$ be an $(\epsilon/3, \delta)$ -coreset for I_S , and f its extension.

Now suppose that $\tilde{C} \in \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega})$ is a γ -approximation for $\tilde{I} = (k, \tilde{X}, \tilde{\Omega}, \mathcal{A}, K, S)$, then $f(\tilde{C})$ is a $(1 + \epsilon)\gamma$ -approximation for I .

We can now prove Theorem 3.5.

PROOF OF THEOREM 3.5: First note that for a positive definite symmetric $(d \times d)$ -matrix A with smallest and largest eigenvalues $\lambda^-(A)$ and $\lambda^+(A)$, respectively, we have

$$\lambda^-(A) \cdot \|x\|_2^2 \leq \|x\|_A^2 \leq \lambda^+(A) \cdot \|x\|_2^2 \quad (x \in \mathbb{R}^d).$$

Recalling that

$$\lambda^-(\mathcal{A}) = \min\{\lambda^-(A_i) : i \in [k]\}, \quad \lambda^+(\mathcal{A}) = \max\{\lambda^+(A_i) : i \in [k]\}.$$

We hence have

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X, \Omega, C, S) &= \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{A_i}^2 \leq \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \lambda^+(\mathcal{A}) \|x_j - s_i\|_2^2 \\ &= \lambda^+(\mathcal{A}) \cdot \text{cost}(X, \Omega, C, S), \end{aligned}$$

and, similarly,

$$\lambda^-(\mathcal{A}) \cdot \text{cost}(X, \Omega, C, S) \leq \text{cost}_{\mathcal{A},K}(X, \Omega, C, S).$$

Since $C = f(\tilde{C})$, Proposition 3.28 therefore yields

$$\begin{aligned} \text{cost}_{\mathcal{A},K}(X, \Omega, C, S) &\leq \lambda^+(\mathcal{A}) \cdot \text{cost}(X, \Omega, C, S) \leq \lambda^+(\mathcal{A})(1 + \epsilon)\gamma \cdot \text{cost}(X, \Omega, S) \\ &\leq (1 + \epsilon)\gamma \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})} \cdot \text{cost}_{\mathcal{A},K}(X, \Omega, S), \end{aligned}$$

which completes the proof.

3.7 Final remarks

As pointed out before, the size bounds for resolution coresets are superior to those for pencil coresets in small dimensions. Comparing the results of Theorem 3.4 and Proposition 3.2 we see, however, that the number of clusters enters as k^d and k^2 , respectively. In fact, resolution coresets utilize the grid structure of X but are themselves restricted to a coarser grid. Pencil coresets, on the other hand, produce very different point densities depending on the distance from the chosen vertices of the pencils. Since for resolution coresets we insist on a *uniform* resolution on each line, we cannot employ the rationale leading to the reduced dependence on k in Proposition 3.2. As known dimension-reducing coreset techniques, [Bra+16; Coh+15; FSS13; SSS20], do generally not preserve the grid structure, it remains an open question whether the term k^d in general and even k^3 in 3D in the size bound of resolution coresets can be further reduced. Also, generalizations to Cartesian point sets with arbitrary positive weights would be interesting. Further, an analysis of nonuniform tilings of $[0, 1]^d$ into axis parallel boxes would be relevant. This corresponds to adaptive thinning of Cartesian point sets and is interesting for fast image processing; see e.g. [FA20] and the papers quoted therein.

Chapter 4

Turning Grain Scans Into Diagrams

This chapter studies mathematical models for representing, imaging, and analyzing polycrystalline materials. We introduce various techniques for converting grain maps into diagram or tessellation representations that rely on constrained clustering. In particular, we show how to significantly accelerate the computation of generalized balanced power diagrams and how to extend it to allow for optimization over all relevant parameters. A comparison of the accuracy of the proposed approaches is given based on a 3D real-world data set of $339 \times 339 \times 599$ voxels.

4.1 Introduction

A challenging problem in materials science is to describe the structure of polycrystalline materials and understand the processes that govern the evolution of the structure with time, e.g., the grain growth taking place during heating; see, e.g., [Ban08; Han+09; Pou04]. Naturally, geometric concepts play an important role, e.g., [MS07], and computations can use voxel-based images at different levels of resolution, in terms of 2D [MSF12; TR18] or 3D *grain maps* [Zha+18]. However, the data acquisition and analysis tend to be slow, prohibiting the study of real-life dynamics. Moreover, the size of some grains may be comparable to the spatial resolution of the methods, rendering detailed visualization of shape impossible. For these reasons, sparse descriptions involving only a few parameters have been introduced. Such “compressed segmentations” are also relevant for materials modelling, as the reduced complexity allows for the analysis of much larger (and more representative) samples.

This text is concerned with deterministic mathematical models for converting polycrystalline images into diagram representations efficiently. We perform a “controlled analysis” based on different types of image discretization, resolutions, and aspects of data compression. Such discrete images, formally introduced in Section 4.2, will generically be referred to as “grain scans”. (This term is more general than the standard term grain map from materials science.) The main goal of our analysis is to further substantiate the potential of diagram representations with a special view, however, on later standard applications for solving the inverse problem of grain mapping when only certain measurements such as centroids, volumes, or centralized second-order moments of grains are available.

Diagrams have been applied for representations of static polycrystalline materials before. The known models, such as *Voronoi* [BBH16; XL09], *Voronoi G* [Gui+11], *Laquerre* [Alt+14; KS08; Lyc+11; Spe+14; Tel+92], *ellipsoidal growth* [TGB15], and *generalized balanced power diagrams (GBPD)* [Alp+15] vary in terms of their accuracy and their complexity. It has been observed in several studies, including [Pet+21; Šed+16; Šed+17; Spe+16; TR18] that the anisotropic power diagrams produced by the GBPD-algorithm capture the physical principles

governing the forming of polycrystals quite well. However, the high accuracy comes at the price of high computational costs, which limits the practical use of the method. In fact, all known approaches are either heuristic in nature or require a significant computational effort. Indeed, there is some need for balancing the computational effort and the desired accuracy for most applications. This problem is aggravated by a strong increase in the size of grain models both experimentally and from a material modelling point of view.

This text introduces new and significantly accelerated variants of the GBPD-algorithm of [Alp+15], which generate *anisotropic power diagrams* which are governed by parameters associated with the location, shape characteristics and the size of the grains (see Section 4.2 for a formal definition). Such diagrams constitute a class of large flexibility containing Voronoi and Laguerre tessellations as special cases. In particular, we use data compression techniques based on *coresets* which allow for a substantial “thinning” of the data without losing too much accuracy. In fact, we can compute optimal diagrams based on low-resolution ground set models whose number of points depend only on the number of grains and the desired accuracy. We will also introduce additional techniques resulting in yet sparser sets, altogether generating a class of new algorithms for computing diagram representations, which we will call *sparse-GBPD* or *s-GBPD*. As it turns out, suitable parameter settings yield algorithms in s-GBPD, which are much faster than the GBPD-algorithm at essentially no loss of accuracy, rendering the new algorithm a viable, practical tool of high accuracy. While s-GBPD is an “indirect” clustering method based on the knowledge of centroids, volumes, and centralized second-order moments of grains which are experimentally measurable parameters, we will also introduce a new technique, addressing the best fit problem directly, which allows us to optimize over all relevant parameters that govern the weight-balanced anisotropic diagrams.

This text is organized as follows. Section 4.2 formalizes the concept of grain scans, introduces diagrams, states the main problems and indicates our main contributions. Section 4.3 introduces the concept of coresets for speeding up exact methods and gives several examples of how such data compression methods can be realized in our context. In particular, we show how the original GBPD-algorithm can be turned into a computationally feasible method for practical 3D problem sizes and reasonable resolutions. Section 4.4 introduces a new, direct, and full-parameter best fit model. The results of our empirical study comparing the different methods on available practical grain scan data are stated in Section 4.5. Finally, Section 4.6 concludes with remarks that particularly address the issues of how the coreset methods can be extended to situations where instead of full grain scans only certain grain measurements are available.

4.2 Turning grain scans into diagrams

Adequate representations of grain scans can be valuable tools for understanding the physical principles of grain growth and the properties of the resulting material. The diagram representations studied in this text require only a few parameters per grain and are thus much easier to process than voxel-represented grain scans. Further, since diagram representations provide a dissection of the underlying (continuous) space into cells with disjoint interiors they can even be used to boost the scan resolution. Moreover, they exhibit geometric and combinatorial features of the grain maps (e.g., its “neighborhood structure”) that are not readily available otherwise. Hence, diagram representations may contribute to predicting and even controlling the forming of new materials.

As previous studies [Alp+15; Pet+21; Šed+16; Šed+17; Spe+16; TR18] show, only a few characteristic parameters govern decisive properties of grain scans. These parameters may, in practice, be derived from measurements of existing polycrystals or demanded by new materials. Here, we take a more fundamental perspective and assume that grain scans are available at some resolution. A specific focus will lie on the accuracy of fit of the derived diagrams. Also, computations should be fast but not necessarily real-time as we may want to spend additional time improving the accuracy of the representation. Accordingly, in Section 4.4 all parameters that govern the diagrams will be subject to optimization in order to illustrate the general fitness of diagram representations. Section 4.3, on the other hand, will (in accordance with available tomographic techniques) assume that certain parameters are directly available through measurements, and focus on efficient computations of the remaining ones; see Table 4.1 for an overview.

In this section we will first formalize the concept of a grain scan and introduce the diagrams used throughout the chapter. Then we put our studies in the subsequent sections into some broader perspective and state our main contributions.

Grain scans

Let $d \in \mathbb{N}$ be the dimension we are working in (e.g., $d = 2$ and $d = 3$ for the study of planar or 3D samples, respectively), and let $k \in \mathbb{N}$ denote the number of different grains in the given polycrystalline sample. Given a set of points $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^d$ in the sample and labels, i.e., $y_1, \dots, y_n \in [k] := \{1, \dots, k\}$, we call the labeled data set $\{(x_j, y_j) : j \in [n]\}$ a *grain scan* or *polycrystal scan*. We will refer to it as (X, Y) with the understanding, that the indices of the labels in the family $Y = \{y_1, \dots, y_n\}$ correspond to the indices of the points in X .

Typically our initial samples arise from digital images, normalized to lie in the range $[0, 1]^d$, and hence exhibit a grid structure. Such images are commonly referred to as *grain maps*. In order to speed up computations, we will, however, compress the data in various ways not all of which will sustain a grid structure.

In this notation, we set

$$G_i = \{x_j \in X : (x_j, i) \in (X, Y)\}, \quad \kappa_i = |G_i|,$$

where $|G_i|$ denotes G_i 's *cardinality*. Note that G_i can be viewed as a discretization of the i th grain in the underlying polycrystal, and κ_i is proportional to its volume.

Diagrams

Given a grain scan (X, Y) we are interested in a diagram that represents the scan as accurately as possible. In its general form, *diagrams* can be defined by functions $h_1, \dots, h_k : \mathbb{R}^d \rightarrow \mathbb{R}$. For each $i \in [k]$, the corresponding *cell* P_i of the diagram is given as

$$P_i = \{x \in \mathbb{R}^d : h_i(x) \leq h_l(x) \ \forall l \in [k]\}.$$

The choice of the functions h_i determines the diagram type; see [BGK17] and [GK17] for examples and properties of different classes of diagrams. The classical *Voronoi diagram* of k distinct points $s_1, \dots, s_k$ in $\mathbb{R}^d$, for instance, is obtained by employing the functions $h_i(x) = \|x - s_i\|_2$ where $\|\cdot\|_2$ denotes the Euclidean norm. In the following we use three types of parameters to control the properties of the cells:

- $\mathcal{A}$: k positive definite symmetric matrices $A_1, \dots, A_k \in \mathbb{R}^{d \times d}$, collected in the family $\mathcal{A} = \{A_1, \dots, A_k\}$. In the special case that each A_i is the unit matrix E_d of $\mathbb{R}^{d \times d}$ the family $\mathcal{A}$ will be denoted by $\mathcal{E}$;
- S : k different *sites* $s_1, \dots, s_k \in \mathbb{R}^d$, collected in the set $S = \{s_1, \dots, s_k\}$;
- Γ : k *sizes* $\gamma_1, \dots, \gamma_k \in \mathbb{R}$, collected in the family $\Gamma = \{\gamma_1, \dots, \gamma_k\}$.

The sites specify the general positions of the grains while the matrices in $\mathcal{A}$ describe characteristics of their shapes. In fact, each grain is equipped with its own *ellipsoidal norm* $\| \cdot \|_{A_i}$ given through A_i by

$$\|x\|_{A_i} := \sqrt{x^\top A_i x}, \quad (x \in \mathbb{R}^d).$$

Finally, the sizes γ_i can be used to control the grain volume.

The *anisotropic power diagram* (**APD**)

$$\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma) = \{P_1, \dots, P_k\}$$

is then the diagram specified by the functions

$$h_i(x) = \|x - s_i\|_{A_i}^2 + \gamma_i \quad (i \in [k]).$$

Let us point out that $\mathcal{P}$ constitutes a decomposition of $\mathbb{R}^d$ and, in particular, two different cells do not share interior points, i.e., $\text{int}(P_i) \cap \text{int}(P_\ell) = \emptyset$ for $i \neq \ell$; see [BGK17] for a proof (even for more general diagrams).

Clearly, APDs generalize the *Voronoi diagrams* (where $A_i = E_d$ and $\gamma_i = 0$ for all $i \in [k]$) and also the *power diagrams* (PD) (where $\mathcal{A} = \mathcal{E}$) whose cells are polyhedra. While the Euclidean norm $\| \cdot \|_{E_d} = \| \cdot \|_2$ corresponds to the *isotropic* case, the more general ellipsoidal norms allow us to model also *anisotropic* growth.

Representation

Given a polycrystalline scan, we are interested in an APD which *represents* the scan, i.e., ideally $G_i = X \cap P_i$ for all $i \in [k]$. Of course, in general, we cannot expect a perfect fit. Hence, we resort to some notion of *best fit*. Let us point out that, throughout this text, we only use the term “represent” in a colloquial sense. Whether even a best fit APD is in practice regarded as “representing a grain map” can only be assessed with a view towards the context of a specific application.

In order to define the accuracy of an APD $\mathcal{P} = \{P_1, \dots, P_k\}$, we consider the associated *classification function*

$$f_{\mathcal{P}} : \mathbb{R}^d \rightarrow \{0, \dots, k\}, \quad x \mapsto \begin{cases} i & \text{if } x \in \text{int}(P_i); \\ 0 & \text{otherwise.} \end{cases}$$

The function $f_{\mathcal{P}}$ maps a point $x \in X$ either to the index of its unique containing cell or to 0 if x lies on the boundary of more than one cell. Using such functions, we can formulate the problem of best-representing polycrystal scans as finding a diagram $\mathcal{P}$ such that $f_{\mathcal{P}}$ minimizes the number of misclassified data points. More precisely, we want to solve the following problem.

APD CLASSIFICATION:

Given $n \in \mathbb{N}$ and an n -element labeled data set (X, Y) , compute, for $i \in [k]$, positive definite symmetric matrices A_i , sites s_i and sizes γ_i such that the *classification error*

$$|\{j \in [n] : f_{\mathcal{P}}(x_j) \neq y_j\}|$$

of the corresponding diagram $\mathcal{P}$ is minimized over all feasible choices of $(\mathcal{A}, S, \Gamma)$. Moreover, return this classification error.

Note that points on the boundary of cells are all mapped to 0 and, hence, regarded as misclassified.

For APD CLASSIFICATION, all three characteristics $\mathcal{A}$, S , and Γ are optimized and constitute the output of the problem, together with the classification error. As this reduces the computational effort, it is often useful to specify some of these characteristics beforehand and optimize the others. Fixing characteristics will be denoted by adding them in brackets. For instance, $\text{APD CLASSIFICATION}(\mathcal{A}, S)$ refers to the situation considered in [Alp+15] that both, $\mathcal{A}$ and S , are given and the optimization involves only Γ . In effect, $\text{APD CLASSIFICATION}(\mathcal{A}, S)$ is hence the problem to compute Γ^* for which, among all $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$

$$\min_{\Gamma} |\{j \in [n] : f_{\mathcal{P}}(x_j) \neq y_j\}|$$

is attained. Even more extreme, $\text{APD CLASSIFICATION}(\mathcal{A}, S, \Gamma)$ indicates that all characteristics are given, therefore, the anisotropic power diagram $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$ is completely determined, and only the classification error $|\{j \in [n] : f_{\mathcal{P}}(x_j) \neq y_j\}|$ evoked by $\mathcal{P}$ is to be computed.

Optimization

Since d and k are fixed, it follows from [BBG17] that APD CLASSIFICATION can be solved by enumerating all relevant APDs in polynomial time. Their number grows, however, exponentially in k and d . Hence, such an enumerative approach is computationally infeasible for virtually all practically relevant instances. The known APX-hardness results of [BDEL03] for the case that k is part of the input indicate that there is not much hope to find much better algorithms for solving APD CLASSIFICATION exactly. Accordingly, various heuristics suggest to fix *all* governing characteristics; see [Alp+23a; Alt+14; BBH16; Gui+11; KS08; Lyc+11; Spe+14; TGB15; TR18; Tel+92; XL09]. The sites S are generally chosen as the measured centroids

$$c(G_i) = \frac{1}{\kappa_i} \sum_{x_j \in G_i} x_j$$

of the grains. To model the shape of the grains, [Lyc+11] used Euclidean norms, i.e., $\mathcal{A} = \mathcal{E}$, while, following [Alp+15], the heuristics of [TR18] use the inverse of the covariance matrices, i.e.,

$$A_i = (\Sigma(G_i))^{-1} = \left(\frac{1}{\kappa_i} (x_j - c(G_i) : x_j \in G_i)(x_j - c(G_i) : x_j \in G_i)^T \right)^{-1}$$

for $i \in [k]$. (It can be assumed that the matrices $\Sigma(G_i)$ have full rank d since grains that are degenerated to volume 0 do not matter.) Further, the sizes Γ are typically estimated using the

radius of the (Euclidean or ellipsoidal) ball whose volume coincides with the measured grain volume; see [Alp+23a] for an analysis in terms of discriminant analysis.

As these heuristics abstain from any optimization whatsoever, they are computationally fast but generally result in a higher misclassification error. Hence, [Alp+15] suggested a different method, called GBPD-algorithm, that allows us to optimize over Γ . The diagrams are specified through the dual variables of a linear program that addresses a constrained clustering problem; see Section 4.3 for details. As even linear programming is often too time-consuming for real-world instances of the desired size, we will, in particular, accelerate this method using data compression without any significant decrease of accuracy by employing *coresets*, which will be introduced in Section 4.3.2.

It is particularly important that this method does not require the knowledge of the grain scan. It suffices to provide estimates for $\mathcal{A}$, S , and the grain volumes, which, in practice, are often much easier to obtain from measurements. In Section 4.4 we will, nevertheless, also provide a quite efficient direct model for optimizing over all three types of parameters, $\mathcal{A}$, S and Γ . This will allow for a better assessment of the quality of the solutions provided by the faster and practically much easier method based on constrained clustering.

Contributions

In the following three sections, we will

- develop sparse variants of the GBPD-algorithm of [Alp+15], called s-GBPD, which significantly improve running times,
- introduce a new model for APD CLASSIFICATION, referred to as DiM, which facilitates the optimization over all characteristic parameters, and
- state the results of a computational study on real-world data to indicate how the different algorithms perform in terms of computation times and accuracy.

The sparse variants s-GBPD will be based on the notion of “coresets” and other data compression techniques while DiM uses a lifting method to obtain a “direct linear programming model” which can be viewed as a support vector machine approach for machine learning classification. Table 4.1 provides an overview.

Contributions	Governing Characteristics			Section
	$\mathcal{A}$	S	Γ	
s-GBPD	covariances	centroids	optimized	4.3
DiM	optimized	optimized	optimized	4.4
Evaluation				4.5

Table 4.1: Contributions: New algorithms, corresponding choice of characteristics and computational study. The entry “covariances” stands for the choices $A_i = (\Sigma(G_i))^{-1}$, where $\Sigma(G_i)$ is the covariance matrix of the i th grain.

4.3 Diagrams on sparse image supports

In this section, we assume that the matrices $\mathcal{A}$ and the sites S are given beforehand (e.g., chosen as inverse covariance matrices and centroids of the grains from the grain scan, respectively), and only the sizes Γ are subject to optimization. As, however, even $\text{APD CLASSIFICATION}(\mathcal{A}, S)$ is computationally difficult, [Alp+15] introduced the *linear programming* algorithm GBPD. The computations can be viewed as a dual method that relies on a relation between diagrams and weight-constrained anisotropic clustering. As shown by experimental computations in [Alp+15], and later supported independently by [Šed+16; Šed+17; Spe+16], GBPD is highly accurate and the underlying model seems to capture the structure of polycrystals quite well. The computational cost, however, severely limits its practical use for 3D data sets of high resolution. Here we will show how to significantly accelerate the method, leading to a new class of algorithms, which we call sparse-GBPD or s-GBPD. In particular, we will show that the clustering can be performed at a rather low resolution, essentially without losing accuracy, and this will lead to significant speed-ups.

4.3.1 Diagrams from weight-constrained clustering

Let us begin by briefly describing the relation between diagrams and clustering used in [Alp+15] which, as we will see, is based on linear programming duality. For more details, we refer the reader to [BGK17], the handbook article [GK17], and the papers quoted therein.

As before, let (X, Y) be a given grain scan, and let $\mathcal{A}$ and S be the given matrices and sites. Further, we collect the measured cardinalities $\kappa_1, \dots, \kappa_k$, which, together with the resolution, approximate the grain volumes in the family $K = \{\kappa_1, \dots, \kappa_k\}$.

While the points of the original grain scan are unweighted, the data compression techniques of Section 4.3.2 will use point weights, and hence the optimization needs to be performed on weighted point sets. Therefore, we will assume that each data point x_j carries a weight $\omega_j \in (0, \infty)$. Of course, the original unweighted case is included by simply assigning the weight 1 to each point of the grain scan. Collecting all weights in the family $\Omega = \{\omega_1, \dots, \omega_n\}$, the weighted data set will then be denoted by (X, Ω) . By convention, all parameters corresponding to the j th point will, again, be indexed by j .

A *clustering* C of (X, Ω) is a vector

$$C = (C_1, \dots, C_k) = (\xi_{11}, \dots, \xi_{1n}, \dots, \xi_{k1}, \dots, \xi_{kn})^\top$$

whose components ξ_{ij} specify the fraction of x_j that is assigned to the i th cluster $C_i = (\xi_{i1}, \dots, \xi_{in})^\top$. If all ξ_{ij} are in $\{0, 1\}$, i.e., each point is assigned to precisely one cluster, the clustering will be called *integer*.

In terms of formal constraints, any clustering satisfies the *assignment conditions*

$$\xi_{ij} \geq 0 \quad (i \in [k], j \in [n]) \quad \text{and} \quad \sum_{i=1}^k \xi_{ij} = 1 \quad (j \in [n]).$$

Further, the *weight* $\omega(C_i)$ of the cluster C_i is given by

$$\omega(C_i) = \sum_{j=1}^n \xi_{ij} \omega_j.$$

Note that the given grain scan (X, Y) naturally defines the integer clustering $G = (G_1, \dots, G_k)$ of the (discretized) grains. It is formally specified by the components $\xi_{ij}^G \in \{0, 1\}$ with $\xi_{ij}^G = 1$ if and only if $x_j \in G_i$. As all points carry the weight 1, cluster G_i has weight $\omega(G_i) = \kappa_i$, $i \in [k]$. It is, of course, this *grain scan clustering* that we want to represent by a suitable diagram $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$. Accordingly, a clustering C and a diagram $\mathcal{P} = (P_1, \dots, P_k)$ are called *compatible* if, for all $i \in [k]$,

$$P_i \supset \text{supp}(C_i) = \{x_j \in X : \xi_{ij} > 0\}.$$

In general, we cannot expect the grain scan clustering to be compatible with any APD. Hence, we resort to the problem of determining a clustering C^* that is “close” to G and which admits a compatible diagram.

A natural and simple condition is to require that the clusters of C^* and G have similar weight, i.e., for given $\epsilon_i \in (0, \infty)$ the *weight-constraints*

$$(1 - \epsilon_i)\kappa_i \leq \omega(C_i) \leq (1 + \epsilon_i)\kappa_i \quad (i \in [k])$$

should hold. In practice, the error terms ϵ_i may reflect the available resolution of the image (or the known or expected accuracy of the measurements).

Our overall rationale will, however, be as follows. First, we evoke linear programming duality to relate optimal constrained clusterings to APDs, as already suggested in [Alp+15]. Then, we apply this model to a rather small weighted point set which is obtained through the data compression techniques described in Section 4.3.2. Finally, we use the so obtained diagram to assign the points of the original data set to clusters. This yields an integer clustering, which we compare with the grain scan clustering to compute its classification error.

In the following, we will actually enforce the strict constraints $\omega(C_i) = \kappa_i$ on the compressed data set for all $i \in [k]$. The set of all such *weight-constrained clusterings* will be denoted by $\mathcal{C}_K(X, \Omega)$. Further, the given shape parameters $\mathcal{A}$ and S are incorporated in the *cost function*

$$\text{cost}_{\mathcal{A}, K}(X, \Omega, C) = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{\mathcal{A}_i}^2,$$

which is to be minimized. The idea behind this choice of objective function is to enforce some kind of “minimum energy principle” by rewarding the assignments of points to the closest site with respect to the ellipsoidal norms.

Altogether, we are considering the problem *weight-constrained anisotropic assignment* or, shorter,

WCA ASSIGNMENT:

Given k, X, Ω, K, S , find $C^* \in \mathcal{C}_K(X, \Omega)$ that minimizes $\text{cost}_{\mathcal{A}, K}(X, \Omega, C)$.

Note that the grain scan clustering G is feasible for the original unweighted data set. In general, the simple condition

$$\sum_{i=1}^k \kappa_i = \sum_{j=1}^n \omega_j$$

guarantees that $\mathcal{C}_K(X, \Omega) \neq \emptyset$, and we will, in the following, assume that this is the case. Hence, a finite minimum always exists for WCA ASSIGNMENT; it will be denoted by $\text{cost}_{\mathcal{A}, K}(X, \Omega)$,

i.e.,

$$\text{cost}_{\mathcal{A},\mathcal{K}}(X, \Omega) = \min_{C \in \mathcal{C}_{\mathcal{K}}(X, \Omega)} \text{cost}_{\mathcal{A},\mathcal{K}}(X, \Omega, C).$$

Next, note that WCA ASSIGNMENT can be formulated as the following linear program in the variables ξ_{ij}

$$\begin{aligned} \min_{(\xi_{ij})} \quad & \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{A_i}^2 \\ \text{(P)} \quad & \sum_{i=1}^k \xi_{ij} = 1 \quad (j \in [n]), \\ & \sum_{j=1}^n \xi_{ij} \omega_j = \kappa_i \quad (i \in [k]), \\ & \xi_{ij} \geq 0 \quad (i \in [k], j \in [n]). \end{aligned}$$

Let us remark that in the case that all weights coincide, and in particular, if all are 1, the coefficient matrix of (P) is *totally unimodular*, hence each basic feasible solution is integer; see e.g. [Sch86] for background information. The dual linear program

$$\begin{aligned} \text{(D)} \quad & \max_{(\eta_j, \gamma_i)} \quad \sum_{j=1}^n \omega_j \eta_j - \sum_{i=1}^k \kappa_i \gamma_i \\ & \eta_j \leq \|x_j - s_i\|_{A_i}^2 + \gamma_i \quad (i \in [k], j \in [n]), \end{aligned}$$

in the variables η_j and γ_i encodes proximity conditions involving the diagram functions based on $\|x - s_i\|_{A_i}^2$. Recall that, due to duality, the optima in (P) and (D) coincide. Further, we have the following theorem.

Theorem 4.1 (Special case of [BGK17, Thm. 1])

Let $(k, X, \Omega, \mathcal{K}, \mathcal{A}, S)$ be an instance of WCA ASSIGNMENT, and $C \in \mathcal{C}_{\mathcal{K}}(X, \Omega)$. Then C is an optimizer of the linear program (P) if and only if there exist size parameters Γ such that C and the diagram $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$ are compatible.

According to Theorem 4.1, an optimal clustering C^* results in an optimal diagram $\mathcal{P}^*$. Hence, we can apply any standard linear programming algorithms to compute optimal sizes. In Section 4.5 we actually use the dual simplex algorithm. Note, however, that optimality is measured here with respect to the objective functions of (P) and (D), and it is not clear in the first place how the cost of a clustering translates into the classification error which is to be minimized in APD CLASSIFICATION($\mathcal{A}, S$). See [FG22b] for related stability and instability results. Hence, in effect, the clustering-based algorithms address APD CLASSIFICATION($\mathcal{A}, S$) only indirectly.

4.3.2 Data compression for accelerating the linear program

If the linear program (P) is based on the original (unweighted) ground set X of the image, say, on the points of $1/\nu \cdot \mathbb{Z}^d$ in $[0, 1]^d$ for some $\nu \in \mathbb{N}$, then it involves $k \cdot (\nu + 1)^d$ variables.

If, for instance, in 3D, $\nu = 9999$ and $k = 50$, $(\mathbf{P})$ has already $5 \cdot 10^{13}$ variables. Of course, as one would expect that $\xi_{ij} = 0$ whenever s_i is not among the first few closest sites to x_j , preprocessing would allow us to reduce the influence of k to a manageable size. It is, however, $|X|$ which affects the number of variables the most.

Image supports and coresets

In the following, we will reduce the number of variables considerably while retaining almost the initial accuracy. Our approach is based on the observation that the point set (X, Ω) is only a means for obtaining the desired diagram but not important in its own right. In fact, once we obtain the diagram $\mathcal{P}$, we can easily generate an image at any desired resolution from it. Hence, we can, in principle, use *any* non-empty finite subset $\tilde{X} \subset [0, 1]^d$ for computing diagrams through clustering – as long as we associate suitable weights to the points to accommodate the weight constraints. We will call such weighted sets $(\tilde{X}, \tilde{\Omega})$ *image supports*.

Naturally, some such sets will be better suited than others, and we will speak of *coresets* $(\tilde{X}, \tilde{\Omega})$ if these sets can be derived from (X, Ω) in such a way that the relative deviation between $\text{cost}_{\mathcal{A}, \mathbf{K}}(X, \Omega)$ and $\text{cost}_{\mathcal{A}, \mathbf{K}}(\tilde{X}, \tilde{\Omega})$ can always be suitably bounded in terms of a given precision. Here, we are interested in choosing the image supports $(\tilde{X}, \tilde{\Omega})$ as part of the “experimental setup”, i.e., prior to scanning or taking characteristic measurements of the polycrystal. Thus, this setting requires the condition for *all* sets of k sites.

More precisely, let $(\tilde{X}, \tilde{\Omega})$ be a weighted data set, $\epsilon \in (0, 1/2]$, $\delta \in [1, \infty)$. Then, following [FG22a], $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -*coreset* for $(k, X, \Omega, \mathbf{K})$ if there exists a mapping $f : \mathcal{C}_{\mathbf{K}}(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_{\mathbf{K}}(X, \Omega)$, called *extension*, and real constants Δ^+, Δ^- , referred to as Δ -*terms* or *offsets*, with $0 \leq \Delta^+ \leq \delta \cdot \Delta^-$ such that the following two conditions hold for all sets of k sites and all clusterings $\tilde{C} \in \mathcal{C}_{\mathbf{K}}(\tilde{X}, \tilde{\Omega})$:

- (a) $(1 - \epsilon) \text{cost}_{\mathcal{A}, \mathbf{K}}(X, \Omega, f(\tilde{C})) \leq \text{cost}_{\mathcal{A}, \mathbf{K}}(\tilde{X}, \tilde{\Omega}, \tilde{C}) + \Delta^+$,
- (b) $\text{cost}_{\mathcal{A}, \mathbf{K}}(\tilde{X}, \tilde{\Omega}) + \Delta^- \leq (1 + \epsilon) \text{cost}_{\mathcal{A}, \mathbf{K}}(X, \Omega).$

If $\delta = 1$ we will simply speak of an ϵ -*coreset*.

Let us point out that other notions of coresets have been introduced for different problems; see e.g. [BLK17; FSS13; FSS20; Fic+13; HPK07; SW18]. The definition used here is specifically customized for WCA clusterings and assignments. It guarantees that approximate solutions of the constrained clustering problems on coresets can be turned into approximate solutions on the original data set; see [FG22a, Thm. 3.5].

In order to practically use coresets for constrained clustering, they need to be small and the extension g must be quickly computable. While, in our context, the coreset constructions (given below) are important, the extension does not play any role, and the coreset conditions are relevant for theoretical reasons only, as they guarantee any desired accuracy at least with respect to the clustering cost. In fact, clustering is not used here in its own right but only as a method for computing diagrams via linear programming duality. Accordingly, we will determine the assignment of the original data points directly from the diagram computed on the coreset data and compare the obtained clustering C with the grain scan clustering G to quantify the classification error; see Section 4.5. In particular, the weights of the so obtained clusters will, in general, deviate from the specified target weights κ_i .

In the following, we will first present two different coreset constructions which rely, in addition to $\mathcal{A}$ and S , only on the measurements $\kappa_1, \dots, \kappa_k$ and, hence, can be used in a much

more general practical context. Subsequently, we will then, however, also consider an image support based on the grain scan clustering $G = (G_1, \dots, G_k)$ to show how far the concept can be pushed in principle.

Pencil coresets

The following result shows that coresets exist and that they can be profitably used to accelerate GBPD computations. Let $\lambda^-(\mathcal{A})$ and $\lambda^+(\mathcal{A})$ denote the smallest and largest eigenvalue of all matrices in $\mathcal{A}$, respectively.

Theorem 4.2 (Special case of [FG22a, Thm. 2.3])

For any instance of WCA ASSIGNMENT and for any $\epsilon \in (0, 1/2]$ and

$$\delta \geq \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})},$$

there is an (ϵ, δ) -coreset $\tilde{X}$ of size

$$|\tilde{X}| \in \mathcal{O}\left(\frac{k^2}{\epsilon^{d+1}}\right).$$

The theorem shows that it suffices to use relatively small point sets for clustering whose “sparseness” is independent of the initial point set X and depends only on k and ϵ . More importantly, Theorem 4.2 is constructive. As the known constructions “live” on pencils of lines in $\mathbb{R}^d$ issuing from a certain set of points; they are called *pencil coresets*. We will report on our implementation of such pencil coresets for different error parameters in Section 4.5. Here we will briefly indicate their structure, which is based on a design by [HPK07] for unconstrained clustering. Its generalization to WCA clusterings and the improved bound are due to [FG22a].

For technical simplicity, we will focus on the Euclidean case and, as this is the relevant situation here, assume that the “source points” coincide with given sites $s_1, \dots, s_k$. For a visualization in 2D see Fig. 4.2. The lines of the i th pencil are chosen in such a way that their points of intersection with the unit sphere centered at the site s_i are “evenly distributed”. Note that this is closely related to coverings of the unit sphere by congruent spherical caps. See, e.g., [Mat02] for general results on ϵ -nets and [Fej72] for sparse coverings of the Euclidean 2-sphere by specific numbers of congruent spherical caps.

Then, we assign each point of the data set to its closest site, breaking ties arbitrarily, and project it onto a line of the corresponding pencil closest to it, again breaking ties arbitrarily. After this projection process, all points lie on lines, and in the remaining step, each line is treated independently. Note that this is possible due to the additivity of the cost function with respect to the data points.

Each such line is now partitioned into batches. Finally, all points in one batch B are replaced by the batch’s centroid $c(B)$ and assigned the sum of weights of all batch points as its weight. Of course, this process evokes a *batch error* $\sum_{x \in B} \|x - c(B)\|_A^2$.

If the lengths of the batches are chosen appropriately, this process results in the desired (ϵ, δ) -coreset. See [FG22a] for the details of the proof.

Resolution coresets

While the construction in the previous section works for general data sets (X, Ω) , the pencils ignore the grid structure of the given grain scan. In particular, the “density” of the image support varies with the distance to the sites. For voxel-based images, it seems more natural and, actually, interesting in its own right, to consider image supports that pertain to a (coarser) grid structure and which can still be interpreted in terms of the image resolution.

In the following, we will briefly introduce the concept of resolution coresets. For technical simplicity, we will assume that the voxels are boxes whose edge lengths are powers of $1/2$. Our exposition follows [FGK22].

For $i \in [d]$, and $\rho = (\rho_1, \dots, \rho_d) \in \mathbb{N}^d$ we consider the uniform partitioning of the interval $[0, 1]$ on the i th coordinate axis into 2^{ρ_i} intervals of equal length. Then each voxel is the Cartesian product of one of the intervals on each axis. More formally, with

$$x_{j_1, \dots, j_d}^\rho = \begin{pmatrix} \frac{1}{2^{\rho_1+1}} + \frac{j_1}{2^{\rho_1}} \\ \vdots \\ \frac{1}{2^{\rho_d+1}} + \frac{j_d}{2^{\rho_d}} \end{pmatrix},$$

the voxels are obtained as

$$x_{j_1, \dots, j_d}^\rho + \frac{1}{2^{\rho_1+1}}[-1, 1] \times \dots \times \frac{1}{2^{\rho_d+1}}[-1, 1]$$

for all choices of $j_1, \dots, j_d$ with

$$j_1 \in \{0, \dots, 2^{\rho_1} - 1\}, \dots, j_d \in \{0, \dots, 2^{\rho_d} - 1\}.$$

We collect the points $x_{j_1, \dots, j_d}^\rho$ in the data set X^ρ . Note that the points are centroids of the voxels and constitute the original data set, i.e., $X^\rho = X$. Of course, $n = |X^\rho| = 2^{\rho_1} \cdot \dots \cdot 2^{\rho_d}$. We will refer to the vector $(\rho_1, \dots, \rho_d)$ as the *resolution* of the image. Note that, if all ρ_i coincide with some given $\bar{\rho} \in \mathbb{N}$ then

$$X^\rho = \frac{1}{2^{\bar{\rho}+1}} \mathbb{1} + \left(\left(\frac{1}{2^{\bar{\rho}}} \mathbb{Z}^d \right) \cap [0, 1)^d \right),$$

where $\mathbb{1}$ is the all-ones vector $(1, \dots, 1)^\top$, i.e., the point set consists of all points in $[0, 1]^d$ of a translate of the integer lattice of side $1/2^{\bar{\rho}}$. Naturally, a lower resolution is obtained by decreasing some or all ρ_i , resulting in sparser image supports. The question is how small we can choose a resolution $\tau \in \mathbb{N}^d$ with $\tau_i \leq \rho_i$ such that X^τ , together with appropriate weights, is a coreset.

As it turns out, it suffices to consider a number of points on each axis which depends only on k and ϵ .

Theorem 4.3 ([FGK22, Thm. 1])

Let $\rho \in \mathbb{N}^d$, (k, X^ρ, Ω, K) be an instance of WCA CLUSTERING and $\epsilon \in (0, 1/2]$. Then, there is a $\bar{\tau} \in \mathbb{N}$ with

$$\bar{\tau} \leq \frac{2^{8/3} k}{\epsilon^{2/3}},$$

such that, for $\tau = (\bar{\tau}, \dots, \bar{\tau}) \in \mathbb{N}^d$, the set X^τ together with appropriate weights is an ϵ -coreset of size

$$|X^\tau| \leq \left(\frac{2^{8/3} k}{\epsilon^{2/3}} \right)^d.$$

Such a set X^τ is referred to as a *resolution coreset*. Theorem 4.3 allows us to reduce the resolution of the scan to a size that is independent of the original resolution and only depends on the number of grains and the desired accuracy (measured in terms of the cost of the produced clustering). Hence, we can use grain scans of relatively low resolution, solve (P) for the correspondingly small data set, and then obtain a diagram representation via (D). Recall that we can, ultimately, use the diagram to produce a grain image of any (arbitrarily high) desired resolution.

Let us briefly address the difference between the bounds in Theorem 4.2 and Theorem 4.3 to place the results into perspective. For example, when $k = 50$ and $\epsilon = 0.01$, not taking constants (hidden, in Theorem 4.2, in the Landau notation) into account, the coreset size guaranteed by Theorems 4.2 and 4.3 in 3D is $2.5 \cdot 10^{11}$ and $1.25 \cdot 10^9$, respectively. Note, however, that Theorems 4.2 and 4.3 give *worst-case upper* bounds which are not known (and not even conjectured) to be best possible. As we will see in Section 4.5, the image supports can indeed be reduced much more radically in our practical computations without sacrificing much of the accuracy.

Let us repeat that, while the coreset property bounds the error in clustering costs it does not directly provide a theoretical bound for the classification error of the corresponding diagram for $\text{APD CLASSIFICATION}(\mathcal{A}, S)$. The practical results in Section 4.5 indicate, however, that the “indirect” quality guarantee which comes with the coreset property also leads to a favorable classification behavior.

Grain scan-based interior removal

The previously constructed pencil and resolution coresets make only use of certain characteristics of grain maps. They can, of course, be derived from a given grain scan (X, Y) , but, in practice, they are often much more easily available through direct tomographic measurements. While this is an important feature for practical applications it is, of course, natural to ask how much is actually lost by not using the full grain scan information. More precisely, using $G = (G_1, \dots, G_k)$, how much further can the size of image supports be reduced while still constituting coresets, i.e., while still facilitating quite accurate clusterings in terms of their costs. Note that this is not just a theoretical issue but also highly relevant whenever given grain scan representations need to be turned into diagram representations to facilitate image post-processing and analysis tasks.

Suppose that, under ideal conditions, the grain growth process indeed results in a diagram. Then we may assume that the given grain scan clustering G is close, up to minor imperfections, to an optimal clustering C^* .

Before continuing, let us mention that such an assumption is in line with several common stochastic grain growth models, including Poisson-Voronoi and Laguerre [Chi+13; Oka+00]. These methods assume growth from a set of point germs $\mu_1, \dots, \mu_k$ with a normal distribution $\mathcal{N}(\mu_i, \Sigma_i)$ modelling the probability of points being contained in the i th grain. In these cases,

the shape parameters are chosen according to maximum likelihood resulting in $s_i = \mu_i$ and $A_i = \Sigma_i^{-1}$. See, e.g., [Alp+23a; Alt+14] for more information.

Under such a “closeness” assumption it seems reasonable to discard all points that are sufficiently far away from the “grain boundaries” of G . More formally, we choose a suitable metric d and parameters $\delta_i \in (0, \infty)$, and define the δ_i -interior of G_i by

$$G_i(\delta_i) = \{x \in G_i : \mathsf{d}(x, x') \geq \delta_i \ \forall x' \notin G_i\}.$$

In the grid based situation of the previous paragraph it seems natural to use for d a variant of the 1-norm normalized by the different resolutions on the axes, i.e., for $z = (\zeta_1, \dots, \zeta_d)^\top$

$$\|z\|_1 = \sum_{i=1}^d 2^{\rho_i} \zeta_i.$$

The corresponding norm-induced metric then counts the “grid distance” which is the length of a shortest path in the underlying grid graph. In Section 4.5, we will use this metric and refer to it as *grid graph distance*. Note, that (for any standard metric) $G_i(\delta_i)$ can be computed efficiently. See [FG22a] for a more comprehensive account on image supports obtained this way.

The parameters δ_i control the size of the interior, and a larger δ_i will result in a smaller δ_i -interior and, hence, in a larger image support but potentially also higher accuracy. The specific choices of the δ_i reflect the initial confidence that G is already close to an optimal clustering. The deviation from G of the grain scan induced by the computed diagram $\mathcal{P}$ may be viewed as an indicator of the state of the process, i.e., of how far it is from an energy-minimal state which is indeed characterized by the underlying model.

For the computations in Section 4.5, we employ image supports obtained by coreset techniques combined with interior removal. Any specific algorithm of the so obtained class of methods will be referred to as *sparse-GBPD* or, short, *s-GBPD*. As we will show, suitable such choices lead to substantial accelerations of diagram computations on real-world data.

4.4 A direct model for APD CLASSIFICATION

The algorithms of the previous section are based on estimates for the matrices and sites and treat APD CLASSIFICATION($\mathcal{A}, S$) indirectly via clustering. We will now propose an optimization model which addresses APD CLASSIFICATION directly. As it optimizes over all three characteristic parameters $\mathcal{A}$, S , and Γ , the larger set of diagrams to choose from will allow for a better fit of (X, Y) . The higher accuracy comes, however, at the price of computational costs which are substantially higher than those of the methods described in Section 4.3. Nonetheless, combining the model with our previous point reduction techniques enables at least its computation for the real-world data set of Section 4.5.

This section has, indeed, a somewhat different focus: On the one hand, it provides a “benchmark method” for a better assessment of the quality of fit produced by other methods for appropriately chosen test cases. On the other hand, it may be helpful for “quantifying the physical state” of a prematurely terminated growth process in terms of the achieved fit. As expected and exemplified in Section 4.5, the new algorithm will only be practically useful if computation time is not the main concern.

The technique that we will now introduce is based on the paradigm of *support vector machines* from data analysis and requires knowledge of (a relevant part of) the grain scan (X, Y) . See, e.g., [SS02] for additional background information on machine learning.

Let, again $G = (G_1, \dots, G_k)$ be the grain scan clustering. If there exists a diagram $\mathcal{P}(\mathcal{A}, S, \Gamma)$, compatible with G , its defining functions $h_i(x) = \|x - s_i\|_{A_i}^2 + \gamma_i$ will separate the points of the clusters, i.e., for $j \in [n]$ and $i, \ell \in [k]$ with $\ell \neq i$,

$$h_i(x_j) \leq h_\ell(x_j) \quad \text{for each } x_j \in G_i.$$

Since, in general, real-world scans will not admit a perfectly fitting diagram, it is natural to relax these conditions and resort to the optimization problem

$$\begin{aligned} \min_{(\zeta_j)} \quad & \sum_{j=1}^n \zeta_j \\ h_i(x_j) \leq \quad & h_\ell(x_j) + \zeta_j \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i) \\ \zeta_j \geq \quad & 0 \quad (j \in [n]). \end{aligned}$$

Here, the non-negative error terms ζ_j allow for potential misclassification, and the objective minimizes the accumulated errors. As, however, we can multiply all matrices A_i and sizes γ_i by a common positive factor without changing the diagram, the infimum of the objective function will always be 0, independently of the fit of the diagram. There are various possibilities to break this invariance. Our approach, reflects the assumption that points near the grain centroids are already correctly classified by the grain scans and classification errors occur only near the boundaries. This will be modeled in terms of the δ_i -interior for suitably chosen $\delta_i \in (0, \infty)$ introduced in the previous section, resulting in the optimization problem

$$\begin{aligned} \min_{(\zeta_j)} \quad & \sum_{j=1}^n \zeta_j \\ h_i(x_j) \leq \quad & h_\ell(x_j) - 1 \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i(\delta_i)) \\ h_i(x_j) \leq \quad & h_\ell(x_j) + \zeta_j \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i \setminus G_i(\delta_i)) \\ \zeta_j \geq \quad & 0 \quad (j \in [n], x_j \in G_i \setminus G_i(\delta_i)). \end{aligned}$$

In this model, all points in the δ_i -interior of G_i are separated strictly. Note that the specific choice of 1 for the “margin” is arbitrary as long as it is positive. Let us point out that, while a positive value of ζ_j in the optimum indicates that the point x_j is misclassified, the objective function does not actually *count* the number of misclassified points. In fact, using indicator variables to express the classification error would result in an integer program which can no longer be solved for appropriate sizes of grain maps within acceptable time.

The above optimization problem has nonlinear constraints as the optimization extends over the entries in A_i , s_i and γ_i and

$$h_i(x_j) = \|x_j - s_i\|_{A_i}^2 + \gamma_i = x_j^T A_i x_j - 2s_i^T A_i x_j + s_i^T A_i s_i + \gamma_i,$$

where only the points x_j are given. We will therefore apply a lifting technique to linearize the problem by introducing new variables for each relevant expression. First, we use the notation

$$x_j = \begin{pmatrix} (x_j)_1 \\ \vdots \\ (x_j)_d \end{pmatrix}, \quad A_i = \begin{pmatrix} (A_i)_{1,1} & \dots & (A_i)_{1,d} \\ \vdots & & \vdots \\ (A_i)_{d,1} & \dots & (A_i)_{d,d} \end{pmatrix}$$

and set

$$a_i = -2A_i s_i, \quad \alpha_i = s_i^\top A_i s_i + \gamma_i$$

to encode all parameters in the *extended parameter vector*

$$\mathfrak{A}_i = \left(\alpha_i, a_i^\top, (A_i)_{1,1}, 2(A_i)_{1,2}, \dots, 2(A_i)_{d-1,d}, (A_i)_{d,d} \right)^\top \in \mathbb{R}^{1+d+\frac{1}{2}d(d+1)}.$$

Note that (A_i, s_i, γ_i) can easily be recovered from $\mathfrak{A}_i$. In fact, the entries of A_i can be read off directly, then $s_i = -1/2A_i^{-1}a_i$ and, finally, $\gamma_i = \alpha_i - s_i^\top A_i s_i$. If we now introduce, for each point x_j , its *extended coefficient vector*

$$\mathfrak{X}_j = \left(1, (x_j)_1, \dots, (x_j)_d, (x_j)_1^2, (x_j)_1(x_j)_2, \dots, (x_j)_1(x_j)_d, \dots, (x_j)_{d-1}(x_j)_d, (x_j)_d^2 \right)^\top,$$

we can write the separation conditions in terms of the linear expression

$$h_i(x_j) = \mathfrak{A}_i^\top \mathfrak{X}_j.$$

Accordingly, we will address APD CLASSIFICATION via the *direct linear programming model*

$$\begin{aligned} & \min_{(\zeta_j)} \sum_{j=1}^n \zeta_j \\ \text{(DiM)} \quad & \mathfrak{A}_i^\top \mathfrak{X}_j - \mathfrak{A}_\ell^\top \mathfrak{X}_j + 1 \leq 0 \quad \begin{array}{l} (j \in [n], i, \ell \in [k], \ell \neq i, \\ x_j \in G_i(\delta_i)) \end{array} \\ & \mathfrak{A}_i^\top \mathfrak{X}_j - \mathfrak{A}_\ell^\top \mathfrak{X}_j - \zeta_j \leq 0 \quad \begin{array}{l} (j \in [n], i, \ell \in [k], \ell \neq i, \\ x_j \in G_i \setminus G_i(\delta_i)) \end{array} \\ & \zeta_j \geq 0 \quad (j \in [n], x_j \in G_i \setminus G_i(\delta_i)). \end{aligned}$$

In the following we will also use the acronym DiM for any algorithm that solves (DiM).

There is one additional issue to address. In order to derive diagram parameters from a solution $(\mathfrak{A}_1^*, \dots, \mathfrak{A}_k^*)$ of (DiM), we need to guarantee that the obtained matrices $A_1^*, \dots, A_k^*$ are symmetric and positive definite. Otherwise, the computations may not result in an anisotropic power diagram. While the symmetry of the matrices is built into the model, there is, however, no precaution taken that the A_i^* produced via DiM are indeed positive definite.

To obtain appropriate matrices, we will use the fact that the constraints in (DiM) are invariant under the operation which replaces all entries $(A_i)_{\ell,\ell}$ in $\mathfrak{A}_i$ by $(A_i)_{\ell,\ell} + \beta$ for any arbitrary $\beta \in \mathbb{R}$ while keeping all other entries as before. In fact, we simply add and subtract

the term $\beta x_j^\top x_j$ in the first two groups of constraints. Hence, neither the constraints nor the objective function values of (DiM) change. In effect, this operation replaces the last $\frac{1}{2}d(d+1)$ entries in $\mathfrak{A}_i$ by $\bar{A}_i = A_i + \beta E_d$.

So, suppose that not all of the resulting matrices A_i are already positive definite. Then we can simply switch over to $\bar{A}_i = A_i + \beta E_d$ for all $i \in [k]$ and some suitable $\beta \in (0, \infty)$ so that all $\bar{A}_i$ become positive definite. We use the matrices $\bar{A}_i$ now for the subsequent computation of the sites and the sizes and set

$$\bar{s}_i = -1/2\bar{A}_i^{-1}a_i, \quad \bar{\gamma}_i = \alpha_i - s_i^\top \bar{A}_i \bar{s}_i.$$

Then $(\bar{A}_i, \bar{s}_i, \bar{\gamma}_i)$ ($i \in [k]$) yield, indeed, a desired diagram $\mathcal{P}(\bar{\mathcal{A}}, \bar{\mathcal{S}}, \bar{\Gamma})$ with the same objective function value in (DiM).

Let us close by pointing out that (DiM) can be combined with the image supports described in Section 4.3.2. Indeed, for practical computations one can largely thin out X first and, in particular, require strict separation only for data points in certain rings $G_i(\delta_i^-) \setminus G_i(\delta_i^+)$ with $\delta_i^- < \delta_i^+$. In Section 4.5 we will employ such accelerating techniques but yet see that (DiM) is significantly more time-consuming than (P).

4.5 Computational results

We will now evaluate the introduced techniques with respect to computation time and accuracy using a 3D real-world grain scan from [LP11]. After briefly describing the computational setup, we will first assess the achievable degree of sparseness of the image supports for s-GBPD. This allows us to choose appropriate “sparseness parameters” to compare s-GBPD and DiM for a range of different quality criteria. This comparison will illustrate the degree of fit produced by the former at a much lower computational cost than the latter.

4.5.1 Computational setup

Our computational study is based on data provided by [LP11]; see also the description in [Alp+15]. It was obtained by a synchrotron micro-tomograph experiment conducted on a metastable beta titanium alloy (Ti β 21S). The material was scanned with a volume of $240\mu\text{m} \times 240\mu\text{m} \times 420\mu\text{m}$ at a resolution resulting in a voxel edge length of $0.7\mu\text{m}$. It comprises 591 grains, 211 are interior to the polycrystal, i.e., they are entirely surrounded by other grains.

We hence have a grain scan (X, Y) with

$$n = |X| = 339 \times 339 \times 599 = 68,837,679 \quad \text{and} \quad k = 591.$$

In the following we refrain from normalizing X and report all results for the original physical dimension stated above. All computations are carried out in 3D. The images depict 2D slices parallel to the plane of the first two coordinates. One such example is shown in Fig. 4.1.

Let us point out that the larger blue patch in the left lower corner and the close-by tiny blue patch surrounded by the green area actually belong to the same grain. While this can be seen in the 3D image, its 50th slice depicts two components. So, let us emphasize again that all subsequent 2D images are for illustration purposes only while all reported results have been fully computed in 3D. This contrasts most computations in [Alp+15] and, as a matter



Figure 4.1: Slice no. 50 of the data set. Neighboring grains are colored in different colors.

of fact, 3D problems of the size considered here are far beyond the capability of the original GBPD-algorithm introduced there.

The reported running times were measured on a virtual machine with ten virtual CPUs corresponding to five Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz CPUs and 45GB of memory (i.e., one CPU processes two virtual CPUs). For better comparison, we solved all linear programs using the dual simplex algorithm on only one virtual CPU. Let us point out, however, that the sparse models, which we show to be sufficient, also run on standard machines with less than 16GB of memory.

4.5.2 s-GBPD: Sparseness vs. fit

The coreset constructions in Theorems 4.2 and 4.3 allow us to work with sample sizes that only depend on k and ϵ without much affecting the clustering cost. Grain-based interior removal can further reduce the size of the image support. We will now study experimentally the effect of sparseness on the quality of the generated diagram.

Relative fit and relative cluster weight error

In the following, we measure the quality of fit. More precisely, let $\mathcal{P} = (P_1, \dots, P_k)$ be the diagram obtained by the algorithm and set

$$\hat{C}_i = \{x_j \in X : x_j \in \text{int}(P_i)\} \quad (i \in [k]), \quad \hat{C} = (\hat{C}_1, \dots, \hat{C}_k).$$

Let us point out that $\hat{C}$ is not a clustering of X but only of $X \cap \bigcup_{i=1}^k \text{int}(P_i)$ (and this is why we use the notation $\hat{C}$ rather than C). We define the *relative fit* Φ_G of $\hat{C}$ as

$$\Phi_G(\hat{C}) = \frac{1}{n} \left| \bigcup_{i=1}^k G_i \cap \hat{C}_i \right|.$$

Note that the points on the cell boundaries are not assigned to any clustering and, hence, count as misclassified.

Also, we will compute the *relative cluster weight error* Ψ_G defined by

$$\Psi_G(\hat{C}) = \frac{1}{n} \sum_{i=1}^k |\kappa_i - \omega(\hat{C}_i)|.$$

Pencil coresets

In this setup, the sites S and the matrices $\mathcal{A}$ are given and we consider pencils issuing from the centroids of the G_i whose lines are distributed with respect to the ellipsoids $s_i + \{x : \|x\|_{A_i} \leq 1\}$.

While all computations are fully 3D, but this is hard to visualize, Fig. 4.2 illustrates the construction in 2D with rays emanating from the centroids of the depicted 2D grains.



Figure 4.2: APDs from pencil coresets. From left to right: Computed diagrams (cell boundaries shown in black) based on decreasing image supports (the non-white and non-black colored pixels in the respective images). The diagram in the left image uses the full image support (the full available data), while the others use image supports living on pencils of decreasing density.

For about 100,000 points, we obtain already a relative fit of 0.93. Enlarging the image support improves Φ_G only in the third decimal digit. The relative cluster weight error Ψ_G benefits slightly more from increasing the density, from 0.025 for 100,000 to 0.02 for 200,000 points. Fig. 4.3 depicts the dependency of the relative fit and the relative cluster weight error on the density of the sample support. While in Theorem 4.2, both, the number of rays and the batch error $\sum_{x \in B} \|x - c(B)\|_A^2$ are functions of k and ϵ , we control these numbers in practice separately.

More precisely, first the number of rays are selected and each point of X is projected on the appropriate ray, as described in Section 4.3.2. Then the points of each ray are partitioned into batches B and replaced by the batches' centroids $c(B)$. We do this dynamically so as to control the batch errors. Accordingly, Fig. 4.3 also shows the influence of the batch error and the number of rays (indicated by the size and color of the points, respectively) on the relative fit and the relative weight error.

In general, the computational results suggest that, within a reasonable range, the number of rays may be more important than the batch error. This observation supports the intuition that the shape information about the grains is better captured by additional rays than by more points on each ray.

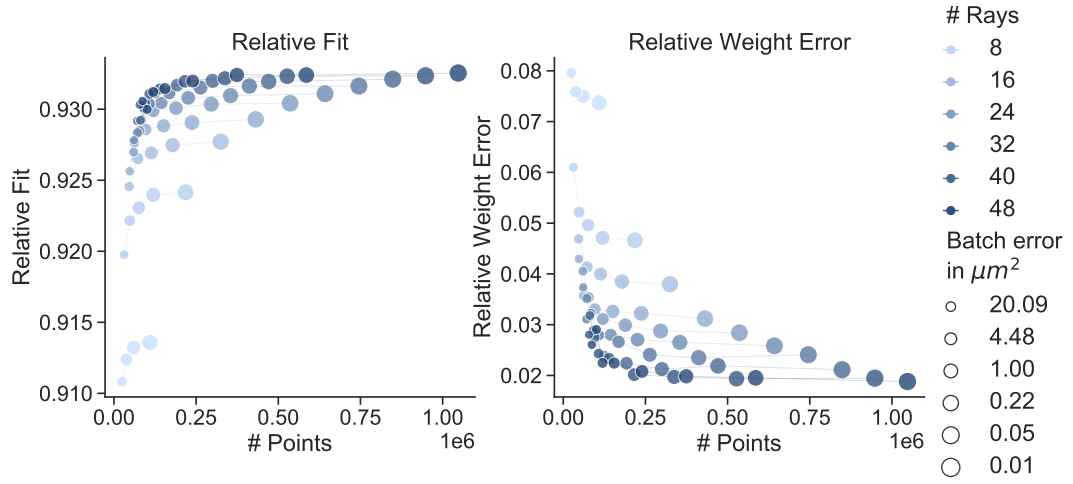


Figure 4.3: Relative fit (left) and relative cluster weight error (right) for pencil coresets evaluated for different cardinalities of image supports. The numbers of rays and the batch errors are signified by color and size. Also, the behavior for a fixed number of rays is indicated by connecting the corresponding points.

Resolution coresets

The results of a similar evaluation for the resolution coresets are illustrated in Fig. 4.4.

Again, we obtain a relative fit of 0.93 for image supports above 100,000 points. However, the resolution coreset seems to cause a relative cluster weight error which is only half of that for pencil coresets, i.e., 0.01 instead of 0.025 for the latter. With increasing resolution, Ψ_G decreases further to 0.0022 for 1 million points. Fig. 4.5 depicts the details.

The experimental results suggest that perturbing the grid structure in the image support

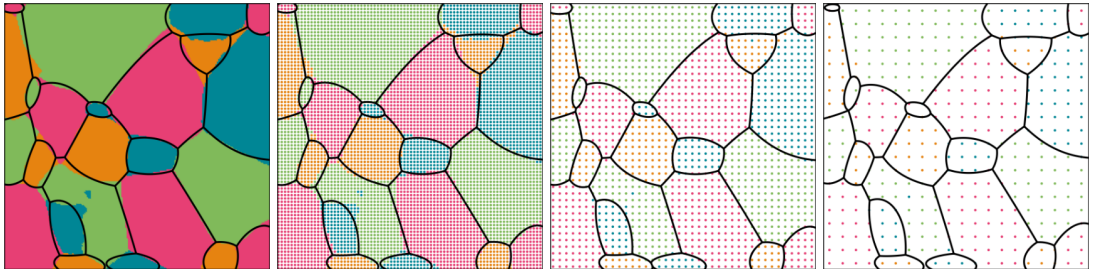


Figure 4.4: APDs from resolution coresets. From left to right: Computed diagrams (cell boundaries shown in black) based on decreasing image supports (the non-white and non-black colored pixels in the respective images). The diagram in the left image uses the full image support (the full available data), while the others use image supports living on resolution coresets of decreasing density.

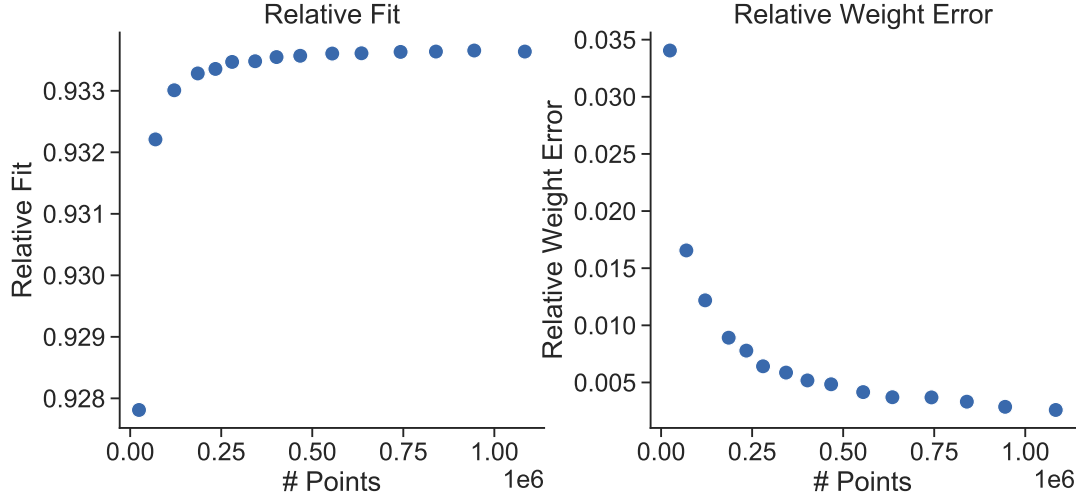


Figure 4.5: Relative fit (left) and relative cluster weight error (right) for resolution coresets evaluated for different cardinalities of image supports.

yields, at the same level of sparsity, diagrams which are closer to the optimal diagram for the original instance.

Additional grain scan-based interior removal

In our practical evaluation of grain scan-based interior removal, we used a uniform setting, i.e., $\delta_i = \bar{\delta}$ for all $i \in [k]$ and employed the grid graph distance. See Fig. 4.6 for a visualization in 2D. (Here, and in Figure 4.7, we used the values $\bar{\delta} = 2, 6, 10$ to enhance the visual impression; the computations were done with a variety of different values, and we report the results for $\bar{\delta} = 2, 3, 4$ as these are most relevant.)



Figure 4.6: APDs for different degrees of grain scan-based interior removal (from left to right: none, and $\bar{\delta} = 10, 6, 2$). Shown are the computed diagrams (cell boundaries in black) based on the decreasing image supports (the non-white and non-black colored pixels in the respective images).

In order to be able to preserve the size constraints in (P) for the reduced image support,

we represent each removed $\bar{\delta}$ -interior $G_i(\bar{\delta})$ by a single point at G_i 's centroid, weighted by the cardinality of the removed interior points. While, particularly for large grains, a considerable number of points may be omitted, the total length of the unmodified “grain boundaries” is typically quite large. Hence, as a sole thinning strategy, interior removal is insufficient for reducing the image support to render the linear program tractable in practice.

When combined with the previous coresets strategies, grain scan-based interior removal can, however, add a significant additional contribution. Pencil and resolution coresets still yield many points in each grain's interior, which can be removed for further speeding up (P).

Figure 4.7 illustrates the combination of the coresets with interior removal in 2D. Again, the resulting diagrams are, as theory suggests, very similar to the original image.

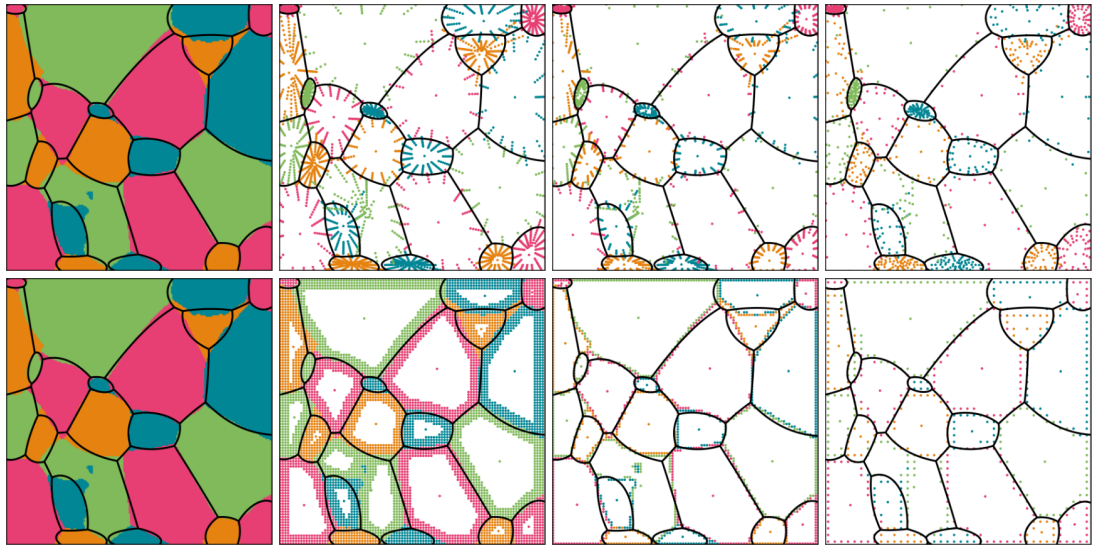


Figure 4.7: Pencil (top) and resolution coresets (bottom) combined with interior removal (from left to right: none, and $\bar{\delta} = 10, 6, 2$). Shown are the computed diagrams (cell boundaries in black) based on the decreasing image supports (the non-white and non-black colored pixels in the respective images).

In the experimental study, we investigated the effect of different parameter choices on the relative fit and the relative cluster weight error. Since we discussed the dependence of Φ_G and Ψ_G on the coresets parameters already, we focus on the dependence on the parameter $\bar{\delta}$. Recall that the $\bar{\delta}$ -interior $G_i(\bar{\delta})$ decreases with increasing $\bar{\delta}$. Fortunately, already a grid graph distance of $\bar{\delta} = 2$ produced excellent fits. Best results were obtained for $\bar{\delta} = 4$.

Figures 4.8 and 4.9 show the effect of different choices for the parameters on the relative fit and the relative cluster weight error.

In comparison, the combined thinning process based on resolution coresets outperformed that of pencil coresets: Its relative fit is higher, and its relative cluster weight error is smaller for all investigated coresets sizes. The difference is, however, generally only in the second or third digit.

All in all, using as few as 85,828 points was sufficient to obtain very high relative fit and

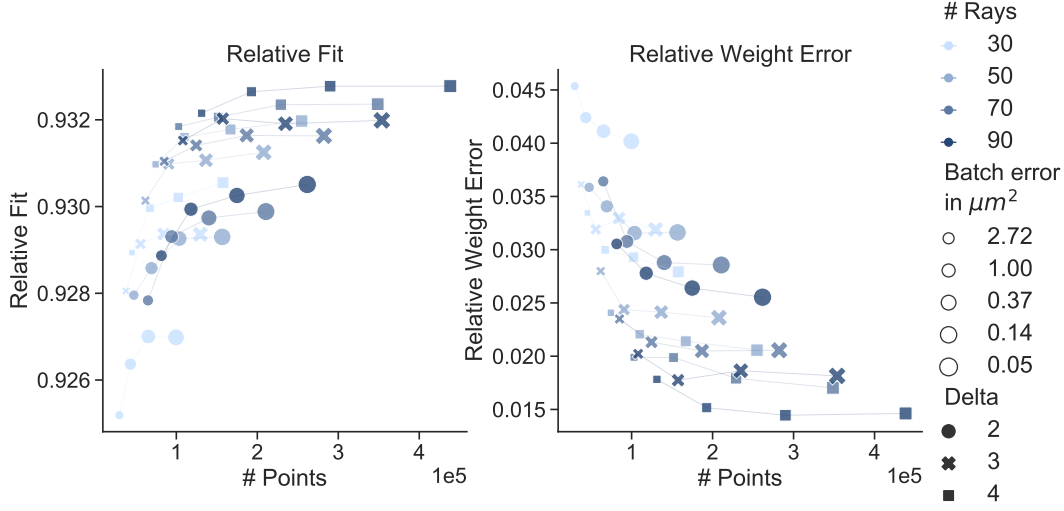


Figure 4.8: Combination of pencil coresets with interior removal: Dependence of $\Phi(G)$ and Ψ_G on the size of the resulting image support. Again, the color indicates the number of rays, the size refers to the batch error while the shape of the symbol signifies the different values of $\bar{\delta}$.

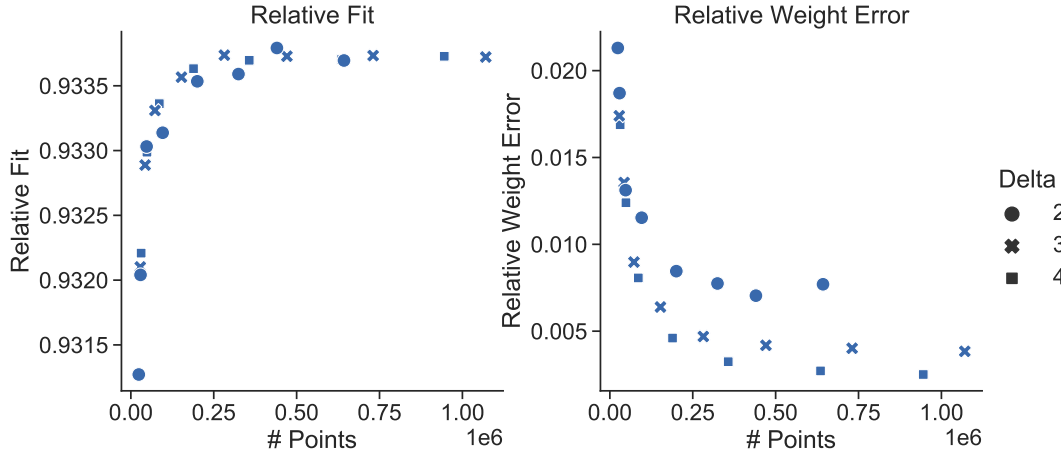


Figure 4.9: Combination of resolution coresets with interior removal: Dependence of $\Phi(G)$ and Ψ_G on the size of the resulting image support.

low relative cluster weight error. Increasing the size of the image support to 1,000,000 points results in only very marginal additional improvements. In our setting, an average of only 145 points per grain sufficed for obtaining nearly the same quality of fit as the 116,476 points in the original grain scan.

4.5.3 s-GBPD vs. DiM

While we showed in the previous paragraph that the indirect clustering-based methods yield already excellent accuracy for small sample sizes, we will now add a comparison with the direct benchmark method DiM. Specifically, we computed $\mathcal{A}, S, \Gamma$ via DiM and derived the clustering $\hat{C}$ from the diagram $\mathcal{P}(\mathcal{A}, S, \Gamma)$. To decrease the number of constraints we reduced the image support with the above techniques to 85,828 points, and used only distance constraints for neighboring grains in the scan.

In addition to the relative fit and the relative cluster weight error and, of course, the running time, we investigated various additional parameters. In particular, we computed the relative deviation of the centroids $c(G_i)$ and $c(\hat{C}_i)$, and also of the covariance matrices $\Sigma(G_i)$ and $\Sigma(\hat{C}_i)$. More precisely, the *relative centroid error* and the *relative covariance error* are defined as

$$\frac{1}{n} \sum_{i=1}^k \kappa_i \|c(G_i) - c(\hat{C}_i)\|_2 \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^k \kappa_i \|\Sigma(G_i) - \Sigma(\hat{C}_i)\|_2.$$

In the latter case, the spectral norm $\|\cdot\|_2$ of a matrix measures the largest singular value of the covariance differences and, thus, essentially captures the difference in the largest singular vectors of the covariance matrices.

As characteristics of the grain topology affect the physical properties of polycrystals, we also address the cell combinatorics. In particular, we computed the percentage of *correct neighborhoods* of the grains. More precisely, we count a voxel as adjacent to a given one if it is contained in what is called the 26-neighborhood in discrete topology, i.e. if its centroid is among the 27 points which can be reached by at most one move in each coordinate direction of the grid graph. A grain is a neighbor of another grain if a voxel of the former is adjacent to a voxel of the latter. The neighborhood is correct if all grain neighbors in the ground truth are also neighbors in the representation.

We also computed the percentage of grains whose neighborhood has at most 2 errors, where an error is a missing or additional grain neighbor not present in the ground truth. In view of the results of the previous section, we focus on the resolution coresets with interior removal as the specific version of s-GBPD, with image support of 85,828 points. Table 4.2 depicts these results.

Performance characteristics	s-GBPD	DiM
Relative fit Φ_G	0.9334	0.9559
Relative cluster weight error Ψ_G	0.01	0.02
Relative centroid error (in μm)	0.62	0.61
Relative covariance error (in μm^2)	15.53	8.72
Correct neighborhoods (in %)	35.70	63.45
Correct neighborhoods up to 1 error (in %)	72.08	91.71
Correct neighborhoods up to 2 errors (in %)	92.39	98.14
Computation time (in min)	2.60	627.00

Table 4.2: Comparison of s-GBPD and DiM for polycrystal representation.

While the Laguerre method from [Lyc+11] achieves a relative fit Φ_G of 0.86 on this data set (see [Lyc+11]), both our methods relying only on parts of the data improve the relative fit to a value of over 0.93. As expected, the direct model algorithm DiM performs better in most criteria but requires a significantly higher running time than s-GBPD. Since the differences are mostly marginal, the experimental findings indicate that we can compute a high precision 3D diagram representation of the grain scan in less than 3 minutes using s-GBPD. We expect that s-GBPD behaves similarly for other data sets as long as the resolution and the number of grains are of a similar magnitude.

4.6 Outlook

As we have seen, s-GBPD generates high-quality diagram representations of polycrystalline scans at speeds acceptable in many applications. If some applications require much faster algorithms, then – given the current state of the art – one needs to resort to algorithms that avoid optimization altogether by fixing the parameters to values that were found useful in empirical studies. Although some of these parameter choices can be interpreted in the context of Bayes classifiers [Alp+23a], currently none of them comes with approximation guarantees.

Let us close with some remarks on the role the grain scan plays in applying s-GBPD. While we focussed here on the situation that the whole grain scan (X, Y) is available, s-GBPD can be adjusted to require only certain measurements of the polycrystal. Grain volumes, centers, and second-order moments can, in many cases, be measured directly using tomographic techniques, and the described clustering approach can be solely based on such measurements. Both, pencil and resolution coresets, can still be constructed then. Grain scan-based interior removal, on the other hand, is (as emphasized correctly by its very name) based on the knowledge of (X, Y) .

It is, however, possible to remove the dependency on (X, Y) by adding an assumption that is common in *stochastic grain growth models*; see e.g. [Chi+13; Oka+00]. In fact, in such models, the underlying grain growth is often viewed as some realization of a stochastic process involving multivariate normal distributions $\mathcal{N}(\mu_i, \Sigma_i)$ for each grain. Their 3-dimensional mean vectors μ_i and 3×3 -covariance matrices Σ_i are determined by the measurements and correspond to $\mathcal{A}$ and S . Under such an assumption, interior points can be removed from the coreset by removing points with high density measured by the corresponding multivariate density function. As points close to the centroids have high density, this approach removes the interior points. Hence, it is possible to obtain additional speed-ups even when only the above measurements are available.

Acknowledgements

We thank Henning Friis Poulsen for fruitful discussions and valuable comments on an earlier version of the present text. We are also grateful to Henning Friis Poulsen and Allan Lyckegaard for providing the real-world data set used in this chapter. Also, we would like to express our great appreciation to the anonymous referees for their valuable comments on a previous version on the text.

Chapter 5

Dynamic Grain Models via Fast Heuristics for Diagram Representations

This text introduces a mathematical model for studying dynamic grain growth. In particular, we show how characteristic measurements, grain volumes, centroids, and central second-order moments at discrete moments in time can be turned quickly into a continuous description of the grain growth process in terms of geometric diagrams (which largely generalize the well-known Voronoi and Laguerre tessellations). We evaluate the computational behavior of our algorithm on real-world data.

5.1 Introduction

Grain growth is an important field of study in materials science as the resulting grain structures strongly influence the mechanical and physical properties of metals, ceramics, and other polycrystalline materials [Pou04; RRH04]. Although a number of different models have been introduced over the years, starting with works of Smith, von Neumann, and Mullins in the 1950s [Mul56; Neu52; Smi52], understanding and controlling grain growth remains challenging, both in theory and practice. For instance, the recent empirical study [Zha+18] demonstrates that neither the classical Hillert’s model [Hil65] nor the MacPherson-Srolovitz model [MS07] capture anisotropic grain growth adequately. Additional challenges are discussed in [HTS18; Hu+21; RZ18].

This text introduces a mathematical interpolation/extrapolation model for quickly turning information acquired at discrete moments in time into a continuous description of the grain growth process. It employs diagram representations of grain maps (which will be formally introduced in Section 5.2) based on measured estimates of the grain volumes, centroids, and, if available, central second-order moments. Such characteristics can be obtained, for instance, by utilizing tomographic techniques; see [Pou04; PV19].

The diagram representations considered in this chapter are generalizations of previously used (isotropic) Laguerre and Voronoi tessellations [Lyc+11; Tel+92; Tel+96] and allow for curved boundaries and non-convex grains. While these representations are based on a phenomenological model whose physical foundation is still not fully understood, they have consistently been reported to capture the physical principles governing the forming of polycrystals quite well; see [Alp+15; Alt+14; Pet+21; Šed+16; Šed+17; Spe+16; TGB15; TR18]. Several sophisticated optimization algorithms are available for computing or approximating such diagrams with high accuracy, [Alp+15; Pet+21], but they require relatively large computational cost. This can be a bottleneck, particularly in dynamic 3D studies. We therefore focus in this text on

extremely fast optimization-free heuristics for fitting diagrams to measured data. Methods for the related but generally different task of generating synthetic grain structures that follow a given distribution are discussed in [BBH16; Bou+20; Gui+11; KS08; XL09].

First, we use discriminant analysis to interpret a heuristic, subsequently referred to as H1, which was introduced in [Lyc+11] for the isotropic case and extended by [TR18] to the anisotropic case. Our analysis leads naturally to an even simpler and faster heuristic H2, proposed in [Alt+14], which can now be explained to perform equally well under certain conditions. Also, the comparison of H1 and H2 suggests that diagram parameters may provide additional insight into the stage of the growth process. We apply these heuristics to available grain maps in order to assess their quality. Then we introduce a dynamic grain growth model to obtain a continuous description of the process based on measurements at only a few moments in time. Finally, we evaluate the computational behavior of our algorithm on real-world data. Our study indicates that the growth process converges towards particularly simple diagrams, which seem to be energetically favorable.

The chapter is organized as follows. Section 5.2 introduces the relevant diagrams and formalizes the concept of grain scans. Sections 5.3 and 5.4 address optimization-free heuristics, both, in terms of a theoretical analysis and a practical evaluation on real-world data. Section 5.5 introduces a dynamic model which is capable of extending measurements at a few time steps to a continuous grain growth model very quickly. The results of our empirical study on practical grain data are then given in Section 5.5. Section 5.6 concludes with some final remarks.

5.2 Diagrams and grain scans

Adequate representations of grain maps can be valuable tools for understanding the physical principles of grain growth and the properties of the resulting material. In fact, the diagram representations studied in this text require only a few parameters per grain and are thus relatively easy to process. As they exhibit geometric and combinatorial features of grain maps which are not readily available otherwise, such representations may contribute to predicting and, potentially, even controlling the forming of new materials.

Some or all parameters which specify a diagram can be estimated from certain tomographic measurements, and optimizing the remaining ones then constitutes an inverse problem; see [Alp+23b]. Previous studies [Alp+15; Alt+14; Pet+21; Šed+16; Šed+17; Spe+16; TGB15; TR18] indicate that, in particular, grain volumes, centroids, and central second-order moments govern decisive properties of grain maps.

Let us now formally introduce the appropriate class of diagrams. Let $d \in \mathbb{N}$ denote the *dimension* of space. Here, we are mainly interested in the 3 dimensional case. However, our approach works for general d , and, in fact, planar sections or surfaces of polycrystals are also relevant in practice. Further, let $k \in \mathbb{N}$ denote the number of different grains in the underlying grain map. An *anisotropic power diagram* is specified by the following data for each $i \in [k] := \{1, \dots, k\}$:

- a positive definite symmetric matrix $A_i \in \mathbb{R}^{d \times d}$,
- a *site* $s_i \in \mathbb{R}^d$, and a
- *size* $\gamma_i \in \mathbb{R}$.

For simpler reference, we collect these parameters in the two families $\mathcal{A} = \{A_1, \dots, A_k\}$ and $\Gamma = \{\gamma_1, \dots, \gamma_k\}$, where repetitions of elements are permitted, and the set $S = \{s_1, \dots, s_k\}$. The sites specify the positions of the grains while the matrices in $\mathcal{A}$ describe characteristics of their shapes. In fact, each grain is equipped with its own *ellipsoidal norm* $\| \cdot \|_{A_i}$ given through A_i by

$$\|x\|_{A_i} = \sqrt{x^\top A_i x}, \quad (x \in \mathbb{R}^d).$$

The sizes γ_i are used to control the grain volume.

Given $\mathcal{A}$, S and Γ , the *anisotropic power diagram* (in the following often simply referred to as diagram)

$$\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma) = \{P_1, \dots, P_k\},$$

is a decomposition of $\mathbb{R}^d$ into *cells* P_i defined by

$$P_i = \{x \in \mathbb{R}^d : \|x - s_i\|_{A_i}^2 + \gamma_i \leq \|x - s_\ell\|_{A_\ell}^2 + \gamma_\ell, \quad \forall \ell \in [k] \setminus \{i\}\}.$$

In particular, two different cells do not share interior points, i.e., $\text{int}(P_i) \cap \text{int}(P_\ell) = \emptyset$ for $i \neq \ell$. See [BGK17] and [GK17] for examples and further properties of different classes of diagrams.

Diagrams can be viewed as continuous (or resolution-independent) representations of polycrystalline samples. Conventionally, the samples are represented as voxel-based digital images at some resolution, called *grain maps*. More generally, given a set of points $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^d$ in the sample and labels, i.e., $y_1, \dots, y_n \in [k]$, with $k \in \mathbb{N}$ denoting the number of different grains in the sample, we will call the labeled data set $\{(x_j, y_j) : j \in [n]\}$ a *grain scan* or *polycrystal scan*. In the following, we refer to it as (X, Y) with the understanding, that the indices of the labels in the family $Y = \{y_1, \dots, y_n\}$ correspond to the indices of the points in X . A grain scan therefore provides a partitioning of X into (the discretizations of) the *grains* $G_i = \{x_j : (x_j, i) \in (X, Y)\}$. We can then measure the fit of the diagram in terms of the symmetric difference of G_i and $X \cap P_i$ for all $i \in [k]$. (Points on the boundary of two or more cells are assigned according to some preference criterion or, conservatively, can be viewed as classification error, but are essentially irrelevant in our context and hence generally discarded.)

Usual grain maps are special grain scans where the sample is assumed to be in the range $[0, 1]^d$ and points of X are identified with the points of $[0, 1]^d$ in the grid $\rho\mathbb{Z}^d$ for some *resolution* $\rho \in (0, 1)$. Note that, with $\kappa_i = |G_i|$, the term $\rho^d \kappa_i$ can be regarded as an approximation of the volume ν_i of the i th grain.

If a grain scan is available the computation of a best fitting diagram requires the optimization of all three characteristics $\mathcal{A}$, S and Γ . While this can indeed be done, see [Alp+23b], it is, however, too time-consuming, given current optimization technology, for dynamic 3D applications. Hence, we will in the following assume that $\mathcal{A}$ and S are available through measurements and Γ will be determined by a heuristic which may, additionally, use grain volume information, i.e., (approximations of) $\nu_1, \dots, \nu_k$. As such heuristics abstain from any optimization, they are computationally fast but potentially result in a higher misclassification error.

5.3 Optimization-free heuristics

This section considers two heuristics that derive all characteristic parameters $\mathcal{A}, S, \Gamma$ for an anisotropic power diagram directly from given measurements. These heuristics avoid

optimization routines altogether, are fast and, as we will see, already provide a reasonable classification accuracy. More specifically, we provide a new interpretation of the heuristics introduced in [Lyc+11; TR18] and [Alt+14] within the framework of discriminant analysis.

Choices for $\mathcal{A}$ and S

Discriminant analysis is a popular classification technique in statistics and provides a different perspective on the a priori choices for the characteristic parameters $(\mathcal{A}, S, \Gamma)$; see e.g. [HTF09]. Within its realm, the grain scan is regarded as a sample of the multivariate random variable $(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X}$ specifies a point in $[0, 1]^d$ and $\mathbf{Y}$ gives the index $i \in [k]$ of the corresponding grain. Discriminant analysis estimates the distribution of the multivariate random variable from samples and assigns a point to the class with the highest probability p . Hence, the decision boundary between two classes i, i' consists of those points x that have an equal conditional probability of lying in either class, i.e.,

$$\{x \in \mathbb{R}^d : p(\mathbf{Y} = i | \mathbf{X} = x) = p(\mathbf{Y} = i' | \mathbf{X} = x)\}.$$

For its estimation, discriminant analysis applies the maximum likelihood principle under the assumption that the members from each class are normally distributed. As each normal distribution $\mathcal{N}(\mu_i, \Sigma_i)$ is fully determined by its d -dimensional mean vector μ_i and its $(d \times d)$ -covariance matrix Σ_i , it is these characteristics that need to be estimated. It is well known that the maximum likelihood principle results in the choices of the sites S as the centroids and the matrices $\mathcal{A}$ as the inverse of the covariance matrices of the sample, i.e., more precisely,

$$\begin{aligned} \mu_i = c_i &= \frac{1}{|G_i|} \sum_{x_j \in G_i} x_j, \\ A_i &= (\Sigma(G_i))^{-1} = \left(\frac{1}{|G_i|} (x_j - c_i : x_j \in G_i)(x_j - c_i : x_j \in G_i)^T \right)^{-1} \end{aligned}$$

for $i \in [k]$. Note that the latter requires that the matrices $\Sigma(G_i)$ have full rank d . This can be assumed here as, in the underlying model, grains that are degenerate to volume 0 do not matter. The above maximum likelihood setting is precisely the choice for $\mathcal{A}$ and S suggested in [Alp+15], and subsequently also used in [TR18]. Also, estimates for these parameters are available through tomographic measurement.

Let us point out that, in practice, discriminant analysis is often applied even when the underlying assumption of a normal distribution is only met approximately (as is the case here). This is justified as the procedure is generally reported to be quite robust; see, e.g., [LG79].

Choices for Γ

We will now turn to the third set of characteristics, i.e., the sizes Γ . We show first that the choice of Γ in heuristic H1 can also be interpreted within the paradigm of discriminant analysis with a specific prior. Building on this interpretation, a different natural choice will lead to H2.

To begin with, let us briefly recall the volume argument of [Lyc+11] and (in its anisotropic generality) [TR18] that leads to the specific choice for Γ in heuristic H1.

Let $A \in \mathbb{R}^{d \times d}$ be positive definite and symmetric. As A can be diagonalized there exist an orthogonal matrix $U \in \mathbb{R}^{d \times d}$ and a diagonal matrix $D = \text{diag}(\lambda_1, \dots, \lambda_d) \in \mathbb{R}^{d \times d}$ with

$\lambda_1, \dots, \lambda_d > 0$ such that $A = UDU^T$. Then, using the abbreviation $D^{1/2} = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_d})$, we have for the unit ball $\mathbb{B}_A^d$ and $\rho \in (0, \infty)$

$$\begin{aligned}\rho\mathbb{B}_A^d &= \{x \in \mathbb{R}^d : \|x\|_A \leq \rho\} = \{x \in \mathbb{R}^d : x^T A x \leq \rho^2\} \\ &= \{x \in \mathbb{R}^d : x^T U D U^T x \leq \rho^2\} = \rho U D^{-1/2} \mathbb{B}_2^d.\end{aligned}$$

Hence, the volume $\text{vol}(\rho\mathbb{B}_A^d)$ of this ellipsoid is given by

$$\text{vol}(\rho\mathbb{B}_A^d) = \rho^d \text{vol}(U D^{-1/2} \mathbb{B}_2^d) = \rho^d \det(A)^{-1/2} \text{vol}(\mathbb{B}_2^d).$$

Regarding the ellipsoid $c_i + \rho\mathbb{B}_{A_i}^d$ as an approximation of the i th grain, it is now natural to choose γ_i according to the volume condition

$$\text{vol}(\{x \in \mathbb{R}^d : \|x\|_{A_i}^2 + \gamma_i \leq 0\}) = \nu_i$$

involving the available approximation ν_i for the volume of the i th grain. A simple computation yields

$$\gamma_i = -\rho^2 = -\left(\frac{\nu_i}{\text{vol}(\mathbb{B}_{A_i}^d)}\right)^{2/d} = -\left(\frac{\nu_i \sqrt{\det(A_i)}}{\text{vol}(\mathbb{B}_2^d)}\right)^{2/d}$$

For $d = 3$, and with the specific setting of $A_i = (\Sigma(G_i))^{-1}$, this yields the choice

$$\gamma_i = -\left(\frac{3\nu_i}{4\pi\sqrt{\det(\Sigma(G_i))}}\right)^{2/3}$$

from [Lyc+11; TR18]. In order to interpret this choice via discriminant analysis, we determine which *probability distribution*, called *prior*, will lead to the same decision boundaries between the classes. So, let

$$\pi_1, \dots, \pi_k \in [0, \infty), \quad N = \sum_{i=1}^k \pi_i \neq 0.$$

Then $(\pi_1, \dots, \pi_k)$ specifies the a priori “belief” about the *prior probability* π_i/N that a point (regardless of its position) belongs to cluster i , i.e.,

$$\frac{\pi_i}{N} = p(\mathbf{Y} = i), \quad (i \in [k]).$$

The computation of the decision boundary is then facilitated via *Bayes’ Theorem*. As each class is assumed to be normally distributed, Bayes’ Theorem involves the conditional density functions

$$f_{\mathbf{X}|\mathbf{Y}=i}(x) = \frac{\det(A_i)^{1/2}}{(2\pi)^{d/2}} \cdot \exp\left(-\frac{1}{2}(x - c_i)^\top A_i(x - c_i)\right)$$

of the k estimated normal distributions $\mathcal{N}(\mu_i, \Sigma_i)$, i.e., with

$$\mu_i = c_i, \quad \Sigma_i = \Sigma(G_i) = A_i^{-1}.$$

In this situation, Bayes' Theorem states that

$$p(\mathbf{Y} = i | \mathbf{X} = x) = \frac{f_{\mathbf{X} | \mathbf{Y}=i}(x) \pi_i}{\sum_{\ell=1}^k f_{\mathbf{X} | \mathbf{Y}=\ell}(x) \pi_\ell}.$$

Therefore, the maximum likelihood decision boundaries between two classes i, i' can be determined by solving the equation

$$f_{\mathbf{X} | \mathbf{Y}=i}(x) \cdot \pi_i = f_{\mathbf{X} | \mathbf{Y}=i'}(x) \cdot \pi_{i'}.$$

Taking the logarithm, using that $(x - c_i)^\top A_i (x - c_i) = \|x - c_i\|_{A_i}^2$, and simplifying, this condition turns into the equation

$$\|x - c_i\|_{A_i}^2 - \log(\det(A_i)) - 2 \log(\pi_i) = \|x - c_{i'}\|_{A_{i'}}^2 - \log(\det(A_{i'})) - 2 \log(\pi_{i'}).$$

Hence, these decision boundaries coincide with that of the diagram with the parameters $(\mathcal{A}, S, \Gamma)$ if and only if

$$\gamma_i = \gamma_i^{(1)} = -\log(\det(A_i)) - 2 \log(\pi_i) + \gamma, \quad (i \in [k]),$$

where $\gamma \in \mathbb{R}$ is an arbitrary constant. (Recall that the diagrams are invariant under such common additions to the sizes.) For the above choice of the sizes in H1, we hence obtain

$$\pi_i = \pi_i^{(1)} = \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}} \cdot \exp\left(\frac{1}{2} \left(\frac{\nu_i \sqrt{\det(A_i)}}{\text{vol}(\mathbb{B}_2^d)}\right)^{2/d}\right), \quad (i \in [k]).$$

Since

$$\frac{1}{\sqrt{\det(A_i)}} = \sqrt{\det(\Sigma(G_i))} = \frac{1}{\text{vol}(\mathbb{B}_2^d)} \cdot \text{vol}(\mathbb{B}_{A_i}^d),$$

the first factor is proportional to the volume of the ellipsoid $\mathbb{B}_{A_i}^d$. The second factor is a “correction term” which increases exponentially if the measured grain volume $\rho^d \kappa_i$ exceeds $\text{vol}(\mathbb{B}_{A_i}^d)$. The first term seems natural while the second is quite unusual.

Within the realm of discriminant analysis it appears reasonable to simply set $\gamma_i = 0$ for all clusters i . Then we obtain the priors

$$\pi_i = \pi_i^{(2)} = \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}}, \quad (i \in [k]),$$

which are proportional to the volumes of the ellipsoids corresponding to the covariance matrices of the grains. Thus, this simpler prior distribution encodes the assumption that the ellipsoid volumes already represent the grain volumes well. This justifies the choice

$$\gamma_1^{(2)} = \dots = \gamma_k^{(2)} = 0,$$

proposed in [Alt+14]. We will refer to the respective heuristic as H2.

In the next section we compare the behavior of H1 and H2 on available real-world grain scans.

5.4 Comparison of H1 and H2

We compare the two heuristics H1 and H2 on a real-world 3D grain scan and indicate how their difference can be interpreted. This may be of particular interest for understanding the state of a grain growth process and will be taken up in Section 5.5.

Experimental evaluation

We compare the methods with a data set taken from [Lyc+11] that has been obtained by a synchrotron micro-tomograph experiment conducted on a metastable beta titanium alloy (Ti β 21S). The material was scanned with the volume of $240\mu\text{m} \times 240\mu\text{m} \times 420\mu\text{m}$ at a resolution corresponding to a voxel size of $0.7\mu\text{m}$. This scan resulted in a 3D voxel image of size $339 \times 339 \times 599$ composed of a total of 591 grains. All our computations are carried out in 3D, i.e., $d = 3$.

In the following, we assess the quality of fit by means of different measures. Let $\mathcal{P} = \{P_1, \dots, P_k\}$ be a diagram (obtained by H1, H2, or, for that matter, any other algorithm). We set

$$\hat{C}_i = \{x_j \in X : x_j \in \text{int}(P_i)\} \quad (i \in [k]), \quad \hat{C} = (\hat{C}_1, \dots, \hat{C}_k).$$

Note that $\hat{C}$ is not a clustering of X but only of $X \cap \bigcup_{i=1}^k \text{int}(P_i)$; this is indicated by writing $\hat{C}$ rather than C .

A natural measure how well $\mathcal{P}$ captures the grain scan $G = (G_1, \dots, G_k)$ is the *relative fit* or *accuracy* Φ_G of $\hat{C}$ defined by

$$\Phi_G(\hat{C}) = \frac{1}{n} \left| \bigcup_{i=1}^k (G_i \cap \hat{C}_i) \right|.$$

Let us point out that, as they are not assigned within $\hat{C}$, all points on the cell boundaries count as misclassified.

In Table 5.1 we will also report on the behavior with respect to the *relative cluster weight error*

$$\Psi_G(\hat{C}) = \frac{1}{n} \sum_{i=1}^k |\kappa_i - |\hat{C}_i||,$$

and the relative deviation of the centroids $c(G_i)$ and $c(\hat{C}_i)$, and the covariance matrices $A_i^{-1} = \Sigma_i = \text{Cov}(G_i)$ and $\text{Cov}(\hat{C}_i)$, respectively. Formally, the *relative centroid error* and the *relative covariance error* are defined as

$$\frac{1}{n} \sum_{i=1}^k \kappa_i \|c(G_i) - c(\hat{C}_i)\|_2 \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^k \kappa_i \|\Sigma_i - \text{Cov}(\hat{C}_i)\|_2.$$

For the latter, we used the spectral norm. This matrix norm measures the largest singular value of the covariance differences and, thus, intuitively captures the difference in the largest singular vectors of the covariance matrices. Note that, as defined, the latter two measures

depend on the original physical dimensions of the sample. We refrain from any additional normalization as we do not conduct inter-sample studies here.

We will also address combinatorial features. More precisely, we consider the percentage of grains with a correct neighborhood, i.e., with all grain neighbors in the ground truth being also neighbors in the representation. Here, a grain is a neighbor of another grain if a voxel of the former grain is adjacent (face-connected, edge-connected or vertex-connected) to a voxel of the later grain. We also provide this percentage when one and, respectively, two errors in the neighborhood are allowed, i.e., when there is either one additional or one missing voxel.

The results of our evaluation are shown in Table 5.1.

Performance Characteristics	H1	H2
Accuracy (in %)	92.98652	92.98658
Relative Cluster Weight Error	0.02869	0.02869
Relative Centroid Error (in μm)	0.64346	0.64347
Relative Covariance Error (in μm^2)	15.25372	15.25376
Correct Neighborhoods (in %)	42.30118	42.30118
Correct Neighborhoods up to 1 Error (in %)	80.54146	80.54146
Correct Neighborhoods up to 2 Errors (in %)	94.92386	94.92386
Running Time (in s)	1.49641	0.00000

Table 5.1: Comparison of H1 and H2.

Let us point out that the displayed running times involve the heuristics but, for a fairer comparison, do not include the computation of the performance criteria. Hence, the entry 0 for the running time of H2 refers to the fact that all parameters defining $\mathcal{P}$ are given upfront. Also, while we exercised appropriate care for the implementations, we do not claim that our code is fully optimized.

In terms of fit, we observe that both heuristics perform extremely well, in fact, almost identical, on the given real-world data set. We will now offer an explanation for the latter behavior.

Similarity and difference

While both heuristics, H1 and H2, employ the same matrices $\mathcal{A}$ and sites S , they generally differ in the sizes Γ of the resulting diagrams. Hence, the nearly identical performance depicted in Table 5.1 for all applied quality criteria may come as a surprise. As observed before, the priors underlying H1 and H2 denoted $\pi_i^{(1)}$ and $\pi_i^{(2)}$, as before, differ by a factor that depends exponentially on the deviation of $\text{vol}(\mathbb{B}_{A_i}^d)$ from the measured grain volume ν_i .

Now, let α_i be a positive real number such that

$$\nu_i = \alpha_i \cdot \text{vol}(\mathbb{B}_{A_i}^d) = \alpha_i \cdot \frac{\text{vol}(\mathbb{B}_2^d)}{\sqrt{\det(A_i)}}.$$

Then, for each $i \in [k]$,

$$\begin{aligned}\pi_i^{(1)} &= \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}} \cdot \exp\left(\frac{1}{2} \left(\frac{\nu_i \sqrt{\det(A_i)}}{\text{vol}(\mathbb{B}_2^d)}\right)^{2/d}\right) = \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}} \cdot \exp\left(\frac{1}{2} \alpha_i^{2/d}\right) \\ &= \pi_i^{(2)} \cdot \exp\left(\frac{1}{2} \alpha_i^{2/d}\right),\end{aligned}$$

and, expressed in terms of the sizes $\gamma_i^{(1)}$ used in H1, we obtain

$$\gamma_i^{(1)} = - \left(\frac{\nu_i}{\text{vol}(\mathbb{B}_{A_i}^d)} \right)^{2/d} = -\alpha_i^{2/d}.$$

Now, suppose that all factors α_i are identical, i.e., $\nu_i = \alpha \cdot \text{vol}(\mathbb{B}_{A_i}^d)$ for some positive real number α independent of i . Then, of course,

$$\pi_i^{(1)} = \pi_i^{(2)} \cdot \exp\left(\frac{1}{2} \alpha^{2/d}\right), \quad \gamma_i^{(1)} = -\alpha^{2/d}, \quad (i \in [k]).$$

Hence, the (normalized) priors for H1 and H2 coincide, which implies that the maximum likelihood decision boundaries for the two heuristics are the same. Further, since diagrams are invariant under the addition of a constant to each size, the diagrams produced by H1 and H2 are identical.

We suspect that grain structures for which the volumes of the unit balls $\mathbb{B}_{A_i}^d$ with respect to $A_i = (\Sigma(G_i))^{-1}$ are nearly proportional to the grain volumes ν_i might turn out to be energetically favorable. According to the above reasoning, the former may be quantified by the variance of the powers $\Gamma^{d/2} = (\gamma_1^{d/2}, \dots, \gamma_k^{d/2})$ computed in H1, i.e.,

$$\text{var}(\Gamma^{d/2}) = E(\Gamma^{d/2} - E(\Gamma^{d/2})) = \frac{1}{k} \sum_{i=1}^k \left(\gamma_i^{d/2} - \frac{1}{k} \sum_{i=1}^k \gamma_i^{d/2} \right)^2.$$

As an example, we will in Section 5.5 compute the evolution of $\text{var}(\Gamma^{3/2})$ at different moments in time for a real-world 3D grain growth process. In terms of H1 and H2, we might expect that if the two heuristics perform differently in practice, the current grain structure will not be energetically minimal, and further growth will result in an improved structure.

We close this section with a “constructed” theoretical example which sheds some additional light on the behavior of the two heuristics. Since the general principle of the construction is the same in higher dimension and can, hence, already be illustrated in 2D we restrict the following description to the case $d = 2$.

As depicted in the top row of Fig. 5.1, the image consists of two grains G_1 and G_2 ; one (orange) is the square $[0, \beta] \times [0, \beta]$ for different values of $\beta \in (0, 1)$, the other (blue) is its complement in $[0, 1] \times [0, 1]$. Of course, one would not expect such samples to originate from real-world grain scans. The bottom row of Fig. 5.1 shows the corresponding ellipses $c_i + \mathbb{B}_{A_i}^2$ with respect to $A_i = (\Sigma(G_i))^{-1}$.



Figure 5.1: A family of grain scans involving a parameter β to illustrate differences between H1 and H2. Top, from left to right: Grain scans for $\beta = 0.1$, $\beta = 0.5$, and $\beta = 0.9$. Grain G_1 (orange) fills the square $[0, \beta] \times [0, \beta]$, G_2 (blue) fills the rest of the unit square. The bottom row shows the computed ellipses $c_i + \mathbb{B}_{A_i}^2$ with $A_i = (\Sigma(G_i))^{-1}$ for $i = 1, 2$.

As argued before, the variance of the ratios $\nu_i / \text{vol}(\mathbb{B}_{A_i}^2)$ is a relevant parameter. Figure 5.2 therefore provides plots of

$$\frac{\beta^2}{\text{vol}(\mathbb{B}_{A_1}^2)} \quad \text{and} \quad \frac{1 - \beta^2}{\text{vol}(\mathbb{B}_{A_2}^2)}$$

as a function of β .

Note that these ratios differ considerably for larger values of β . Hence, we would expect that the heuristics H1 and H2 behave quite differently in these cases. This is confirmed by the resulting diagrams depicted in Fig. 5.3.

The behavior for $\beta = 0.9$ suggests that H1 tries to preserve the areas of the grains, while H2 seems to favor their shapes. If the relevant volume ratios are close, as is the case for small β , both heuristics achieve both aims and perform similarly.

Figure 5.4 shows the ratios of the grain volume ν_i and the areas of the diagram cells under the two models H1 and H2, respectively.

5.5 Dynamic grain growth

We now describe a simple interpolation/extrapolation model that is capable of quickly turning information acquired at a few moments in time into a continuous description of the grain

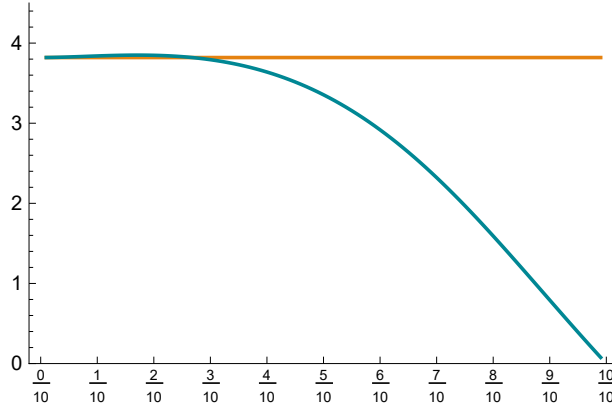


Figure 5.2: Ratios $\nu_i / \text{vol}(\mathbb{B}_{A_i}^2)$ of the areas of the grain and the corresponding ellipsoid for $i = 1$ (orange) and $i = 2$ (blue) as a function of β .

growth process. Then we report on the results of an empirical evaluation on a real-world grain scan time series. Finally, we study the correlation between the diagram size parameters Γ and the disappearance of grains in the process.

The model

Since well-chosen diagrams have been demonstrated to yield high-quality representations of grain structures of many types of polycrystalline materials, it is natural to exploit their rather low-dimensional parameter space for deriving continuous models from data measured at discrete times. Recall that a diagram is fully characterized by the triple $(\mathcal{A}, S, \Gamma)$. In order to simulate the dynamic evolution of grain structures over time, we can fit a polynomial (or any other desired type of function) for each diagram parameter extracted from a discrete-time series polycrystal data set. These functions then represent the continuous change in diagram parameters during the growth process. Understanding the evolution of the diagram parameters over time may help predict changes in the material during an annealing process and may provide further insights into the material characteristics.

Suppose, that 3D-grain scans (or at least appropriate characteristic measurements) are available at the discrete times $t_0, t_1, \dots, t_m$. Then we compute, for each $t = t_\ell$, the parameters of an anisotropic power diagram $\mathcal{P}_t = \mathcal{P}(\mathcal{A}_t, S_t, \Gamma_t)$ which represent the grain scans. Subsequently, we select a class $\mathcal{F}$ of parameterized functions and determine one that best fits the data $\mathcal{A}_t, S_t, \Gamma_t$. In our experimental setting we choose polynomials of a small degree to avoid overfitting, and we measure the fit in terms of least-squares deviation.

In 3D, the interpolation/extrapolation involves ten functions of $\mathcal{F}$ for each grain, three for the coordinates of the diagram sites, three for the rotation part of the grains' covariance matrix, three for the length of the principal axis of the covariance matrix, and one for the size parameter. Of course, in order to set up the computation of the functions, we need to “track” each grain, i.e., we need to identify which grain corresponds to which at the different moments t in time.



Figure 5.3: Diagrams for the family of grain scans from Fig. 5.1 obtained via H1 (top row) and H2 (bottom row).

During the grain growth process, grains may disappear. We assume, however, that no new grains emerge. In the experimental series that will be described in more detail below, about 40% of all initially present grains disappeared throughout the process. Hence we distinguish between such *non persistent* grains and *persistent* grains that exist throughout the observed time interval. Of, course, for non persistent grains, the functions are fitted only as long as the grain “definitely exists”. More specifically, if a previously existing grain disappears from the measurement, we choose as last diagram parameters the covariance matrix from the previous time step and compute a size parameter such that its grain volume is essentially zero. If, on the other hand, for some moment in time our prediction generates a covariance matrix and a size parameter which result in a grain volume close to zero, we exclude the grain from further consideration.

The diagrams $\mathcal{P}_t$ can, in principle, be computed by any suitable method. In addition to the obvious criteria that any employed algorithm should be computationally fast and produce diagrams of appropriate accuracy, one has to pay attention to its behavior concerning the invariances within the parameter space to avoid corruption of the interpolation/extrapolation. As we are using the heuristic H1 described in Section 5.3, all parameters are, however, uniquely determined by the measurements, and we can hence, in the following, ignore this issue.

The data

The data used for our experimental study is taken from [Zha+18]. It comprises 15 3D scans of a 99.9% pure polycrystalline iron sample taken at different moments in time of an annealing process. More precisely, the material was first cold-rolled and annealed for 30 minutes at 700°C to fully recrystallize before it was first scanned (time t_0). Subsequently, the sample was annealed again at 800°C for 10 minutes to simulate grain growth, cooled to stabilize, and

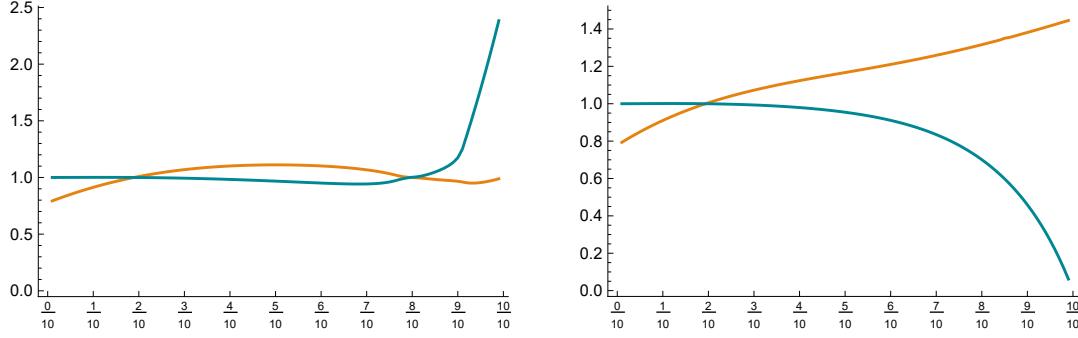


Figure 5.4: Ratio of the area ν_i of the original grain and the area of the corresponding diagram cell obtained by H1 (left) and H2 (right) for $i = 1$ (orange) and $i = 2$ (blue) as a function of β .

then scanned again (time t_1). This was repeated another 13 times with annealing periods of 5 minutes (times $t_2, \dots, t_{14}$). The effect of the process is already visible by the number of grains in the sample. While the first scan exhibits 1327 grains, the last one contains only 776, and the number of interior grains strictly decreases from 387 to 189; see [Zha+18, Table 1] for more details. The identification of grains over the 15 moments in time is already provided by [Zha+18].

For this data set, we computed 15 anisotropic power diagrams. We used H1 as we were also interested in featuring the temporal development of the corresponding sizes in view of the relation of H1 and H2 elaborated in Section 5.3.

Evaluation

We apply a cross-validation to assess the quality of the model. To compute the accuracy at time t_ℓ , we exclude the data for t_ℓ and fit the functions on the data for the remaining times $\{t_0, \dots, t_{14}\} \setminus \{t_\ell\}$. Then we evaluate the obtained interpolation/extrapolation functions at time t_ℓ to compute diagram parameters $\hat{\mathcal{A}}_\ell, \hat{S}_\ell, \hat{\Gamma}_\ell$. Finally, we determine the accuracy Φ of the resulting diagram $\hat{\mathcal{P}}_\ell$ of representing the grain scan at t_ℓ .

The results for polynomials of degree 1, 2, and 3 are depicted in Fig. 5.5 in yellow, orange, and red, respectively. Let us point out that the piecewise linear curves which connect the 15 data points for each color are only drawn for a better perception of the trends over time.

We observe that the degree of the employed polynomials does not seem to have a strong impact on the fit. It would be interesting to link this observation to the “uniform setup” in terms of the actions over time of the underlying grain growth process.

As an additional reference the blue markers give an upper “orientation” for the accuracy that one may expect from the model. In fact, the blue markers depict the accuracy of H1 at each of the 15 time steps. The curves are dashed at both ends to signify that the model changes from interpolation to extrapolation at these extremes. Also, note that the earlier annealing times differ from the latter ones.

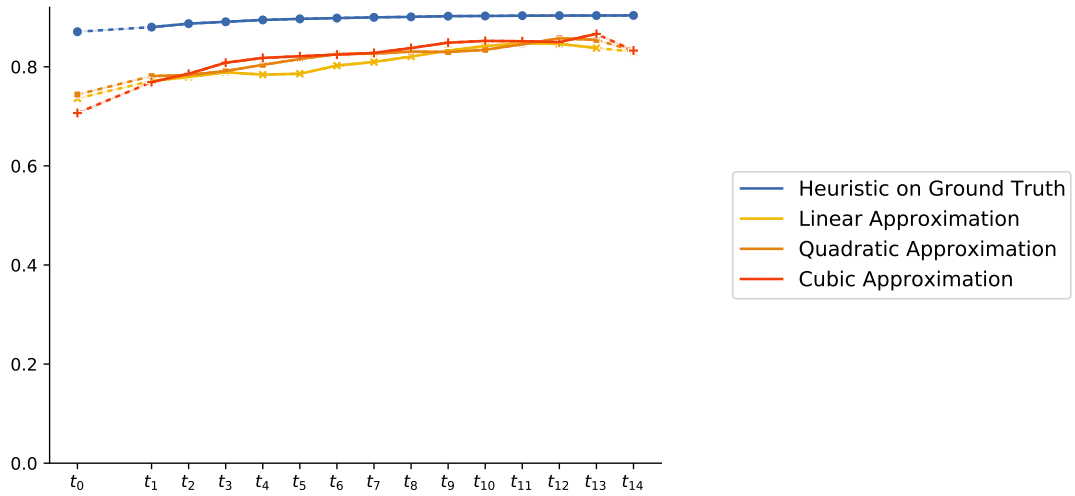


Figure 5.5: Accuracy (within the cross-validation paradigm) of the interpolation/extrapolation model plotted for each time step using polynomials of degree 1, 2, and 3, respectively for the fit.

Figure 5.5 shows that the accuracy of our models is generally high. With quadratic polynomials, we obtain an initial accuracy of 74%, which increases over time to approximately 86% for later time steps. Note that this is not much below the accuracy obtained by H1 on the ground truth. The general trend that accuracy increases with time (and hence prediction becomes easier) for this data set is in perfect agreement of the findings in [Zha+18; ZGZ20], which show that over time the system approaches the self-similar regime.

Figure 5.6 show a fixed 2D slice of the 3D polycrystal at various moments in time and the diagrams resulting from our model (in the cross-validation scheme described above).

Size variation

In the remaining part of this section, we study how the growth process affects the size parameters Γ_t . As Fig. 5.5 shows, the accuracy of the model increases with the duration of the process. This may be seen as an indication that the employed diagram structures are energetically favorable. Complementing the analysis of Section 5.4, we will now empirically study how the size parameters Γ of the diagrams change as the grains grow.

Let us begin by recalling the interpretation for the similarities and differences of H1 and H2. In fact, based on the analysis of Section 5.4 we suspected that for persistent grains the variances of the powers $\Gamma_t^{3/2}$ of the sizes chosen by H1 should be significantly smaller than for non persistent grains. As Tables 5.2 and 5.3 show, these variances are indeed all small but significantly smaller for the persistent grains. This means, on the one hand, that the sites do not vary much, and we might as well have employed the simpler heuristic H2. On the other hand, in terms of the sites, non persistent grains are less “regular” than persistent grains.

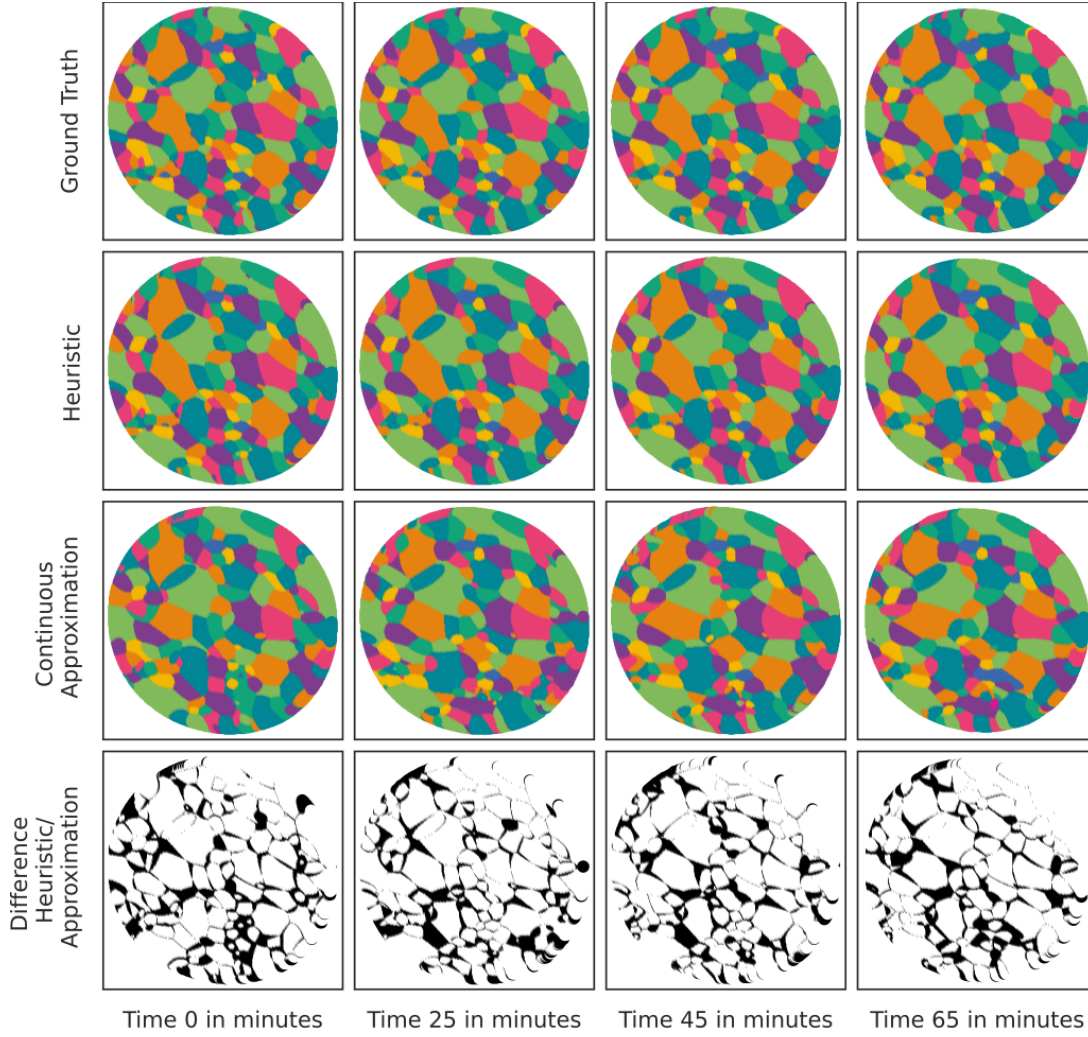


Figure 5.6: Visual comparison of diagrams obtained from full data (available for $t_0, \dots, t_{14}$) and the interpolation/extrapolation model that uses all data except for the respective moment in time. A fixed 2D slice through the 3D data set at times t_0, t_5, t_9 and t_{13} (top row); the corresponding diagram representation obtained via H1 (2nd row) and, respectively, via the interpolation/extrapolation model that uses only data from the other moments in time and best-fitting quadratic polynomials (3rd row); symmetric difference between the images from the 2nd and 3rd row (bottom row).

	$t_{\ell+0}$	$t_{\ell+1}$	$t_{\ell+2}$	$t_{\ell+3}$	$t_{\ell+4}$
$\ell = 0$	$6.409 \cdot 10^{-8}$	$6.214 \cdot 10^{-8}$	$5.130 \cdot 10^{-8}$	$3.016 \cdot 10^{-6}$	$2.544 \cdot 10^{-7}$
$\ell = 5$	$6.719 \cdot 10^{-6}$	$5.839 \cdot 10^{-6}$	$1.057 \cdot 10^{-5}$	$7.025 \cdot 10^{-4}$	$9.815 \cdot 10^{-7}$
$\ell = 10$	$2.286 \cdot 10^{-4}$	$7.425 \cdot 10^{-5}$	$5.749 \cdot 10^{-5}$	$1.185 \cdot 10^{-3}$	$1.823 \cdot 10^{-5}$

Table 5.2: Variance of $\Gamma_{t_\ell}^{3/2}$ for $\ell \in \{0, 1, \dots, 14\}$ of persistent grains.

	$t_{\ell+0}$	$t_{\ell+1}$	$t_{\ell+2}$	$t_{\ell+3}$	$t_{\ell+4}$
$\ell = 0$	$7.570 \cdot 10^{-3}$	$1.040 \cdot 10^{-2}$	$4.915 \cdot 10^{-3}$	$1.974 \cdot 10^{-2}$	$2.993 \cdot 10^{-3}$
$\ell = 5$	$1.278 \cdot 10^{-2}$	$1.834 \cdot 10^{-2}$	$2.790 \cdot 10^{-2}$	$4.158 \cdot 10^{-3}$	$8.959 \cdot 10^{-3}$
$\ell = 10$	$2.608 \cdot 10^{-2}$	$2.804 \cdot 10^{-2}$	$2.955 \cdot 10^{-2}$	$3.541 \cdot 10^{-3}$	—

Table 5.3: Variance of $\Gamma_t^{3/2}$ for $\ell \in \{0, 1, \dots, 14\}$ of non persistent grains.

A closer look at Table 5.2 shows that, somewhat unexpectedly, not even for persistent grains the variance of $\Gamma_t^{3/2}$ decreases monotonously in time. One may have anticipated that with decreasing “system energy” the behavior of the volumes of the unit balls $\mathbb{B}_{A_i}^3(t)$ with respect to $A_i(t) = (\Sigma(G_i(t)))^{-1}$ become increasingly similar to that of the grain volumes $\nu_i(t)$. The results in Table 5.2, however, seem to suggest that, in the presence of grain disappearance, the neighboring grains fill the newly available space more directly and it requires additional energy to reach a more regular mass distribution.

Grain disappearance

Next, we use a standard logistic regression model for predicting the probability $p_i(t)$ of the binary response variable which signifies whether grain i vanishes (response value 1) at the next observed moment $t + 1$ or not (response value 0). As predictor variables we use the grain’s volume and its size parameter as computed by H1. Hence, we estimate three parameters, factors β_1 and β_2 for each of the two variables and one parameter β_0 for the intercept, i.e., we use the regression model

$$p_i(t) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \nu_i(t) + \beta_2 \gamma_i(t))}}.$$

Note that the parameters β_0, β_1 and β_2 are independent of t . Hence the model is identical for all time steps in the sense that it depends only on the grain characteristic. Our *null hypothesis* is that the size and volume parameter do not correlate with grain disappearance. As a significance level we choose 0.05.

Table 5.4 depicts the computed coefficients and p -values. Let us, in addition, point out that the McFadden’s Pseudo R^2 value is 0.322, indicating an excellent model fit.

	coefficient	p -value
β_0 (intercept)	-4.0537	<0.0001
β_1 (volume)	-0.0010	<0.0001
β_2 (size parameter)	-1.0924	<0.0001

Table 5.4: Coefficients estimated by maximum likelihood estimation and their p -values for the described logistic regression model.

As it turns out, we can reject the null hypothesis at a significance level of 0.05. This suggests that the diagram size parameter Γ_t is statistically significant in this model for predicting grain disappearance.

5.6 Final remarks

As we have seen, measurements at discrete times of a grain growth process can be turned into a continuous description of the process using diagram representations. Our findings about the increasing fit of anisotropic diagrams, the variance of $\Gamma_t^{3/2}$ and the relevance of the sizes for predicting grain disappearance indicate that, in the course of the growth process, the grain scans tend towards particularly “simple” diagrams with essentially uniform size parameters.

Since all computations are fast, and can, if necessary, generally even be speeded up by using H2 rather than H1, we believe that the described model has the potential of becoming a practical tool for simulation and further studies of grain growth in practice. A next step is to evaluate the model more extensively on various types of grain scan data. It would be desirable if a consolidating foundation of the model can be achieved based on first principles.

Finally, let us point out that the model allows for some flexibility to incorporate additional insight. It is possible, for instance, to incorporate available characteristics of grain growth into the curve fitting process.

Acknowledgements

We thank Henning Friis Poulsen for fruitful discussions and valuable comments on an earlier version of the present text. We are also grateful to Henning Friis Poulsen and Allan Lyckegaard for providing the real-world data set used in this text.

Chapter 6

Power-SLIC: Fast Superpixel Segmentations by Diagrams

Superpixel algorithms grouping pixels with similar color and other low-level properties are increasingly used for pre-processing in image segmentation. In recent years, a focus has been placed on developing geometric superpixel methods that facilitate the extraction and analysis of geometric image features. Diagram-based superpixel methods are important among the geometric methods as they generate compact and sparsely representable superpixels. Introducing generalized balanced power diagrams to the field of superpixels, we propose a diagram method called Power-SLIC. Power-SLIC is the first geometric superpixel method to generate piecewise quadratic boundaries. Its speed, competitive with fast state-of-the-art methods, is unprecedented for diagram approaches. Extensive computational experiments show that Power-SLIC outperforms existing diagram approaches in boundary recall, under segmentation error, achievable segmentation accuracy, and compression quality. Moreover, Power-SLIC is robust to Gaussian noise.

6.1 Introduction

Superpixels are small, non-overlapping groups of connected, perceptually homogeneous pixels. Since its introduction by Ren and Malik in 2003 [RM03], superpixel generation has become an important pre-processing step in many imaging applications such as object localization [FVS09; Yan+15], image segmentation [Gou+08; Gou+14] and classification [Don+20; SARS20], optical flow [MG15], body model estimation [Mor05], object tracking [Shu+11], depth estimation [ZK07], image denoising [Li+16; MRR21], and background subtraction [Che+18; CSL20].

Superpixels reduce the number of inputs for subsequent algorithms and provide meaningful image representations from which application-relevant parameters can be extracted. It is commonly desirable that superpixels adhere well to object boundaries, are quickly generated, and compact [Ach+12; BLC18; SHL18].

Traditionally, superpixel algorithms are pixel-based because they operate on the pixel level, generating regions delimited by sets of pixels. Such pixel-level superpixel representations can often approximate small details and adhere well to boundaries. However, they are generally non-compact and free of higher-level geometric information.

Addressing this aspect, recent studies [AS17; BL18; DL15; KM20; Ma+21], have focused on developing geometric superpixel methods. These methods partition the image into small, preferably size-controllable, geometric objects, e.g., polygonal cells. As these objects ‘live’ in the continuous world, they provide a resolution-independent superpixel representation

and facilitate geometric manipulations, feature extractions, and other higher-level geometric post-processing tasks. The applicative potential of geometric superpixel methods has been demonstrated for diverse computer vision tasks such as object contouring [BL18; Ma+21], image vectorization [CXC14], image compression [FKF16; Ma+21], and segmentations of scenes containing man-made environments or other strong geometric signatures [DL15].

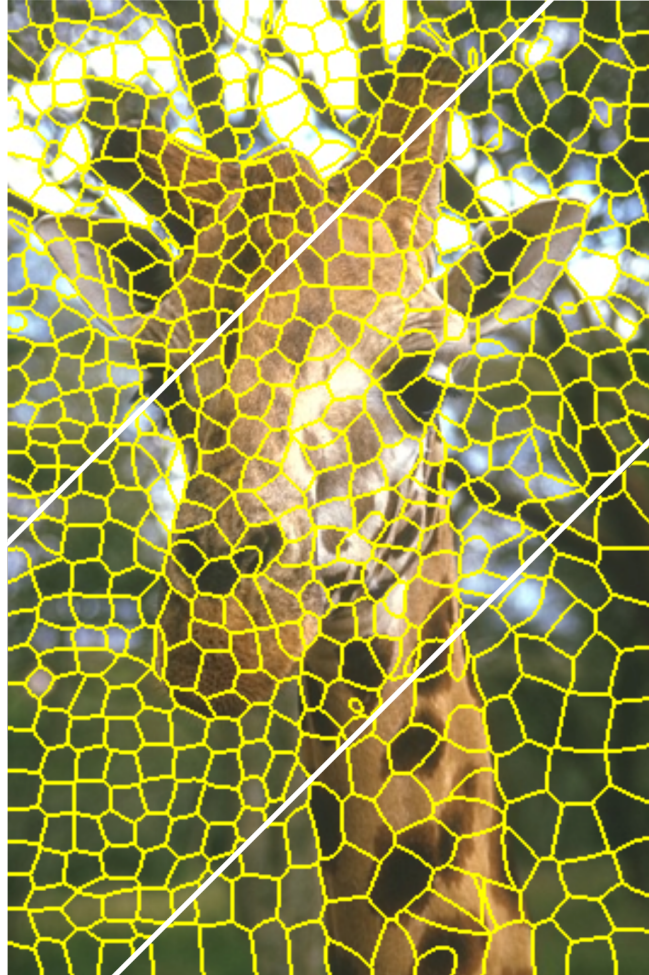


Figure 6.1: Superpixel segmentation using Power-SLIC with 900, 600, and 300 superpixels.

Many pixel methods, e.g., [Ach+12; JX12; Liu+16; Liu+18], employ Voronoi diagrams as intermediate structure, but they depart from it to better control the sizes and shapes of the superpixels. In contrast, the geometric methods Varane [DL15; DL] and, most recently, EC-CPD [Ma+21] return diagram cells as superpixels that are compact and sparsely representable. However, both of these diagram approaches employ time-consuming post-optimization to control the superpixel sizes and shapes. Motivated by unifying and extending the diagram approaches, we propose a novel method, called Power-SLIC, which handles non-linear boundaries and brings

the speed of state-of-the-art pixel methods to the geometric world.

6.1.1 Technical Contribution

The main technical contribution of our work is that for the first time, we provide a bridge between *pixel* and *geometric* superpixel methods. We achieve this by combining: (i) a linear programming primal-dual connection, established in [BG10] (see Theorem 6.2), between constrained clustering and diagrams, (ii) the heuristic [TR18] for approximating generalized balanced power diagrams, and (iii) SLIC’s initialization & assignment step.

Contribution to the geometric world: A diagram method that, for the first time, has a speed similar to fast pixel-based methods and that outperforms existing approaches in boundary recall, under segmentation error, achievable segmentation accuracy, and compression quality. To our knowledge, this is the first geometric approach capable of generating non-linear superpixel boundaries.

Contribution to the pixel world: A fast and noise-robust method that clusters pixels into superpixels but provides the benefits of diagram approaches such as providing sparse geometric superpixel representations by diagrams, facilitating geometric image operations, and providing relatively compact and regular superpixels at the same time.

6.1.2 Related Works

Superpixel algorithms are often categorized according to their high-level algorithmic approaches. One can distinguish, for instance, between graph-based, gradient descent-based, and clustering-based methods [Ach+12; SHL18; Wan+17]. In our context it is more appropriate to distinguish between pixel methods and geometric methods (see, e.g. [Ma+21]). The former use pixel representations in computations and for output, the latter aim at extracting continuous parameter representations of the superpixel boundaries. In the following we will give a brief overview. A comprehensive evaluation of many of the below mentioned superpixel algorithms can be found in [SHL18; Wan+17].

Pixel methods

Many popular state-of-the-art superpixel generation methods are pixel-based. A large number of pixel methods, such as SLIC [Ach+12], VCells [JX12], TurboPixel [Lev+09], and SEEDS [Van+15], employ clustering algorithms refining the clusterings until a convergence criterion is satisfied.

SLIC, a superpixel adaptation of the perhaps most influential and widely-used clustering method *k*-means (or Lloyd’s algorithm) [Llo82; Mac67], is particularly popular due to its simplicity and high performance. Among the recent approaches employing SLIC one can find applications in areas such as Dong2020 attack design [Don+20], remote sensing [ZCL19], and corrosion detection [HZX21]. A number of modifications and adaptations of SLIC have been proposed resulting in algorithms such as SLIC0 [Ach+20], SNIC [AS17], Manifold-SLIC [Liu+16], Intrinsic Manifold-SLIC [Liu+18], NR-SLIC [DZ16], dSLIC [Mai+18], and Dynamic Spectral Residual Superpixels [Zha+21].

Another class of pixel methods, like Entropy Rate Superpixel Segmentation (ERS) [Liu+11], Normalized Cuts [RM03], Linear Spectral Clustering (LSC) [LC15], Lazy Random Walk

(LRW) [Jia+14], Dynamic Random Walk (DRW) [KZM20] and qd-CSS [Ye+19] formulate the superpixel generation task as an optimization problem on a graph structure.

The recent success of deep networks in machine learning has inspired the development of several pixel methods that employ convolutional neural networks (CNNs). Among the first in this respect is the method Segmentation-Aware Loss (SEAL) [Tu+18]. It aims at learning pixel affinity from large data using a deep neural network. This is then fed into the rather time consuming ERS to generate the final superpixels. The Superpixel Sample Network (SSN) [Jam+18], on the other hand, integrates SLIC and performs faster. The iterative clustering step of SSN is simplified in the simple fully-connected convolutional network (SFCN) [Yan+20]. In [Zhu+21] the CNN-based superpixel segmentation is redefined as a lifelong clustering task and an unsupervised CNN-based method called LNS-Net is proposed. The method RIM [Suz20] trains a randomly initialized CNN to reconstruct the input image while minimizing the entropy among each superpixel. Aiming for topological consistency between the superpixels, [Pan+22] proposes a method that employs SSN for feature learning. While these methods can outperform conventional methods, it should be noted that they involve training a large network and generalization abilities of such networks tend to be limited.

It has been noted several times (e.g., in [BMZ17; Roe+16; SHL18]) that the performance of superpixel algorithms degrades severely as noise levels increase. To address this issue, [DZ16] proposes a Mahalanobis-based k -means variant (NR-SLIC). The Mahalanobis distance is also used in [BLC18] and [CG16], which introduce a Gaussian mixture model (GMMSP) and a unimodular Gaussian generative model (GGM), respectively. Gaussian mixture models based on Normal-Inverse Wishart priors and Inverse Wishart priors for the covariances are used in FSCSP [FLF15] and BASS [URF19], respectively. Common to these approaches is that they consider Mahalanobis distances for both the pixels' spatial and color components. Their clusterings in 5D, spatio-color clusterings, can be viewed as resulting from five-dimensional generalized balanced power diagrams (GBPDs) [Alp+15]. The diagram structure, however, is lost as the clusters are projected into 2D yielding the superpixels. As projections from 5D into 2D, their spatial cell boundaries are, however, rather irregular; see GMMSP and NR-SLIC in Fig. 6.5. A similar effect can be observed for the superpixels from [Don+22], where an anisotropic metric is used for the spatial components and an Euclidean metric for the color components. Our approach uses only the color components to determine the Mahalanobis distance, which, in turn, is only applied to the spatial components of the pixels. As we will demonstrate this will result in two-dimensional GBPDs with smooth, quadratic cell boundaries.

Geometric methods

Geometric methods (also known as resolution-independent or polygonal decomposition methods) aim at extracting continuous parameter representations of the superpixel boundaries.

The algorithm from [DL15], referred to as Varane [DL], partitions the image into convex polygonal superpixels by building Voronoi diagrams that conform to preliminarily detected line segments. The color values inside these Voronoi cells can be approximated; for example, [CXC14] proposes an approximation by polynomials. More general than Voronoi diagram cells, ECCPD from [Ma+21] proposes to segment images into power diagram cells. The algorithm iterates between updates of sites and weights and applies an edge-alignment optimization step.

Mesheres of convex polygonal cells that are guaranteed to have no small angles are generated in [FKF16]. An improved method is given in [KM20]. The algorithms SNIC and SNICPOLY

from [AS17] produce superpixels with piecewise linear boundaries. KIPPI from [BL18] uses a kinetic approach that extends line segments until they meet each other. KIPPI can generate polygons of different sizes, avoiding over-segmentations of large uniform areas, but it does not allow explicit control over the number of superpixels. A Delaunay Point Process (DPP), which segments images into triangular patches, is proposed in [Fav+20].

Noticeably, many algorithms compute an intermediate diagram structure. Diagrams are returned as final superpixel segmentation only in Varane and ECCPD. Therefore, we refer to them as diagram approaches. They return Voronoi and power diagrams, respectively, which have linear boundaries. In contrast, our diagram class of GBPDs contains the latter diagram classes as subclasses. Boundaries can be quadratic, but the cell convexity may be lost.

6.2 Overview of our Approach

We borrow the initialization & assignment step from SLIC (Simple Linear Iterative Clustering) [Ach+12] to obtain possibly non-connected groups of pixels, which typically have a high boundary adherence but fuzzy boundaries. In contrast to SLIC’s post-processing, which guarantees connected superpixels, we compute a generalized balanced power diagram (GBPD) based on the spatial Mahalanobis distance determined for every previously obtained group of pixels. The cells of the GBPD will ultimately be the resulting superpixels. As the computation of GBPDs can be time-consuming, we approximate it to achieve a competitive running time. A high-level visualization of Power-SLIC is illustrated in Fig. 6.2.

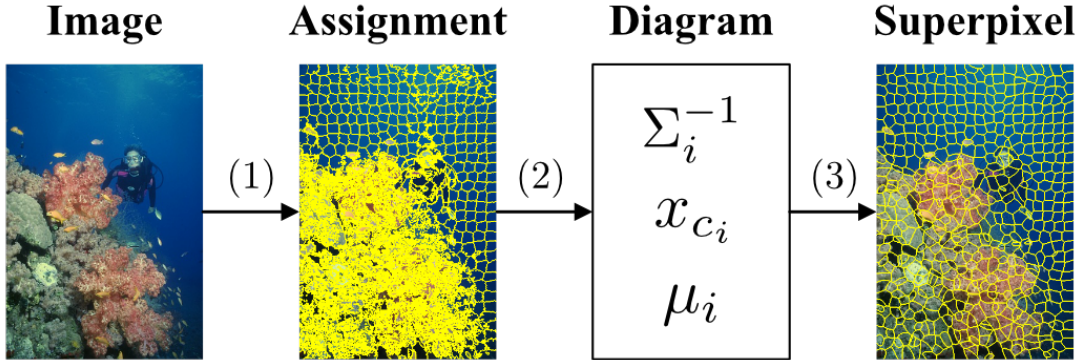


Figure 6.2: Main steps of Power-SLIC: (1) Perform SLIC’s initialization & assignment step. (2) Extract diagram parameters from the assignment yielding a geometric representation of the superpixels. (3) Assign each pixel to its containing diagram cell (yielding a pixel representation of the superpixels).

SLIC’s assignment phase ideally suits our purposes, as it constructs a five-dimensional Voronoi diagram, a special case of a GBPD, with linear boundaries. Our approach aims at approximating and regularizing this 5D structure by a GBPD with quadratic cell boundaries in the two spatial dimensions.

6.3 SLIC's Initialization & Assignment Phase

SLIC [Ach+12] is seen as one of the most popular and widely used methods for superpixel generation. Given a color image of N pixels, SLIC clusters the pixels within the five-dimensional space (x_p, l_p) , with x_p denoting the spatial position of pixel p (a two-dimensional vector) and l_p its color (a three-dimensional vector) in the CIELAB color space. The algorithm is divided into an initialization & assignment and a post-processing phase. The latter phase ensures connectivity by reassigning near-by superpixels. As we do not make use of this phase, we omit a description.

Initialization & Assignment

SLIC takes as its input two arguments k and m , with k specifying the targeted number of superpixels and m denoting a compactness parameter. Throughout this text, we use $m = 10$ as this is SLIC's default value (this, and the placement of cluster centers as discussed next, could, of course, be further optimized and tailored to specific situations if placed into a supervised learning framework; see, e.g., [Don+20]). Initially, SLIC samples cluster centers with spatial positions from a square grid spaced $h = \sqrt{N/k}$ pixels apart. Subsequently, pixels are assigned to their closest cluster center. However, only local assignments of pixels lying within a $2h \times 2h$ spatial window around a cluster center are considered to improve performance. Proximity is measured by a weighted Euclidean distance. The second parameter, m , determines the weight of the Euclidean distance in the color space. For a pixel p and a center c , the squared distance d^2 is given by:

$$(6.1) \quad d^2(p, c) = \|x_p - x_c\|_2^2 + \frac{h^2}{m^2} \cdot \|l_p - l_c\|_2^2.$$

As in traditional k -means, the centers are updated as the centroids of the updated clusters, and the process is iterated until some stopping criterion is fulfilled.

6.4 Generalized Balanced Power Diagrams

Generalized balanced power diagrams (GBPDs) [Alp+15] generalize the concept of power diagrams, which themselves generalize the famous Voronoi diagrams. Here, we are given a set of k distinct sites $S = \{s_1, \dots, s_k\} \subseteq \mathbb{R}^d$. Each site is equipped with a local ellipsoidal norm defined by a positive definite symmetric matrix $A_i \in \mathbb{R}^{d \times d}$. The norm $\|\cdot\|_{A_i}$ is defined as $\|x\|_{A_i} := \sqrt{x^\top A_i x}$ for $x \in \mathbb{R}^d$, $i \in [k]$. Moreover, let $\mu_1, \dots, \mu_k$ be scalars serving as additional balancing parameters.

Definition 6.1

For every $i \in [k]$, the *generalized balanced power diagram cell* is defined as:

$$(6.2) \quad P_i := \{x \in \mathbb{R}^d : \|x - s_i\|_{A_i}^2 + \mu_i \leq \|x - s_l\|_{A_l}^2 + \mu_l, \quad \forall l \in [k]\}.$$

We call the k -tuple of cells $(P_1, \dots, P_k)$ a *generalized balanced power diagram (GBPD)*.

Throughout this chapter, we consider only two-dimensional GBPDs, although our approach canonically extends to images and GBPDs in arbitrary dimensions.

Further, we remark that we obtain power diagrams in the special case in which each $A_1, \dots, A_k$ are the identity matrix. If additionally $\mu_1 = \dots = \mu_k$, we obtain Voronoi diagrams.

6.5 Optimal Power-SLIC and Power-SLIC

Given the assignment L computed by SLIC's initialization & assignment phase, we propose computing the following statistics for each label i in L : the spatial center x_{c_i} , the spatial covariance matrix $\Sigma_i \in \mathbb{R}^{2 \times 2}$, and the area κ_i of each possibly non-connected component $C_i = \{x_p : L(p) = i\}$. From this we compute a GBPD with sites $s_i = x_{c_i}$ and matrices $A_i = \Sigma_i^{-1}$, such that the area of its cells P_i equals or approximates κ_i , $i \in [k]$. The cells $P_1, \dots, P_k$ of the GBPD will be the resulting superpixels.

Optimal Power-SLIC

We first present an algorithm to compute GBPDs as described above. As in [Alp+15], we can achieve this by exploiting a strong relation between constrained clustering and (anisotropic) power diagrams shown in [BG10] (see also [BG12]). We consider the following linear program:

$$\begin{aligned}
 \text{(P)} \quad & \min \sum_{i=1}^k \sum_{j=1}^N \xi_{ij} \|x_{p_j} - s_i\|_{A_i}^2 \\
 & \text{s.t. } \sum_{i=1}^k \xi_{ij} = 1, \quad j \in [N], \\
 & \sum_{j=1}^N \xi_{ij} = \kappa_i, \quad i \in [k], \\
 & \xi_{ij} \geq 0, \quad i \in [k], j \in [N].
 \end{aligned}$$

Its dual is given by:

$$\begin{aligned}
 \text{(D)} \quad & \max \sum_{j=1}^N \eta_j - \sum_{i=1}^k \kappa_i \mu_i \\
 & \text{s.t. } \eta_j \leq \|x_{p_j} - s_i\|_{A_i}^2 + \mu_i, \quad i \in [k], j \in [N].
 \end{aligned}$$

The problem modeled by (P) is the balanced least-squares assignment problem. The task is to assign x_{p_j} to sites in such a way that precisely κ_i of them are assigned to site s_i , for each $i \in [k]$, while the sum of squared distances of the assigned x_{p_j} to their assigned sites, measured via the norm $\|\cdot\|_{A_i}$, is minimized. As the constraint matrix of the problem is totally unimodular, there will be a binary optimal solution to (P), i.e., in this solution, ξ_{ij}^* is either 0 or 1 where $\xi_{ij}^* = 1$ if, and only if, x_{p_j} is assigned to site s_i .

Optimal solutions to (P) can be characterized in terms of GBPDs. To state this precisely, we apply the following notion: a diagram is said to induce a binary assignment if, and only if, all x_{p_j} assigned to the same cluster lie in the same diagram cell.

Algorithm 1: Optimal Power-SLIC.

Data: Image, targeted number k of superpixels.
Result: Assignment, GBPD.
 Run SLIC's initialization & assignment phase with parameters k and $m = 10$ to obtain assignment L .
for each pixel label i in L **do**
 Compute spatial center x_{c_i} , covariance matrix Σ_i , and area κ_i of C_i .
 for each pixel p_j inside of C_i 's smallest bounding box **do**
 Generate the variable ξ_{ij} of (P).
 end
end
 Solve (P) for $s_i = x_{c_i}$ to retrieve both primal ξ_{ij}^* , and dual solution μ_i^*, η_j^* ,
 $i \in [k], j \in [N]$.
 Set $\mu = (\mu_1^*, \dots, \mu_k^*)$.
return assignment (computed from the ξ_{ij}^*) and diagram parameters
 $(\Sigma_i^{-1}, x_{c_i}, \mu_i), i \in [k]$.

Theorem 6.2 (Special case of [BGK17])

A balanced assignment is a basic solution of the linear program (P) if and only if there exists a parameter vector $\Gamma \in \mathbb{R}^k$ such that the GBPD with parameters $(A_i, s_i, \mu_i), i \in [k]$, induces this binary assignment. Moreover, with a solution (μ^*, η^*) of (D) that fulfills strict complementary slackness, we can choose $\mu = \mu^*$.

Hence, when optimally assigning pixels to superpixel centers, i.e., for $s_i = x_{c_i}$, superpixel areas prescribed to equal κ_i , and distance matrices $A_i = \Sigma_i^{-1}$, the resulting superpixels can be characterized by cells of a GBPD. Moreover, the diagram parameters are extracted from the dual solution and yield a low-dimensional, resolution-independent representation of the superpixels (an algebraic representation of the boundary between two neighboring cells is provided by the quadratic equation that results from changing the respective inequality relation in (6.2) to an equality relation).

In terms of computational complexity, we remark that (P) has Nk variables and $2N + k$ constraints. The linear program can, however, be solved in $\mathcal{O}(N^3)$ using the Hungarian method, for example. In our evaluation, we use the dual simplex of Gurobi [Gur20]. To improve performance, we apply a locality assumption similar to SLIC by adding only those variables ξ_{ij} for which p_j lies within a regional window of the center c_i . To ensure that every pixel is assigned, we choose this window to be the smallest bounding box of C_i . The resulting algorithm, called Optimal Power-SLIC, is summarized in Algorithm 1.

We found that building the model takes up most of the time, while solving the model usually requires less than 10s per image. This suggests that more significant speed-ups can be achieved by avoiding Gurobi calls, e.g., by using a more specialized and adapted solver, such as min-cost flow.

Power-SLIC

To match the SLIC linear running time, we now introduce a heuristic version of Optimal Power-SLIC, which we call Power-SLIC. The main ideas are rooted in two key observations.

First, we observe that most of the computation time in Optimal Power-SLIC is spent on computing $\mu_1, \dots, \mu_k$ while the sites and norm matrices of the diagram are essentially readily available. To effect a significant speed-up, we replace the computation of the optimal μ_i with an estimation. The idea is to think of μ_i as a scaling factor and estimate it such that the ellipsoid given by the inverse covariance matrix $A_i = \Sigma_i^{-1}$ has an area equal to κ_i . Thus, we want to choose μ_i such that

$$\text{Area}(\|x\|_{\Sigma_i^{-1}}^2 + \mu_i \leq 0) = \kappa_i$$

which is equivalent to

$$\text{Area}(\|x\|_{\Sigma_i^{-1}}^2 \leq -\mu_i) = \kappa_i.$$

Since the area of such an ellipsoid is given by $\pi\sqrt{\mu_i^2 \det \Sigma_i}$, we solve $\kappa_i = \pi\sqrt{\mu_i^2 \det \Sigma_i}$ for μ_i and choose the appropriate solution for which $-\mu_i > 0$, hence

$$-\mu_i = \frac{\kappa_i}{\pi\sqrt{\det(\Sigma_i)}}.$$

Our evaluation will show that this choice provides a very good compromise between speed and boundary adherence. While, generally, cell sizes cannot be guaranteed to lie within the prescribed bounds anymore, we remark that a theoretical foundation of this choice of the μ_i can be given within the framework of Bayes classifiers; see [Alp+23a]. (For a use of Section 6.5 in a materials science context, see [TR18].)

The second observation relates to the fact that the superpixel statistics can be extracted efficiently from the final SLIC assignment iteration without requiring any additional passes over the image data. While this is also true for Optimal Power-SLIC, it becomes particularly relevant with fast implementations of Power-SLIC. In fact, the spatial centers $x_{c_1}, \dots, x_{c_k}$ can be obtained after the final SLIC iteration simply by projecting $c_1, \dots, c_k$ into the spatial image space by dropping their color components. The spatial covariance matrices $\Sigma_1, \dots, \Sigma_k$, can be computed together with the centers within the final iteration of the SLIC assignment, using a single pass algorithm for the variance.

With these parameters, Power-SLIC computes a final assignment that aims at assigning each pixel p_j to its closest spatial center x_{c_i} , $i \in [k]$. Distances are measured in terms of $\|x_{p_j} - x_{c_i}\|_{\Sigma_i^{-1}}^2 + \mu_i$. As in SLIC, we consider only those pixels lying in a $2h \times 2h$ window around the cluster center for the final assignment. In a cleaning step, the still unassigned pixels are assigned to its containing diagram cell. In our experiments, this cleaning step was hardly ever required. Thus, not even a naive implementation would affect the practical running time. A summary of Power-SLIC is given in Algorithm 2.

Theorem 6.3

Power-SLIC runs in $\mathcal{O}(N)$.

Proof. SLIC's initialization & assignment phase runs in $\mathcal{O}(N)$, the computation of the diagram parameters is on the fly, and the diagram cell computation for obtaining the superpixels is essentially one additional assignment iteration with a modified distance measure. Thus, we obtain an overall running time of $\mathcal{O}(N)$, which is independent of the number of superpixels k . $\square$

Algorithm 2: Power-SLIC.

Data: Image, targeted number k of superpixels.
Result: Assignment, GBPD.
 Run SLIC's initialization & assignment phase with parameters k and $m = 10$ to obtain spatial centers x_{c_i} and spatial covariance matrices Σ_i , $i \in [k]$.
 Compute μ_i using Section 6.5.
 Set $L(p_j) = -1$ and $D(p_j) = \infty$ for each pixel p_j .
for each cluster $i \in [k]$ **do**
 for each pixel p_j in a spatial $2h \times 2h$ window around x_{c_i} **do**
 if $\|x_{p_j} - x_{c_i}\|_{\Sigma_i^{-1}}^2 + \mu_i < D(p_j)$ **then**
 Set $D(p_j) = \|x_{p_j} - x_{c_i}\|_{\Sigma_i^{-1}}^2 + \mu_i$.
 Set $L(p_j) = i$.
 end
 end
end
for each pixel p_j with $L(p_j) = -1$ **do**
 Determine cell P_i of the GBPD containing p_j and set $L(p_j) = i$.
end
return L and diagram parameters $(\Sigma_i^{-1}, x_{c_i}, \mu_i)$, $i \in [k]$.

6.5.1 Resolution-Independent Power-SLIC

This section designs a fast superpixel algorithm for high-resolution images by exploiting the resolution-independence of Power-SLIC. Traditional superpixel algorithms require significant time to process high-resolution images. SLIC is 40 times slower for images of size 2048x1536 than for images of size 320x240 [Ach+12]. Due to the linear runtime and the larger image having 40 times more pixels, this matches expectations. In absolute numbers, SLIC requires about 5 seconds on images of size 2048x1536 despite being one of the fastest superpixel algorithm. Hence, we require new methods for fast superpixel generation for high-resolution images.

As Power-SLIC computes an intermediate diagram representation of superpixels, we can easily scale this representation to different resolutions by scaling the position of the diagram sites as described in Algorithm 3. Together with the identical diagram parameters computed on the low-resolution image, we obtain diagrams cells that represent superpixels for the full-resolution image. This allows us to run the assignment phases with a much smaller image resolution. Since the assignment phase dominates the runtime, this leads to a significant speedup of the algorithm including the additional cost for scaling down the image. The process of Resolution-Independent Power-SLIC is depicted in Fig. 6.3. We provide the formal algorithm in Algorithm 3.

Note that Resolution-Independent Power-SLIC takes an RGB-image as input, reduces its resolution and then converts the low-resolution image into the CIELAB color space. This order is much faster than converting the full-resolution image into the CIELAB space.

To reduce the resolution of an image, we resample using pixel area relations. This method

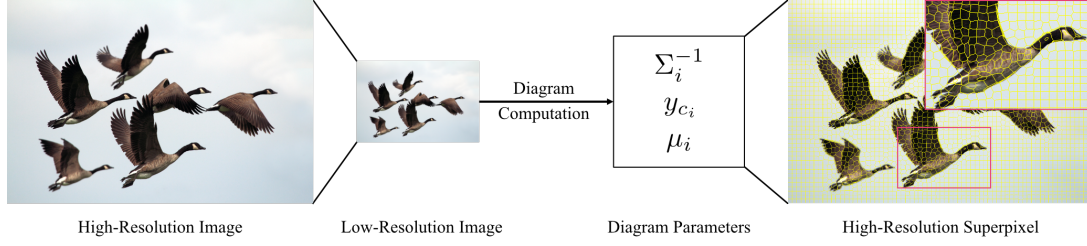


Figure 6.3: Power-SLIC extracts diagram parameters from a low-resolution image that are scaled afterward.

can be described as follows. For a *scale* parameter, the image is divided into windows of size

$$\frac{1}{scale} \times \frac{1}{scale}$$

starting with the first windows in the top left corner of the full resolution image. By choice of the window size, each window corresponds to a pixel in the scaled image. Each pixel of the scaled image is computed as an average over all pixels that lie in a window. If a pixel, considered a 1×1 square, is partially contained in multiple windows, a weighted average is computed where the weight equals the pixel area in each window. We motivate this method of reducing the resolution in the following.

Connection to Coresets Reducing an image's resolution raises the question if the low-resolution image contains sufficient information to extract good superpixels for the high-resolution image. Of course, Power-SLIC is a heuristic by nature. Nonetheless, we interpret the low-resolution image as a coreset. To the best of our knowledge, using coresets for the computation of superpixels is new to the literature. The authors of [BNS10] compare the idea of summarizing pixels in a superpixel with summarizing data points in a weighted point, a

Algorithm 3: Resolution-Independent Power-SLIC

Data: RGB-Image with width w and height h , *scale* in $(0, 1]$, targeted number k of superpixels.

Result: Assignment, GBPD.

Scale the image to a width of $w \cdot scale$ and a height of $h \cdot scale$

Run SLIC's initialization & assignment phase with parameters k and $m = 10$ to obtain spatial centers x_{c_i} and spatial covariance matrices Σ_i , $i \in [k]$.

Compute μ_i using Section 6.5.

Set $y_{c_i} = x_{c_i}/scale$ for each $i \in [k]$.

Continue with the assignment of pixel as in Power-SLIC but for the full resolution image.

return L .

technique used by many coresets. However, the authors of [BNS10] do not make this connection formal or use coresets for calculating superpixel.

We recap that the Power-SLIC assignment phase is a heuristic version of least-squares clustering in 5-dimensional space. In Chapter 3, we showed that reducing the resolution of points on a grid results in a coreset for least-squares clustering. We can interpret the 5-dimensional image data as points lying on a (sparse) grid of size $h \times w \times [0, 255]^3$ where the last three dimensions represent the RGB color value of each pixel. Reducing the resolution can be interpreted as follows: A window of size

$$\frac{1}{scale} \times \frac{1}{scale} \times 255 \times 255 \times 255$$

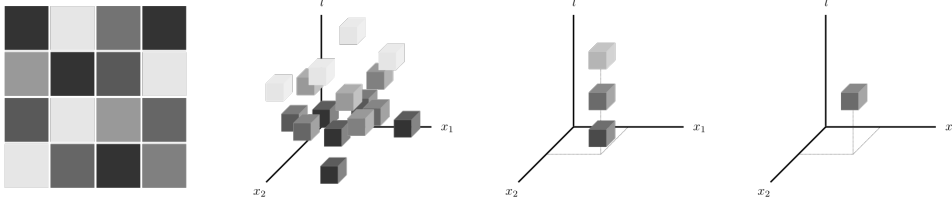
is moved across the 5-dimensional image space. In the first two spatial dimensions the approximation depends on the selected scale and points in the same window are replaced with a weighted average. In each color channel the approximation is quite strong and independent of the scale parameter, as all colors are simple averaged. In particular, if two pixels with strongly different colors are in the same window, the approximations are worse compared to the case where all pixels in the same bin are of similar color.

Note that this process is identical to our coreset construction introduced in Chapter 3 with a good approximation (small ϵ for the coreset construction) in the first two dimensions and a coarse approximation in the color dimensions (large ϵ for the coreset-construction). This shows that creating the 5-dimensional point space used by Power-SLIC of the high-resolution image and applying our coreset construction from Chapter 3 to it (with different epsilons for the first two dimensions and the last three) is identical to reducing the resolution of the original image and then creating the 5-dimensional point space. As a result, our coreset result in Chapter 3 give us guarantees on the quality of the solution.

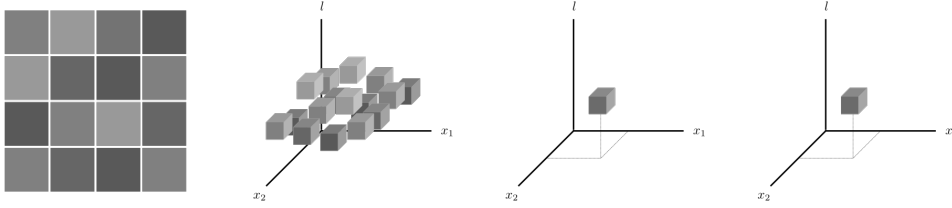
Naturally, this raises the question why we do not apply the same ϵ to all five dimensions in the point space of the high-resolution image and compute the clustering on the coreset in this space. Theoretically, this should result in better approximation guarantees for the clustering phase of Resolution-Independent Power-SLIC, as we create a better approximation for the color space. However, in the final step of Resolution-Independent Power-SLIC, we still require to assign each pixel to exactly 1 superpixel, meaning that we would need to create a decision-rule anyway when projecting multiple points from the 5-dimensional space back to the identical position in the 2-dimensional spatial image space.

The difference between reducing the resolution and constructing a coreset is depicted in Fig. 6.4. Figure 6.4b shows a situation in which both constructions coincide, whereas Fig. 6.4a shows an example where they differ. We will understand in Section 6.6.1 in more detail under which image conditions this coarse color approximation works well and under which conditions caution is required.

Independent of creating a coreset with multiple color bins or by reducing the resolution, we can compute the least-squares clustering in the assignment phase of Power-SLIC using the coreset. The runtime reduces to $\mathcal{O}(\tilde{n})$ with $\tilde{n}$ being the number of points in the coreset. Moreover, as the coreset bins each color channel separately, Resolution-Independent Power-SLIC works like Power-SLIC on any number of image channels, including greyscale and 3D images with depth information.



- (a) The grayscale image has strongly varying grey values. As a result, points are spread along the l -axis. The coreset bins the color channel represented by a grayscale image into multiple points at the same spatial position. The number of bins is determined by the chosen approximation guarantee of the coreset. Reducing the resolution simply averages over all color values which results in a single point.



- (b) The grayscale image has very similar grey values. As a result, points are close to each other along the l -axis. The coreset bins all points in the same color bin as the resolution reduction does. Thus, the coreset and the resolution reduction are identical.

Figure 6.4: Comparing the coreset and the resolution reduction. From left to right: A grayscale image, a resulting 3-dimensional point space used by Power-SLIC and its variants (2 spatial, one color channel), the resulting coreset, the result of reducing the resolution. ($scale = 0.25$).

6.6 Evaluation

In this section, we compare Power-SLIC and Optimal Power-SLIC on the Berkeley Segmentation Data Set 500 (BSDS500) [Mar+01] with the state-of-the-art superpixel algorithms SLIC [Ach+12], SLIC0 [Ach+20], NR-SLIC [DZ16], GMMSP [BLC18], Varane [DL15], ECCPD [Ma+21], KIPPI [BL18], ERS [Liu+11], SNIC [AS17], and SEEDS [Van+15].

Quantitative Evaluation

For our quantitative evaluation, we compute boundary recall, undersegmentation error, achievable segmentation accuracy, and compactness for k number of superpixels between 300 and 2,000. Additionally, we report the running times and peak signal-to-noise ratios for image compression for various k . Finally, we measure the noise robustness by considering the trade-off between boundary recall and compactness.

Boundary recall (BR) measures the percentage of detected ground truth boundary pixels. With T_p and F_p denoting the number of true positives and false negatives, respectively, BR is defined as $BR = T_p / (T_p + F_n)$. In practice, we follow [Ach+12] and classify a ground truth

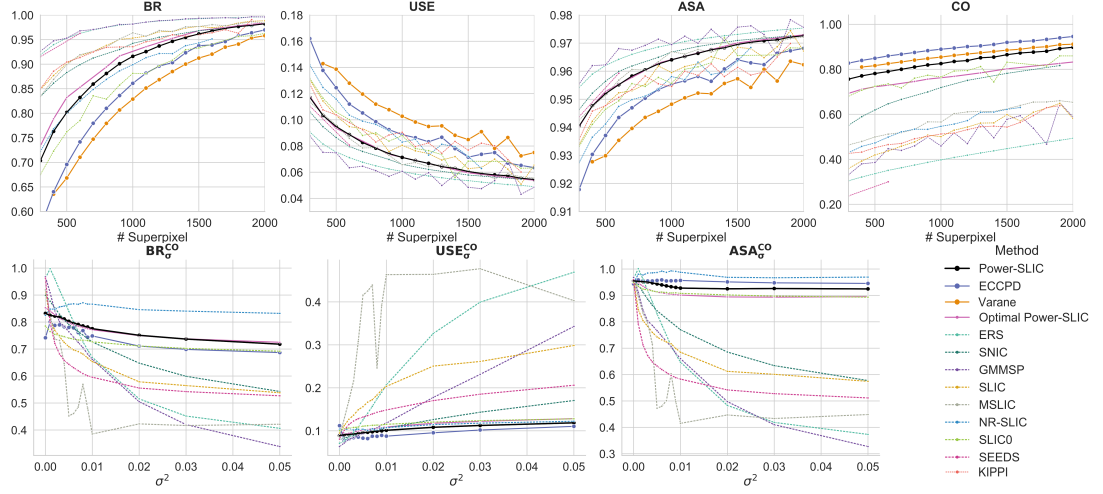


Figure 6.5: Top: Quantitative comparison of multiple superpixel algorithms. Bottom: Noise robustness of multiple superpixel algorithms and $k = 600$. Here, no results are shown for Varane and KIPPI because the superpixels could not be obtained for noisy images.

boundary pixel as a true positive if its spatial two-neighborhood contains a superpixel boundary pixel. Accordingly, F_p equals the number of superpixel boundary pixels not in a spatial two-neighborhood of a ground truth boundary pixel.

We also compute the undersegmentation error (USE), which increases when different ground truth regions are grouped into the same superpixel. With $\mathcal{G}$ denoting the set of segments in the ground truth and $\mathcal{S}$ denoting the set of superpixels, USE can be defined as:

$$USE = \frac{1}{\sum_{G \in \mathcal{G}} |G|} \left(\sum_{S \in \mathcal{S}} \sum_{G \in \mathcal{G}} \min(|S \cap G|, |S \setminus G|) \right).$$

The achievable segmentation accuracy (ASA) provides an upper bound on the best obtainable segmentation accuracy of the image when segmenting on the superpixel level. It is defined as

$$ASA = \frac{\sum_{S \in \mathcal{S}} \max_{G \in \mathcal{G}} |S \cap G|}{\sum_{G \in \mathcal{G}} |G|}.$$

To evaluate compactness, we compute the compactness of the superpixels as introduced in [SFS12]. Let b_i and κ_i denote the i th superpixel's total number of boundary pixels and area. With Q_i the iso-perimetric quotient, compactness (CO) is defined as:

$$Q_i = \frac{4\pi\kappa_i}{b_i^2}, \quad CO = \sum_{i=1}^k Q_i \frac{\kappa_i}{N}.$$

Compactness essentially measures the similarity of the superpixels to a disc.

Figure 6.5 provides a comparison among the superpixels algorithms. First, note that the algorithms Varane, ECCPD, and (Optimal) Power-SLIC are the only algorithms that compute a low-dimensional diagram representation of the superpixels. As these superpixels are diagram cells, they result in high compactness values.

While Power-SLIC generates slightly less compact superpixels than Varane and ECCPD, it outperforms the diagram approaches in BR, USE, and ASA for any number k of superpixels. Remarkably, Power-SLIC’s performance in these categories falls into the range achieved by pixel methods.

Pixel methods are often highly sensitive to changing image conditions and can fail when small amounts of noise are present in the image. For example, looking at GMMSP in Fig. 6.6 its superpixel segmentation degenerates quite considerably if noise is added: The algorithm identifies too many image boundaries, and almost all pixels are identified as boundaries. While this results in a high boundary recall by definition, the segmentation deteriorates and the compactness value decreases significantly. It is therefore reasonable to penalize the measures BR, ASA, USE for a decreasing compactness value; see [Ma+21], which also relates the measures to the compactness value. Let CO_σ denote the compactness value for the noise level σ^2 . Similarly, we denote by BR_σ , ASA_σ , USE_σ the measures at noise level σ^2 . We define

$$\begin{aligned} BR_\sigma^{CO} &= BR_\sigma \cdot \frac{CO_\sigma}{CO_0}, & ASA_\sigma^{CO} &= ASA_\sigma \cdot \frac{CO_\sigma}{CO_0}, \\ USE_\sigma^{CO} &= USE_\sigma \cdot \frac{CO_0}{CO_\sigma}. \end{aligned}$$

In other words, we consider the compactness relatively to the compactness for noise-free images so that the measures do not depend on the compactness of the superpixel segmentation for noise-free images. For a perfectly noise-robust superpixel algorithm, we expect the measures to be constant and therefore independent of σ . Figure 6.5 summarizes the results for $k = 600$ (a similar behavior is observed for other typical values of k). We observe that NR-SLIC, an algorithm specifically designed for noisy images, performs best overall. On the other hand, those algorithms with very high boundary recall on clean images tend to deteriorate quickly, even for small amounts of noise ($\sigma^2 < 0.01$), mainly because they detect too many “image boundaries” in the noisy cases (see also Fig. 6.6). Power-SLIC and ECCPD are relatively robust to noise, and their performance decreases only slightly with increasing noise.

We now focus on two aspects of the diagram approaches. First, we measure computational speed, a bottleneck of diagram methods so far. Second, we assess the compression quality achieved by the highly sparsely-encoded superpixel segmentation provided by diagrams.

Timings are shown in Table 6.1. Power-SLIC is several orders of magnitude faster than the other diagram approaches and its running time increases only slightly with k .

As mentioned above, diagram approaches are particularly suited to provide sparse image encodings on which one can base subsequent higher-level computer vision tasks or employ them for image compression [Ma+21]. Using Power-SLIC, for instance, one can represent and store an image by specifying $9k$ (floating point) parameters: for each superpixel, two for its cell site, three for covariance matrix, one for the size parameter, and three for the average color.

Following papers such as [Ma+21], we evaluate the quality of the generated sparse image representation by reporting the peak signal-to-noise ratio (PSNR) between the original and compressed image. With x_{p_j} and $\hat{x}_{p_j}$ denoting the color value of the j -th pixel in the original

		$k=500$	$k=1,000$	$k=1,500$	$k=2,000$
Time (in s)	Varane	2.549	10.191	18.620	47.022
	ECCPD	28.703	96.408	149.929	171.321
	Power-SLIC	0.070	0.074	0.078	0.080
PSNR	Varane	29.667	29.998	30.271	30.318
	ECCPD	29.716	30.101	30.369	30.259
	Power-SLIC	30.201	30.578	30.840	31.027

Table 6.1: Top: Running times on a laptop with an Intel Core i7 5600U processor. Bottom: Peak signal-to-noise ratios.

and compressed image, respectively. the PSNR is defined as

$$MSE = \frac{1}{3N} \sum_{j=1}^N \|x_{p_j} - \hat{x}_{p_j}\|_2^2,$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{MSE} \right).$$

Larger PSNR values indicate a better quality of the compressed image. The resulting PSNR values for the BSDS500 dataset are shown in Table 6.1. In all cases, Power-SLIC gives the highest PSNR values despite having the smallest running times. It seems that the anisotropic diagram structure, involving quadratic boundaries, allows Power-SLIC to improve on Varane and ECCPD, both of which use less flexible diagram classes.

Qualitative Evaluation

Table 6.2 provides a qualitative comparison of multiple superpixel algorithms. (Algorithms are classified as ‘diagram-based,’ ‘fast,’ ‘noise-robust,’ and ‘compact’ if their superpixels are described by diagram cells, their average running times for $k = 500$ are below 1 second, all their plots in Fig. 6.5 are close to horizontal lines, and their plots are in the top of the two groups for compactness in Fig. 6.5.) Figure 6.6 gives a qualitative comparison of ten superpixel algorithms for varying levels of noise.

A comparison of the post-processing steps of Power-SLIC and SLIC is given in Fig. 6.7. Although Power-SLIC uses SLIC’s clustering approach, it is notable that the superpixels are fundamentally different due to the post-processing step. This difference can be seen by comparing the results with the second column of Fig. 6.7, which depicts the clustering result of SLIC without any post-processing. Only the local post-processing by SLIC shown in the third column creates a reasonable segmentation, which deteriorates for more extensive noise levels. In contrast, Power-SLIC optimizes more globally and returns a diagram representation. At the same time, this approach acts as a regularizer for high noise levels.

	Diagram	Fast	Compact	Noise-Robust
Power-SLIC	✓	✓	✓	✓
Varane	✓	-	✓	-
ECCPD	✓	-	✓	✓
ERS	-	✓	-	-
SNIC	-	✓	✓	-
GMMSP	-	✓	-	-
SLIC	-	✓	-	-
MSLIC	-	✓	-	-
NR-SLIC	-	✓	-	✓
SLIC0	-	✓	✓	✓
KIPPI	-	✓	-	-
SEEDS	-	✓	-	-

Table 6.2: Qualitative comparison of multiple superpixel algorithms. Power-SLIC is the only fast diagram-based superpixel algorithm.

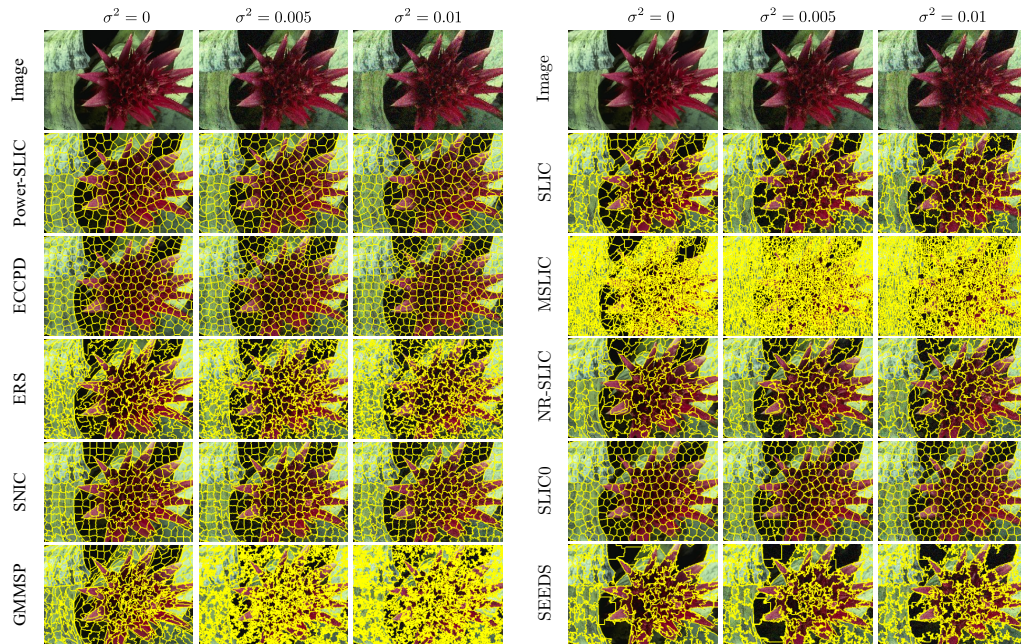


Figure 6.6: Visual comparison of five superpixel algorithms for various levels of noise ($k = 700$).

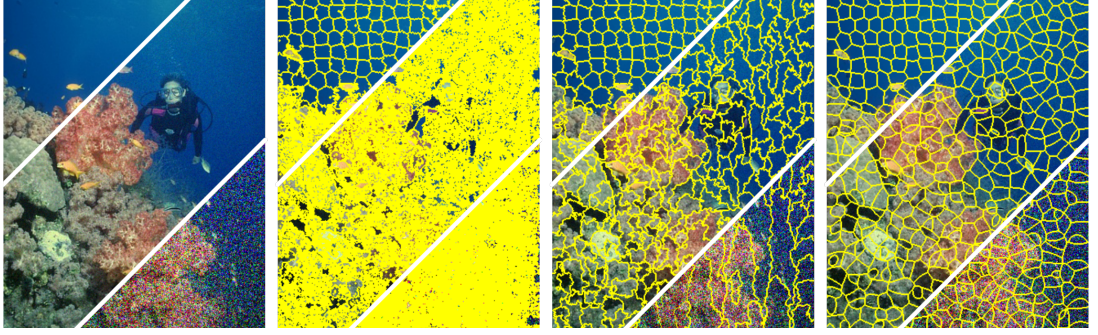


Figure 6.7: Algorithms applied to an image with three levels of noise $\sigma^2 \in \{0, 0.001, 0.1\}$ and $k = 700$. Left to right: original image, SLIC before post-processing, SLIC, and Power-SLIC.

6.6.1 Evaluation: Resolution-Independent Power-SLIC

Here, we empirically evaluate Resolution-Independent Power-SLIC for different scale parameters. Having measured boundary recall, boundary precision, and compactness of Power-SLIC on full-resolution images, we can infer the quality of superpixels generated on a low-resolution image by comparing them with the superpixels generated on the full-resolution image. This is particularly helpful as there is a lack of high-resolution data sets with a ground truth containing image boundaries. To test Resolution-Independent Power-SLIC on high-resolution images, we use the DIV2K data set, which includes diverse color images with 2040px in the longest image axis [AT17].

The idea of SLIC and Power-SLIC is to group pixels of similar color that are spatially connected. Therefore, we can represent the i -th superpixel C_i by its average color value. Thus, we can compare two superpixels C_i and C'_i by comparing the average deviation in each color channel. To compare two superpixel segmentations, we need to specify a pairing of superpixels for comparison. For this, we identify a superpixel by its position in the initial grid of the algorithm. As this grid is independent of the scale parameter, we can pair the two superpixels with the same position in the grid. The average color channel deviation is abbreviated by $ACCD$ and defined as

$$ACCD = \frac{1}{3k} \sum_{i=1}^k \left\| \frac{1}{|C_i|} \sum_{j \in C_i} z_j - \frac{1}{|C'_i|} \sum_{j \in C'_i} z_j \right\|_1 \in [0, 1],$$

where $\|\cdot\|_1$ is the 1-norm. The factor $1/3k$ normalizes the term to be in $[0, 1]$ assuming three RGB channels, each in $[0, 1]$, and k superpixels. An alternative could be to pair superpixels by their spatial proximity in the resulting segmentation for example via solving a minimal matching problem. Such an approach is more general and will consider two superpixel segmentations closer to each other than our fixed order. However, we do not require the additional complexity, and $ACCD$ will suffice to show closeness of the resulting segmentations.

Figure 6.8 compares the $ACCD$ and the runtime of Resolution-Independent Power-SLIC

for various scale parameters computed on the validation data set of the DIV2K data set. We observe that the superpixel computation for the high-resolution images takes an average of more than 5s and can be reduced to less than 600ms when reducing the resolution by a scale of 0.3. For an image of size 2040x2040, a scale factor of 0.3 means that the resulting image is 612x612. This is a reduction of 91% in the number of image pixels. The significant reduction in runtime comes with an $ACCD$ between 0.0175 and 0.021, depending on k .

Generally, the larger k , and therefore k , the higher is the color channel deviation for the same scale. This matches theoretical expectations as the size of a coreset to obtain a specific approximation guarantee increases with the number of superpixels k , as shown in Chapter 3. Consequently, the approximation guarantee of the same coreset increases for decreasing k .

Figures 6.9 and 6.10 give visual comparisons of the superpixel segmentation for various scale parameters. We observe that the resulting superpixels adhere to important image boundaries for essentially all the scale parameters. It seems that superpixels deviate most strongly in very textured areas with no visible image boundary of the images like in the wolf's fur.

An extreme case is depicted in Fig. 6.11. Highly-textured regions like the tree needles lose many details when the resolution is reduced, as explained in Fig. 6.4a. In these areas, the superpixels generated on different resolutions tend to deviate strongly from each other. However, even for the human eye, identifying the actual boundaries of the tree is difficult. Using coresets with multiple color bins per channel instead of one bin when reducing the resolution may help to preserve more information but requires an additional decision rule as mentioned previously.

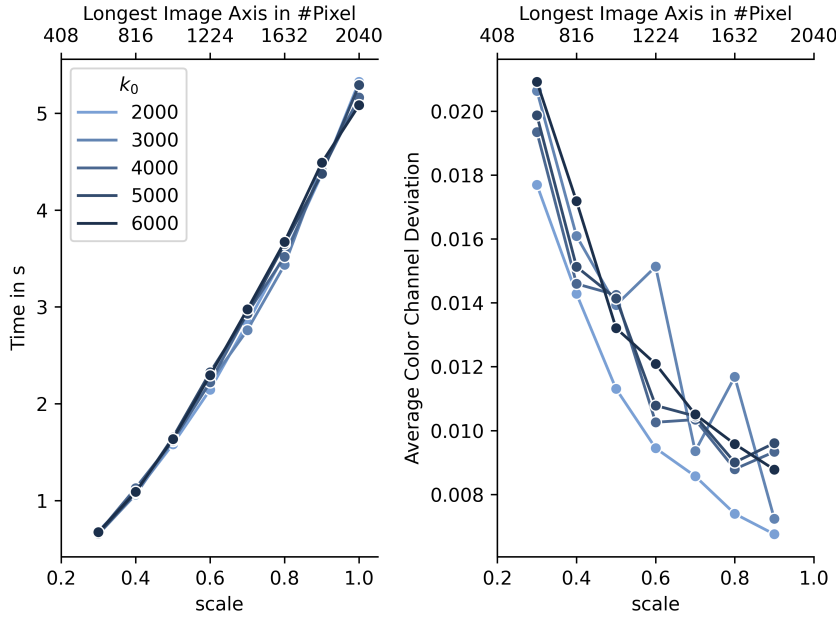


Figure 6.8: Performance of Resolution-Independent Power-SLIC for different $scale$ and k parameters.

6.7 Conclusion

We presented a new superpixel algorithm, called Power-SLIC, whose superpixels are described by cells of diagrams. Using GBPDs, the quadratic boundaries appear to provide enough flexibility to capture complex image boundaries, while providing sufficient regularization to obtain very compact superpixels. This results in a high level of robustness under varying levels of additive white Gaussian noise. Additionally, Power-SLIC matches the running times of SLIC in theory and practice. Power-SLIC's added benefit of being resolution-independent can be used to speed up superpixel generation for larger images, as a full segmentation can be computed on a low-resolution version of the image.

Although we proposed to run Power-SLIC with the standard parameters of SLIC in the general setting, we remark there is potential for further improvement if task-dependent optimal parameters are learned from training data. Extensions to other color spaces are possible and, in fact, straightforward in our framework by adapting SLIC's initialization & assignment phase.

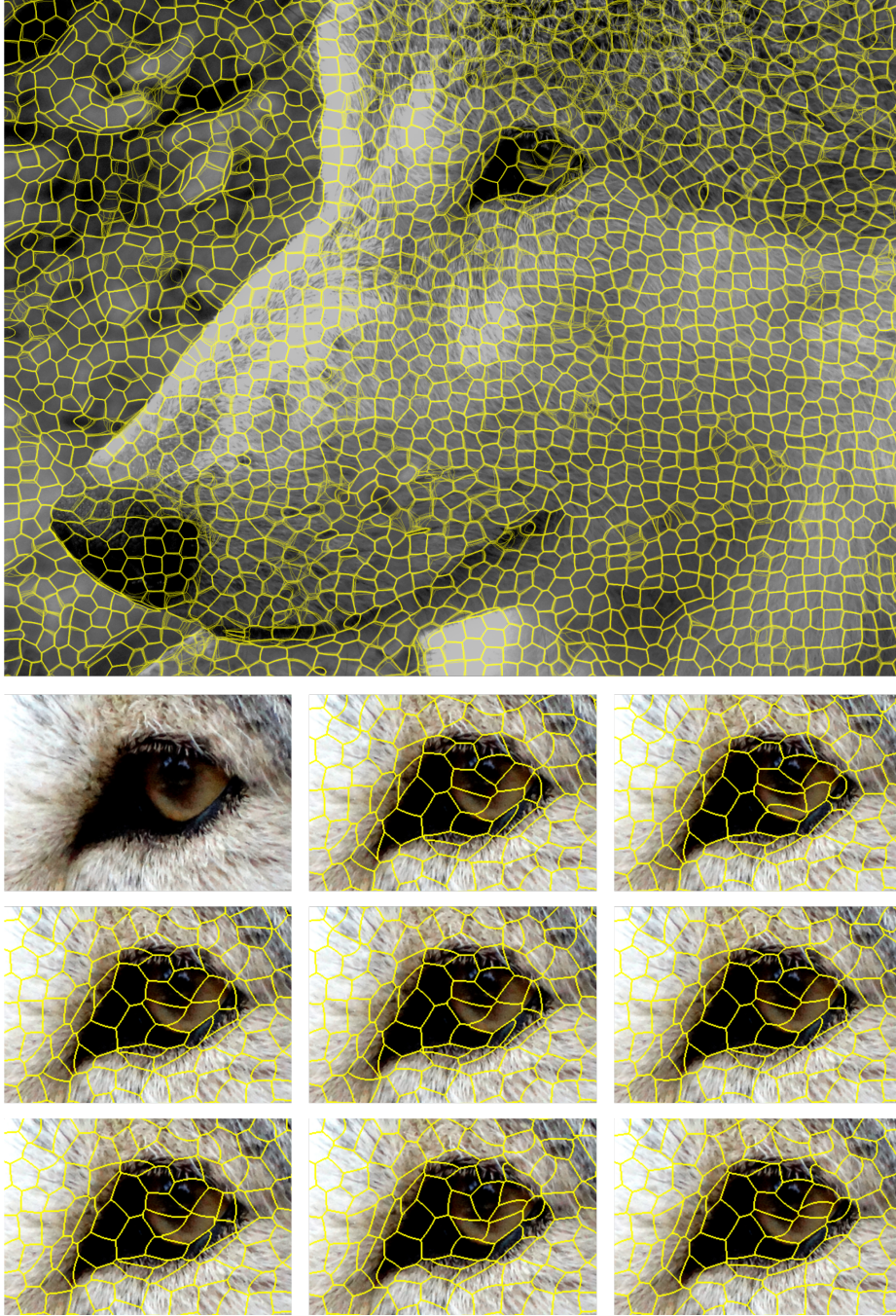


Figure 6.9: Resolution-Independent Power-SLIC for different scales (1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3) and $k = 3000$. The top image layers all segmentation results. Not that many boundaries are identical for all the scales. The grid below shows from left to right and top to bottom: Slice of the image, Resolution-Independent Power-SLIC results with scale in descending order. Boundaries are identified for all the scale parameters.

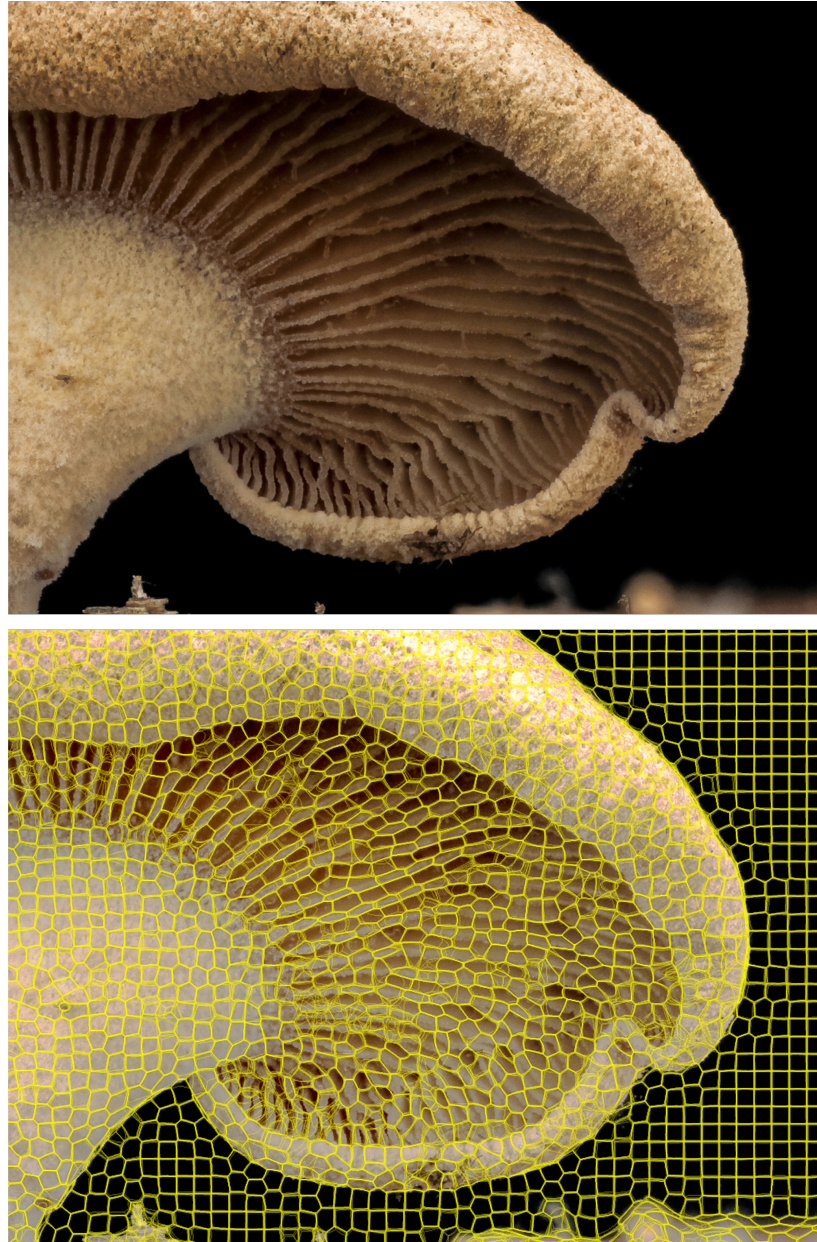


Figure 6.10: Comparing Resolution-Independent Power-SLIC for different scale parameters (1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3) and $k = 6000$ on an extract of the image. The major image boundaries are identified at almost all scale parameters and superpixels deviate mostly in regions with no image boundary to adhere to.

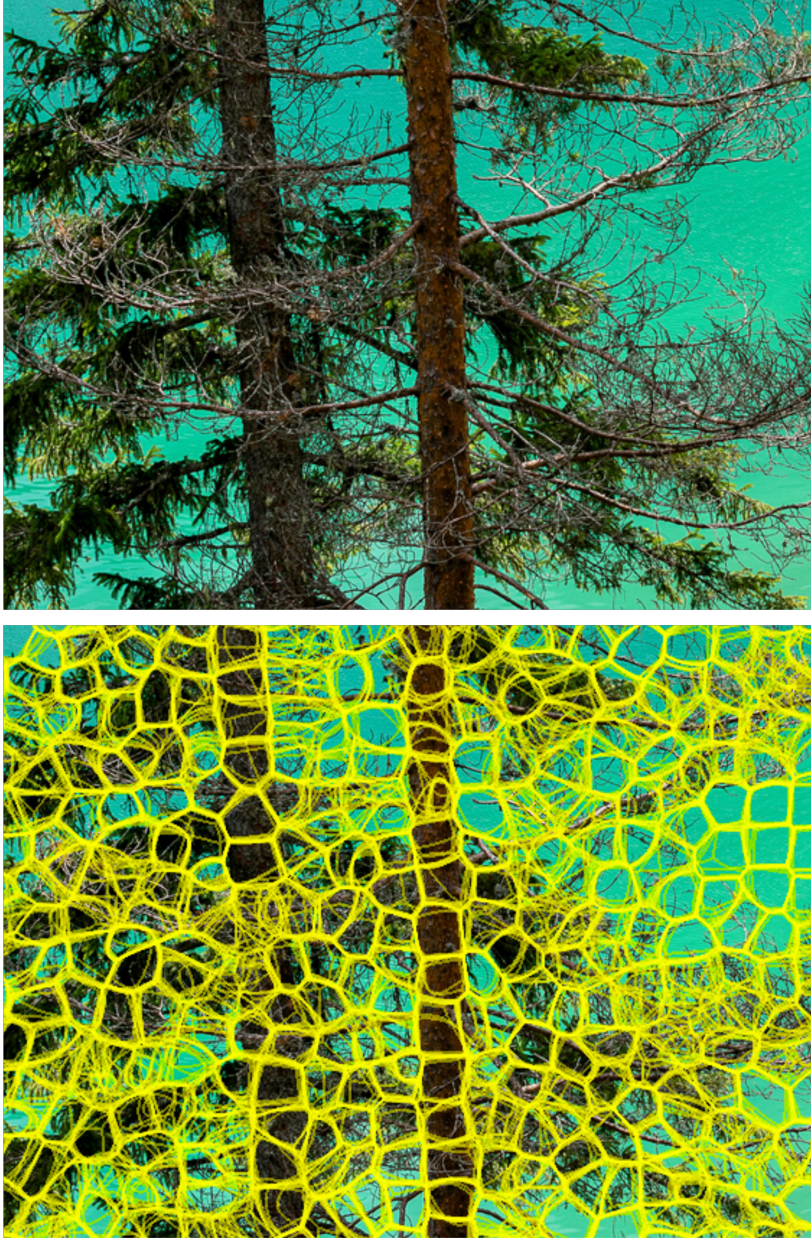


Figure 6.11: Comparing Resolution-Independent Power-SLIC for different scale parameters (1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3) and $k = 3000$ on an extract of the image. Highly-textured areas like the tree needles lose many details when the resolution is reduced. In these areas, the superpixels generated on different resolutions tend to deviate strongly from each other.

Appendix A

Inclusion of Published Core Publications

This chapter provides the published journal versions of our core publications (Chapters [4](#) and [5](#)) to fulfill the formal requirements of a cumulative dissertation. The journal approval for using the published journal versions can be found after the one-page summaries in Section [1.7](#) (not included in this published version).

Turning Grain Maps into Diagrams*

Andreas Alpers[†], Maximilian Fiedler[‡], Peter Gritzmann[‡], and Fabian Klemm[‡]

Abstract. The present paper studies mathematical models for representing, imaging, and analyzing polycrystalline materials. We introduce various techniques for converting grain maps into diagram or tessellation representations that rely on constrained clustering. In particular, we show how to significantly accelerate the computation of generalized balanced power diagrams and how to extend it to allow for optimization over all relevant parameters. A comparison of the accuracy of the proposed approaches is given based on a three-dimensional real-world data set of $339 \times 339 \times 599$ voxels.

Key words. imaging, diagram, grain map, coresets, data reduction, polycrystal

MSC codes. 94A08, 68U10, 90

DOI. 10.1137/22M1491988

1. Introduction. A challenging problem in materials science is to describe the structure of polycrystalline materials and understand the processes that govern the evolution of the structure with time, e.g., the grain growth taking place during heating; see, e.g., [21, 30, 5]. Naturally, geometric concepts play an important role, e.g., [25], and computations can use voxel-based images at different levels of resolution, in terms of two-dimensional (2D) [39, 27] or 3D *grain maps* [42]. However, the data acquisition and analysis tend to be slow, prohibiting the study of real-life dynamics. Moreover, the size of some grains may be comparable to the spatial resolution of the methods, rendering detailed visualization of shape impossible. For these reasons, sparse descriptions involving only a few parameters have been introduced. Such “compressed segmentations” are also relevant for materials modeling, as the reduced complexity allows for the analysis of much larger (and more representative) samples.

The present paper is concerned with deterministic mathematical models for converting polycrystalline images into diagram representations efficiently. We perform a “controlled analysis” based on different types of image discretization, resolutions, and aspects of data compression. Such discrete images, formally introduced in section 2, will generically be referred to as “grain scans.” (This term is more general than the standard term *grain map* from materials science.) The main goal of our analysis is to further substantiate the potential of diagram representations with a special view, however on later standard applications for solving the inverse problem of grain mapping when only certain measurements such as centroids, volumes, or centralized second-order moments of grains are available.

* Received by the editors April 21, 2022; accepted for publication (in revised form) September 21, 2022; published electronically February 7, 2023.

<https://doi.org/10.1137/22M1491988>

[†] Mathematical Sciences, University of Liverpool, Liverpool, L69 7ZL, UK (Andreas.Alpers@liverpool.ac.uk).

[‡] Zentrum Mathematik, Technical University of Munich, Garching b. München, 85747, Germany (fiedlema@ma.tum.de, gritzmann@tum.de, klemm@ma.tum.de).

Diagrams have been applied for representations of static polycrystalline materials before. The known models, such as *Voronoi* [41, 6], *Voronoi G* [20], *Laguerre* [24, 36, 40, 3, 23], *ellipsoidal growth* [38], and *generalized balanced power diagrams (GBPD)* [1], vary in terms of their accuracy and their complexity. It has been observed in several studies, including [33, 34, 39, 37, 29], that the anisotropic power diagrams (APDs) produced by the GBPD-algorithm capture the physical principles governing the forming of polycrystals quite well. However, the high accuracy comes at the price of high computational costs, which limits the practical use of the method. In fact, all known approaches either are heuristic in nature or require a significant computational effort. Indeed, there is some need for balancing the computational effort and the desired accuracy for most applications. This problem is aggravated by a strong increase in the size of grain models both experimentally and from a material modeling point of view.

The present paper introduces new and significantly accelerated variants of the GBPD-algorithm of [1], which generate APDs which are governed by parameters associated with the location, shape characteristics, and size of the grains (see section 2 for a formal definition). Such diagrams constitute a class of large flexibility containing Voronoi and Laguerre tessellations as special cases. In particular, we use data compression techniques based on *coresets* which allow for a substantial “thinning” of the data without losing too much accuracy. In fact, we can compute optimal diagrams based on low-resolution ground set models whose number of points depends only on the number of grains and the desired accuracy. We will also introduce additional techniques resulting in yet sparser sets, altogether generating a class of new algorithms for computing diagram representations, which we will call *sparse-GBPD* or *s-GBPD*. As it turns out, suitable parameter settings yield algorithms in s-GBPD, which are much faster than the GBPD-algorithm at essentially no loss of accuracy, rendering the new algorithm a viable, practical tool of high accuracy. While s-GBPD is an “indirect” clustering method based on the knowledge of centroids, volumes, and centralized second-order moments of grains which are experimentally measurable parameters, we will also introduce a new technique, addressing the best fit problem directly, which allows us to optimize over all relevant parameters that govern the weight-balanced anisotropic diagrams.

The paper is organized as follows. Section 2 formalizes the concept of grain scans, introduces diagrams, states the main problems, and indicates our main contributions. Section 3 introduces the concept of coresets for speeding up exact methods and gives several examples of how such data compression methods can be realized in our context. In particular, we show how the original GBPD-algorithm can be turned into a computationally feasible method for practical 3D problem sizes and reasonable resolutions. Section 4 introduces a new, direct, and full-parameter best fit model. The results of our empirical study comparing the different methods on available practical grain scan data are stated in section 5. Finally, section 6 concludes with remarks that particularly address the issues of how the coreset methods can be extended to situations where instead of full grain scans only certain grain measurements are available.

2. Turning grain scans into diagrams. Adequate representations of grain scans can be valuable tools for understanding the physical principles of grain growth and the properties of the resulting material. The diagram representations studied in this paper require only a few parameters per grain and are thus much easier to process than voxel-represented grain

scans. Further, since diagram representations provide a dissection of the underlying (continuous) space into cells with disjoint interiors they can even be used to boost the scan resolution. Moreover, they exhibit geometric and combinatorial features of the grain maps (e.g., its “neighborhood structure”) that are not readily available otherwise. Hence, diagram representations may contribute to predicting and even controlling the forming of new materials.

As previous studies [1, 33, 34, 37, 29, 39] show, only a few characteristic parameters govern decisive properties of grain scans. These parameters may, in practice, be derived from measurements of existing polycrystals or demanded by new materials. Here, we take a more fundamental perspective and assume that grain scans are available at some resolution. A specific focus will lie in the accuracy of fit of the derived diagrams. Also, computations should be fast but not necessarily real-time as we may want to spend additional time improving the accuracy of the representation. Accordingly, in section 4 all parameters that govern the diagrams will be subject to optimization in order to illustrate the general fitness of diagram representations. Section 3, on the other hand, will (in accordance with available tomographic techniques) assume that certain parameters are directly available through measurements, and will focus on efficient computations of the remaining ones; see Table 1 for an overview.

In this section we will first formalize the concept of a grain scan and introduce the diagrams used throughout the paper. Then we put our studies in the subsequent sections into some broader perspective and state our main contributions.

Grain scans. Let $d \in \mathbb{N}$ be the dimension we are working in (e.g., $d = 2$ and $d = 3$ for the study of planar or 3D samples, respectively), and let $k \in \mathbb{N}$ denote the number of different grains in the given polycrystalline sample. Given a set of points $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^d$ in the sample and labels, i.e., $y_1, \dots, y_n \in [k] := \{1, \dots, k\}$, we call the labeled data set $\{(x_j, y_j) : j \in [n]\}$ a *grain scan* or *polycrystal scan*. We will refer to it as (X, Y) with the understanding that the indices of the labels in the family $Y = \{y_1, \dots, y_n\}$ correspond to the indices of the points in X .

Typically our initial samples arise from digital images, normalized to lie in the range $[0, 1]^d$, and hence exhibit a grid structure. Such images are commonly referred to as *grain maps*. In order to speed up computations, we will, however, compress the data in various ways, not all of which will sustain a grid structure.

In this notation, we set

$$G_i = \{x_j \in X : (x_j, i) \in (X, Y)\}, \quad \kappa_i = |G_i|,$$

where $|G_i|$ denotes G_i 's *cardinality*. Note that G_i can be viewed as a discretization of the i th grain in the underlying polycrystal, and κ_i is proportional to its volume.

Diagrams. Given a grain scan (X, Y) we are interested in a diagram that represents the scan as accurately as possible. In its general form, *diagrams* can be defined by functions $h_1, \dots, h_k : \mathbb{R}^d \rightarrow \mathbb{R}$. For each $i \in [k]$, the corresponding *cell* P_i of the diagram is given as

$$P_i = \{x \in \mathbb{R}^d : h_i(x) \leq h_l(x) \quad \forall l \in [k]\}.$$

The choice of the functions h_i determines the diagram type; see [9] and [19] for examples and properties of different classes of diagrams. The classical *Voronoi diagram* of k distinct points $s_1, \dots, s_k$ in $\mathbb{R}^d$, for instance, is obtained by employing the functions $h_i(x) = \|x - s_i\|_2$,

where $\|\cdot\|_2$ denotes the Euclidean norm. In the following we use three types of parameters to control the properties of the cells:

$\mathcal{A}$: k positive definite symmetric matrices $A_1, \dots, A_k \in \mathbb{R}^{d \times d}$, collected in the family $\mathcal{A} = \{A_1, \dots, A_k\}$. In the special case that each A_i is the unit matrix E_d of $\mathbb{R}^{d \times d}$ the family $\mathcal{A}$ will be denoted by $\mathcal{E}$;

S : k different *sites* $s_1, \dots, s_k \in \mathbb{R}^d$, collected in the set $S = \{s_1, \dots, s_k\}$;

Γ : k *sizes* $\gamma_1, \dots, \gamma_k \in \mathbb{R}$, collected in the family $\Gamma = \{\gamma_1, \dots, \gamma_k\}$.

The sites specify the general positions of the grains while the matrices in $\mathcal{A}$ describe characteristics of their shapes. In fact, each grain is equipped with its own *ellipsoidal norm* $\|\cdot\|_{A_i}$ given through A_i by

$$\|x\|_{A_i} := \sqrt{x^\top A_i x} \quad (x \in \mathbb{R}^d).$$

Finally, the sizes γ_i can be used to control the grain volume.

The APD

$$\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma) = \{P_1, \dots, P_k\}$$

is then the diagram specified by the functions

$$h_i(x) = \|x - s_i\|_{A_i}^2 + \gamma_i \quad (i \in [k]).$$

Let us point out that $\mathcal{P}$ constitutes a decomposition of $\mathbb{R}^d$ and, in particular, two different cells do not share interior points, i.e., $\text{int}(P_i) \cap \text{int}(P_\ell) = \emptyset$ for $i \neq \ell$; see [9] for a proof (even for more general diagrams).

Clearly, APDs generalize the *Voronoi diagrams* (where $A_i = E_d$ and $\gamma_i = 0$ for all $i \in [k]$) and also the *power diagrams* (where $\mathcal{A} = \mathcal{E}$) whose cells are polyhedra. While the Euclidean norm $\|\cdot\|_{E_d} = \|\cdot\|_2$ corresponds to the *isotropic* case, the more general ellipsoidal norms allow us to model also *anisotropic* growth.

Representation. Given a polycrystalline scan, we are interested in an APD which *represents* the scan, i.e., ideally $G_i = X \cap P_i$ for all $i \in [k]$. Of course, in general, we cannot expect a perfect fit. Hence, we resort to some notion of *best fit*. Let us point out that, throughout the paper, we use the term “represent” only in a colloquial sense. Whether even a best fit APD is in practice regarded as “representing a grain map” can only be assessed with a view toward the context of a specific application.

In order to define the accuracy of an APD $\mathcal{P} = \{P_1, \dots, P_k\}$, we consider the associated *classification function*

$$f_{\mathcal{P}} : \mathbb{R}^d \rightarrow \{0, \dots, k\}, \quad x \mapsto \begin{cases} i & \text{if } x \in \text{int}(P_i); \\ 0 & \text{otherwise.} \end{cases}$$

The function $f_{\mathcal{P}}$ maps a point $x \in X$ either to the index of its unique containing cell or to 0 if x lies on the boundary of more than one cell. Using such functions, we can formulate the problem of best-representing polycrystal scans as finding a diagram $\mathcal{P}$ such that $f_{\mathcal{P}}$ minimizes the number of misclassified data points. More precisely, we want to solve the following problem.

APD: Given $n \in \mathbb{N}$ and an n -element labeled data set (X, Y) , compute, for $i \in [k]$, positive definite symmetric matrices A_i , sites s_i , and sizes γ_i such that the *classification error*

$$|\{j \in [n] : f_{\mathcal{P}}(x_j) \neq y_j\}|$$

of the corresponding diagram $\mathcal{P}$ is minimized over all feasible choices of $(\mathcal{A}, S, \Gamma)$. Moreover, return this classification error.

Note that points on the boundary of cells are all mapped to 0 and, hence, regarded as misclassified.

For **APD**, all three characteristics $\mathcal{A}$, S , and Γ are optimized and constitute the output of the problem, together with the classification error. As this reduces the computational effort, it is often useful to specify some of these characteristics beforehand and optimize the others. Fixing characteristics will be denoted by adding them in brackets. For instance, $\text{APD}(\mathcal{A}, S)$ refers to the situation considered in [1] that both $\mathcal{A}$ and S are given and the optimization involves only Γ . In effect, $\text{APD}(\mathcal{A}, S)$ is hence the problem to compute Γ^* for which, among all $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$,

$$\min_{\Gamma} |\{j \in [n] : f_{\mathcal{P}}(x_j) \neq y_j\}|$$

is attained. Even more extreme, $\text{APD}(\mathcal{A}, S, \Gamma)$ indicates that all characteristics are given, therefore the APD $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$ is completely determined, and only the classification error $|\{j \in [n] : f_{\mathcal{P}}(x_j) \neq y_j\}|$ evoked by $\mathcal{P}$ is to be computed.

Optimization. Since both d and k are fixed, it follows from arguments given in [8] (see also [15]) that **APD** can be solved by enumerating all relevant APDs in polynomial time. Their number grows, however, exponentially in k and d . Hence, such an enumerative approach is computationally infeasible for virtually all practically relevant instances. The known APX-hardness results of [7] and [15] for the case that k is part of the input indicate that there is not much hope to find much better algorithms for solving **APD** exactly. Accordingly, various heuristics suggest to fix *all* governing characteristics; see [41, 6, 20, 24, 39, 36, 40, 3, 23, 38, 2]. The sites S are generally chosen as the measured centroids

$$c(G_i) = \frac{1}{\kappa_i} \sum_{x_j \in G_i} x_j$$

of the grains. To model the shape of the grains, [24] uses Euclidean norms, i.e., $\mathcal{A} = \mathcal{E}$, while, following [1], the heuristics of [39] use the inverse of the covariance matrices, i.e.,

$$A_i = (\Sigma(G_i))^{-1} = \left(\frac{1}{\kappa_i} (x_j - c(G_i) : x_j \in G_i) (x_j - c(G_i) : x_j \in G_i)^T \right)^{-1}$$

for $i \in [k]$. (It can be assumed that the matrices $\Sigma(G_i)$ have full rank d since grains that are degenerated to volume 0 do not matter.) Further, the sizes Γ are typically estimated using

the radius of the (Euclidean or ellipsoidal) ball whose volume coincides with the measured grain volume; see [2] for an analysis in terms of discriminant analysis.

As these heuristics abstain from any optimization whatsoever, they are computationally fast but generally result in a higher misclassification error. Hence, [1] suggested a different method, called the GBPD-algorithm, that allows us to optimize over Γ . The diagrams are specified through the dual variables of a linear program that addresses a constrained clustering problem; see section 3 for details. As even linear programming is often too time-consuming for real-world instances of the desired size, we will, in particular, accelerate this method using data compression without any significant decrease of accuracy by employing *coresets*, which will be introduced in subsection 3.2.

It is particularly important that this method does not require the knowledge of the grain scan. It suffices to provide estimates for $\mathcal{A}$, S and the grain volumes, which, in practice, are often much easier to obtain from measurements. In section 4 we will, nevertheless, also provide a quite efficient direct model for optimizing over all three types of parameters, $\mathcal{A}$, S , and Γ . This will allow for a better assessment of the quality of the solutions provided by the faster and practically much easier method based on constrained clustering.

Contributions. In the following three sections, we will

- develop sparse variants of the GBPD-algorithm of [1], called s-GBPD, which significantly improve running times,
- introduce a new model for APD, referred to as DiLPM, which facilitates the optimization over all characteristic parameters, and
- state the results of a computational study on real-world data to indicate how the different algorithms perform in terms of computation times and accuracy.

The sparse variants s-GBPD will be based on the notion of “coresets” and other data compression techniques while DiLPM uses a lifting method to obtain a “direct linear programming model” which can be viewed as a support vector machine approach for machine learning classification. Table 1 provides an overview.

Table 1

Contributions: New algorithms, corresponding choice of characteristics, and computational study. The entry “covariances” stands for the choices $A_i = (\Sigma(G_i))^{-1}$, where $\Sigma(G_i)$ is the covariance matrix of the i th grain.

Contributions	Governing characteristics			Section
	$\mathcal{A}$	S	Γ	
s-GBPD	covariances	centroids	optimized	3
DiLPM	optimized	optimized	optimized	4
Evaluation				5

3. Diagrams on sparse image supports. In this section, we assume that the matrices $\mathcal{A}$ and the sites S are given beforehand (e.g., chosen as inverse covariance matrices and centroids of the grains from the grain scan, respectively), and only the sizes Γ are subject to optimization. As, however, even $\text{APD}(\mathcal{A}, S)$ is computationally difficult, [1] introduced the *linear programming* algorithm GBPD. The computations can be viewed as a dual method

that relies on a relation between diagrams and weight-constrained anisotropic clustering. As shown by experimental computations in [1], and later supported independently by [33, 34, 37], GBPD is highly accurate and the underlying model seems to capture the structure of polycrystals quite well. The computational cost, however, severely limits its practical use for 3D data sets of high resolution. Here we will show how to significantly accelerate the method, leading to a new class of algorithms, which we call sparse-GBPD or s-GBPD. In particular, we will show that the clustering can be performed at a rather low resolution, essentially without losing accuracy, and this will lead to significant speed-ups.

3.1. Diagrams from weight-constrained clustering. Let us begin by briefly describing the relation between diagrams and clustering used in [1] which, as we will see, is based on linear programming duality. For more details, we refer the reader to [9], the handbook article [19], and the papers quoted therein.

As before, let (X, Y) be a given grain scan, and let $\mathcal{A}$ and S be the given matrices and sites. Further, we collect the measured cardinalities $\kappa_1, \dots, \kappa_k$, which, together with the resolution, approximate the grain volumes in the family $K = \{\kappa_1, \dots, \kappa_k\}$.

While the points of the original grain scan are unweighted, the data compression techniques of subsection 3.2 will use point weights, and hence the optimization needs to be performed on weighted point sets. Therefore, we will assume that each data point x_j carries a weight $\omega_j \in (0, \infty)$. Of course, the original unweighted case is included by simply assigning the weight 1 to each point of the grain scan. Collecting all weights in the family $\Omega = \{\omega_1, \dots, \omega_n\}$, the weighted data set will then be denoted by (X, Ω) . By convention, all parameters corresponding to the j th point will, again, be indexed by j .

A *clustering* C of (X, Ω) is a vector

$$C = (C_1, \dots, C_k) = (\xi_{11}, \dots, \xi_{1n}, \dots, \xi_{k1}, \dots, \xi_{kn})^\top$$

whose components ξ_{ij} specify the fraction of x_j that is assigned to the i th cluster $C_i = (\xi_{i1}, \dots, \xi_{in})^\top$. If all ξ_{ij} are in $\{0, 1\}$, i.e., each point is assigned to precisely one cluster, the clustering will be called *integer*.

In terms of formal constraints, any clustering satisfies the *assignment conditions*

$$\xi_{ij} \geq 0 \quad (i \in [k], j \in [n]) \quad \text{and} \quad \sum_{i=1}^k \xi_{ij} = 1 \quad (j \in [n]).$$

Further, the *weight* $\omega(C_i)$ of the cluster C_i is given by

$$\omega(C_i) = \sum_{j=1}^n \xi_{ij} \omega_j.$$

Note that the given grain scan (X, Y) naturally defines the integer clustering $G = (G_1, \dots, G_k)$ of the (discretized) grains. It is formally specified by the components $\xi_{ij}^G \in \{0, 1\}$ with $\xi_{ij}^G = 1$ if and only if $x_j \in G_i$. As all points carry the weight 1, cluster G_i has weight $\omega(G_i) = \kappa_i$, $i \in [k]$. It is, of course, this *grain scan clustering* that we want to represent by a suitable

diagram $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$. Accordingly, a clustering C and a diagram $\mathcal{P} = (P_1, \dots, P_k)$ are called *compatible* if, for all $i \in [k]$,

$$P_i \supset \text{supp}(C_i) = \{x_j \in X : \xi_{ij} > 0\}.$$

In general, we cannot expect the grain scan clustering to be compatible with any APD. Hence, we resort to the problem of determining a clustering C^* that is “close” to G and which admits a compatible diagram.

A natural and simple condition is to require that the clusters of C^* and G have similar weight, i.e., for given $\epsilon_i \in (0, \infty)$ the *weight constraints*

$$(1 - \epsilon_i)\kappa_i \leq \omega(C_i) \leq (1 + \epsilon_i)\kappa_i \quad (i \in [k])$$

should hold. In practice, the error terms ϵ_i may reflect the available resolution of the image (or the known or expected accuracy of the measurements).

Our overall rationale will, however, be as follows. First, we evoke linear programming duality to relate optimal constrained clusterings to APDs, as already suggested in [1]. Then, we apply this model to a rather small weighted point set which is obtained through the data compression techniques described in subsection 3.2. Finally, we use the so obtained diagram to assign the points of the original data set to clusters. This yields an integer clustering, which we compare with the grain scan clustering to compute its classification error.

In the following, we will actually enforce the strict constraints $\omega(C_i) = \kappa_i$ on the compressed data set for all $i \in [k]$. The set of all such *weight-constrained clusterings* will be denoted by $\mathcal{C}_K(X, \Omega)$. Further, the given shape parameters $\mathcal{A}$ and S are incorporated in the *cost function*

$$\text{cost}_{\mathcal{A}, S}(X, \Omega, C) = \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{A_i}^2,$$

which is to be minimized. The idea behind this choice of objective function is to enforce some kind of “minimum energy principle” by rewarding the assignments of points to the closest site with respect to the ellipsoidal norms.

Altogether, we are considering the problem *weight-constrained anisotropic assignment*:

WCAA: Given k, X, Ω, K, S , find $C^* \in \mathcal{C}_K(X, \Omega)$ that minimizes $\text{cost}_{\mathcal{A}, S}(X, \Omega, C)$.

Note that the grain scan clustering G is feasible for the original unweighted data set. In general, the simple condition

$$\sum_{i=1}^k \kappa_i = \sum_{j=1}^n \omega_j$$

guarantees that $\mathcal{C}_K(X, \Omega) \neq \emptyset$, and we will, in the following, assume that this is the case. Hence, a finite minimum always exists for WCAA; it will be denoted by $\text{cost}_{\mathcal{A}, S}(X, \Omega)$, i.e.,

$$\text{cost}_{\mathcal{A}, S}(X, \Omega) = \min_{C \in \mathcal{C}_K(X, \Omega)} \text{cost}_{\mathcal{A}, S}(X, \Omega, C).$$

Next, note that WCAA can be formulated as the following linear program in the variables ξ_{ij} :

$$\begin{aligned}
 & \min_{(\xi_{ij})} \sum_{i=1}^k \sum_{j=1}^n \xi_{ij} \omega_j \|x_j - s_i\|_{A_i}^2, \\
 \text{(P)} \quad & \sum_{i=1}^k \xi_{ij} = 1 \quad (j \in [n]), \\
 & \sum_{j=1}^n \xi_{ij} \omega_j = \kappa_i \quad (i \in [k]), \\
 & \xi_{ij} \geq 0 \quad (i \in [k], j \in [n]).
 \end{aligned}$$

Let us remark that in the case that all weights coincide and, in particular, if all are 1, the coefficient matrix of (P) is *totally unimodular*, hence each basic feasible solution is integer; see, e.g., [32] for background information. The dual linear program

$$\begin{aligned}
 & \max_{(\eta_j, \gamma_i)} \sum_{j=1}^n \omega_j \eta_j - \sum_{i=1}^k \kappa_i \gamma_i, \\
 \text{(D)} \quad & \eta_j \leq \|x_j - s_i\|_{A_i}^2 + \gamma_i \quad (i \in [k], j \in [n]),
 \end{aligned}$$

in the variables η_j and γ_i encodes proximity conditions involving the diagram functions based on $\|x - s_i\|_{A_i}^2$. Recall that, due to duality, the optima in (P) and (D) coincide. Further, we have the following theorem.

Theorem 1 (special case of [9, Thm. 1]). *Let $(k, X, \Omega, K, \mathcal{A}, S)$ be an instance of WCAA, and $C \in \mathcal{C}_K(X, \Omega)$. Then C is an optimizer of the linear program (P) if and only if there exist size parameters Γ such that C and the diagram $\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma)$ are compatible.*

According to Theorem 1, an optimal clustering C^* results in an optimal diagram $\mathcal{P}^*$. Hence, we can apply any standard linear programming algorithm to compute optimal sizes. In section 5 we actually use the dual simplex algorithm. Note, however, that optimality is measured here with respect to the objective functions of (P) and (D), and it is not clear in the first place how the cost of a clustering translates into the classification error which is to be minimized in $\text{APD}(\mathcal{A}, S)$. See [17] for related stability and instability results. Hence, in effect, the clustering-based algorithms address $\text{APD}(\mathcal{A}, S)$ only indirectly.

3.2. Data compression for accelerating the linear program. If the linear program (P) is based on the original (unweighted) ground set X of the image, say, on the points of $1/\nu \cdot \mathbb{Z}^d$ in $[0, 1]^d$ for some $\nu \in \mathbb{N}$, then it involves $k \cdot (\nu + 1)^d$ variables. If, for instance, in three dimensions, $\nu = 9999$ and $k = 50$, (P) has already $5 \cdot 10^{13}$ variables. Of course, as one would expect that $\xi_{ij} = 0$ whenever s_i is not among the first few closest sites to x_j , preprocessing would allow us to reduce the influence of k to a manageable size. It is, however, $|X|$ which affects the number of variables the most.

Image supports and coresets. In the following, we will reduce the number of variables considerably while retaining almost the initial accuracy. Our approach is based on the observation that the point set (X, Ω) is only a means for obtaining the desired diagram but not

important in its own right. In fact, once we obtain the diagram $\mathcal{P}$, we can easily generate an image at any desired resolution from it. Hence, we can, in principle, use *any* nonempty finite subset $X' \subset [0, 1]^d$ for computing diagrams through clustering—as long as we associate suitable weights to the points to accommodate the weight constraints. We will call such weighted sets (X', Ω') *image supports*.

Naturally, some such sets will be better suited than others, and we will speak of *coresets* $(\tilde{X}, \tilde{\Omega})$ if these sets can be derived from (X, Ω) in such a way that the relative deviation between $\text{cost}_{\mathcal{A},S}(X, \Omega)$ and $\text{cost}_{\mathcal{A},S}(\tilde{X}, \tilde{\Omega})$ can always be suitably bounded in terms of a given precision. Here, we are interested in choosing the image supports (X', Ω') as part of the “experimental setup,” i.e., prior to scanning or taking characteristic measurements of the polycrystal. Thus, this setting requires the condition for *all* sets of k sites.

More precisely, let $(\tilde{X}, \tilde{\Omega})$ be a weighted data set, $\epsilon \in (0, 1/2]$, $\delta \in [1, \infty)$. Then, following [16], $(\tilde{X}, \tilde{\Omega})$ is an (ϵ, δ) -coreset for (k, X, Ω, K) if there exists a mapping $g : \mathcal{C}_K(k, \tilde{X}, \tilde{\Omega}) \rightarrow \mathcal{C}_K(X, \Omega)$, called *extension*, and real constants Δ^+, Δ^- , referred to as Δ -terms or *offsets*, with $0 \leq \Delta^+ \leq \delta \cdot \Delta^-$ such that the following two conditions hold for all sets of k sites and all clusterings $\tilde{C} \in \mathcal{C}_K(\tilde{X}, \tilde{\Omega})$:

- (a) $(1 - \epsilon)\text{cost}_{\mathcal{A},S}(X, \Omega, g(\tilde{C})) \leq \text{cost}_{\mathcal{A},S}(\tilde{X}, \tilde{\Omega}, \tilde{C}) + \Delta^+,$
- (b) $\text{cost}_{\mathcal{A},S}(\tilde{X}, \tilde{\Omega}) + \Delta^- \leq (1 + \epsilon)\text{cost}_{\mathcal{A},S}(X, \Omega).$

If $\delta = 1$ we will simply speak of an ϵ -coreset.

Let us point out that other notions of coresets have been introduced for different problems; see, e.g., [22, 13, 12, 14, 35, 4]. The definition used here is specifically customized for weight-constrained anisotropic clusterings and assignments. It guarantees that approximate solutions of the constrained clustering problems on coresets can be turned into approximate solutions on the original data set; see [16, Thm. 3.5].

In order to practically use coresets for constrained clustering, they need to be small and the extension g must be quickly computable. While, in our context, the coreset constructions (given below) are important, the extension does not play any role, and the coreset conditions are relevant for theoretical reasons only, as they guarantee any desired accuracy at least with respect to the clustering cost. In fact, clustering is not used here in its own right but only as a method for computing diagrams via linear programming duality. Accordingly, we will determine the assignment of the original data points directly from the diagram computed on the coreset data and compare the obtained clustering C with the grain scan clustering G to quantify the classification error; see section 5. In particular, the weights of the so obtained clusters will, in general, deviate from the specified target weights κ_i .

In the following, we will first present two different coreset constructions which rely, in addition to $\mathcal{A}$ and S , only on the measurements $\kappa_1, \dots, \kappa_k$ and, hence, can be used in a much more general practical context. Subsequently, we will then, however, also consider an image support based on the grain scan clustering $G = (G_1, \dots, G_k)$ to show how far the concept can be pushed in principle.

Pencil coresets. The following result shows that coresets exist and that they can be profitably used to accelerate GBPD computations. Let $\lambda^-(\mathcal{A})$ and $\lambda^+(\mathcal{A})$ denote the smallest and largest eigenvalues of all matrices in $\mathcal{A}$, respectively.

Theorem 2 (special case of [16, Thm. 2.3]). *For any instance of WCAA and for any $\epsilon \in (0, 1/2]$ and*

$$\delta \geq \frac{\lambda^+(\mathcal{A})}{\lambda^-(\mathcal{A})},$$

there is an (ϵ, δ) -coreset $\tilde{X}$ of size

$$|\tilde{X}| \in \mathcal{O}\left(\frac{k^2}{\epsilon^{d+1}}\right).$$

The theorem shows that it suffices to use relatively small point sets for clustering whose “sparseness” is independent of the initial point set X and depends only on k and ϵ . More importantly, Theorem 2 is constructive. As the known constructions “live” on pencils of lines in $\mathbb{R}^d$ issuing from a certain set of points, they are called *pencil coresets*. We will report on our implementation of such pencil coresets for different error parameters in section 5. Here we will briefly indicate their structure, which is based on a design by [22] for unconstrained clustering. Its generalization to weight-constrained anisotropic clusterings and the improved bound are due to [16].

For technical simplicity, we will focus on the Euclidean case and, as this is the relevant situation here, assume that the “source points” coincide with given sites $s_1, \dots, s_k$. For a visualization in two dimensions, see Figure 2. The lines of the i th pencil are chosen in such a way that their points of intersection with the unit sphere centered at the site s_i are “evenly distributed.” Note that this is closely related to coverings of the unit sphere by congruent spherical caps. See, e.g., [26] for general results on ϵ -nets and [11] for sparse coverings of the Euclidean 2-sphere by specific numbers of congruent spherical caps.

Then, we assign each point of the data set to its closest site, breaking ties arbitrarily, and project it onto a line of the corresponding pencil closest to it, again breaking ties arbitrarily. After this projection process, all points lie on lines, and in the remaining step, each line is treated independently. Note that this is possible due to the additivity of the cost function with respect to the data points.

Each such line is now partitioned into batches. Finally, all points in one batch B are replaced by the batch’s centroid $c(B)$ and assigned the sum of weights of all batch points as its weight. Of course, in our standard setting with unit weights, this process evokes a *batch error* $\sum_{x \in B} \|x - c(B)\|_A^2$.

If the lengths of the batches are chosen appropriately, this process results in the desired (ϵ, δ) -coreset. See [16] for the details of the proof.

Resolution coresets. While the construction in the previous section works for general data sets (X, Ω) , the pencils ignore the grid structure of the given grain scan. In particular, the “density” of the image support varies with the distance to the sites. For voxel-based images, it seems more natural and, actually, interesting in its own right to consider image supports that pertain to a (coarser) grid structure and which can still be interpreted in terms of the image resolution.

In the following, we will briefly introduce the concept of resolution coresets. For technical simplicity, we will assume that the voxels are boxes whose edge lengths are powers of $1/2$. Our exposition follows [18].

For $i \in [d]$, and $\rho = (\rho_1, \dots, \rho_d) \in \mathbb{N}^d$ we consider the uniform partitioning of the interval $[0, 1]$ on the i th coordinate axis into 2^{ρ_i} intervals of equal length. Then each voxel is the Cartesian product of one of the intervals on each axis. More formally, with

$$x_{j_1, \dots, j_d}^\rho = \begin{pmatrix} \frac{1}{2^{\rho_1+1}} + \frac{j_1}{2^{\rho_1}} \\ \vdots \\ \frac{1}{2^{\rho_d+1}} + \frac{j_d}{2^{\rho_d}} \end{pmatrix},$$

the voxels are obtained as

$$x_{j_1, \dots, j_d}^\rho + \frac{1}{2^{\rho_1+1}}[-1, 1] \times \dots \times \frac{1}{2^{\rho_d+1}}[-1, 1]$$

for all choices of $j_1, \dots, j_d$ with

$$j_1 \in \{0, \dots, 2^{\rho_1} - 1\}, \dots, j_d \in \{0, \dots, 2^{\rho_d} - 1\}.$$

We collect the points $x_{j_1, \dots, j_d}^\rho$ in the data set X^ρ . Note that the points are centroids of the voxels and constitute the original data set, i.e., $X^\rho = X$. Of course, $n = |X^\rho| = 2^{\rho_1} \dots 2^{\rho_d}$. We will refer to the vector $(\rho_1, \dots, \rho_d)$ as the *resolution* of the image. Note that if all ρ_i coincide with some given $\bar{\rho} \in \mathbb{N}$, then

$$X^\rho = \frac{1}{2^{\bar{\rho}+1}} \mathbf{1} + \left(\left(\frac{1}{2^{\bar{\rho}}} \mathbb{Z}^d \right) \cap [0, 1)^d \right),$$

where $\mathbf{1}$ is the all-ones vector $(1, \dots, 1)^\top$, i.e., the point set consists of all points in $[0, 1]^d$ of a translate of the integer lattice of side $1/2^{\bar{\rho}}$. Naturally, a lower resolution is obtained by decreasing some or all ρ_i , resulting in sparser image supports. The question is how small we can choose a resolution $\tau \in \mathbb{N}^d$ with $\tau_i \leq \rho_i$ such that X^τ , together with appropriate weights, is a coresset.

As it turns out, it suffices to consider a number of points on each axis which depends only on k and ϵ .

Theorem 3 (see [18, Thm. 1]). *Let $\rho \in \mathbb{N}^d$, (k, X^ρ, Ω, K) be an instance of weight-constrained anisotropic clustering, and $\epsilon \in (0, 1/2]$. Then, there is a $\bar{\tau} \in \mathbb{N}$ with*

$$\bar{\tau} \leq \frac{2^{8/3} k}{\epsilon^{2/3}}$$

such that, for $\tau = (\bar{\tau}, \dots, \bar{\tau}) \in \mathbb{N}^d$, the set X^τ together with appropriate weights is an ϵ -coreset of size

$$|X^\tau| \leq \left(\frac{2^{8/3} k}{\epsilon^{2/3}} \right)^d.$$

Such a set X^τ is referred to as a *resolution coresset*. Theorem 3 allows us to reduce the resolution of the scan to a size that is independent of the original resolution and only depends on the number of grains and the desired accuracy (measured in terms of the cost of the

produced clustering). Hence, we can use grain scans of relatively low resolution, solve (P) for the correspondingly small data set, and then obtain a diagram representation via (D). Recall that we can, ultimately, use the diagram to produce a grain image of any (arbitrarily high) desired resolution.

Let us briefly address the difference between the bounds in Theorems 2 and 3 to place the results into perspective. For example, when $k = 50$ and $\epsilon = 0.01$, not taking constants (hidden, in Theorem 2, in the Landau notation) into account, the coreset size guaranteed by Theorems 2 and 3 in three dimensions is $2.5 \cdot 10^{11}$ and $1.25 \cdot 10^9$, respectively. Note, however, that Theorems 2 and 3 give *worst-case upper* bounds which are not known (and not even conjectured) to be best possible. As we will see in section 5, the image supports can indeed be reduced much more radically in our practical computations without sacrificing much of the accuracy.

Let us repeat that while the coreset property bounds the error in clustering costs, it does not directly provide a theoretical bound for the classification error of the corresponding diagram for $\text{APD}(\mathcal{A}, S)$. The practical results in section 5 indicate, however, that the “indirect” quality guarantee which comes with the coreset property also leads to a favorable classification behavior.

Grain scan-based interior removal. The previously constructed pencil and resolution coresets make use of only certain characteristics of grain maps. They can, of course, be derived from a given grain scan (X, Y) , but, in practice, they are often much more easily available through direct tomographic measurements. While this is an important feature for practical applications, it is, of course, natural to ask how much is actually lost by not using the full grain scan information. More precisely, using $G = (G_1, \dots, G_k)$, how much further can the size of image supports be reduced while still constituting coresets, i.e., while still facilitating quite accurate clusterings in terms of their costs? Note that this is not just a theoretical issue but also highly relevant whenever given grain scan representations need to be turned into diagram representations to facilitate image postprocessing and analysis tasks.

Suppose that, under ideal conditions, the grain growth process indeed results in a diagram. Then we may assume that the given grain scan clustering G is close, up to minor imperfections, to an optimal clustering C^* .

Before continuing, let us mention that such an assumption is in line with several common stochastic grain growth models, including Poisson–Voronoi and Laguerre [28, 10]. These methods assume growth from a set of point germs $\mu_1, \dots, \mu_k$ with a normal distribution $\mathcal{N}(\mu_i, \Sigma_i)$ modeling the probability of points being contained in the i th grain. In these cases, the shape parameters are chosen according to maximum likelihood resulting in $s_i = \mu_i$ and $A_i = \Sigma_i^{-1}$. See, e.g., [3, 2] for more information.

Under such a “closeness” assumption it seems reasonable to discard all points that are sufficiently far away from the “grain boundaries” of G . More formally, we choose a suitable metric $\mathfrak{d}$ and parameters $\delta_i \in (0, \infty)$ and define the δ_i -interior of G_i by

$$G_i(\delta_i) = \{x \in G_i : \mathfrak{d}(x, x') > \delta_i \ \forall x' \notin G_i\}$$

In the grid-based situation of the previous paragraph it seems natural to use for $\|\cdot\|_1$ a variant of the 1-norm normalized by the different resolutions on the axes, i.e., for $z = (\zeta_1, \dots, \zeta_d)^\top$

$$\|z\|_1 = \sum_{i=1}^d 2^{\rho_i} \zeta_i.$$

The corresponding norm-induced metric then counts the “grid distance” which is the length of a shortest path in the underlying grid graph. In section 5, we will use this metric and refer to it as *grid graph distance*. Note that (for any standard metric) $G_i(\delta_i)$ can be computed efficiently.

The parameters δ_i control the size of the interior, and a larger δ_i will result in a smaller δ_i -interior and, hence, in a larger image support but potentially also higher accuracy. The specific choices of the δ_i reflect the initial confidence that G is already close to an optimal clustering. The deviation from G of the grain scan induced by the computed diagram $\mathcal{P}$ may be viewed as an indicator of the state of the process, i.e., of how far it is from an energy-minimal state which is indeed characterized by the underlying model.

For the computations in section 5, we employ image supports obtained by coreset techniques combined with interior removal. Any specific algorithm of the so obtained class of methods will be referred to as *sparse-GBPD* or, shorter, *s-GBPD*. As we will show, suitable such choices lead to substantial accelerations of diagram computations on real-world data.

4. A direct model for APD. The algorithms of the previous section are based on estimates for the matrices and sites and treat $\text{APD}(\mathcal{A}, S)$ indirectly via clustering. We will now propose an optimization model which addresses APD directly. As it optimizes over all three characteristic parameters $\mathcal{A}$, S , and Γ , the larger set of diagrams to choose from will allow for a better fit of (X, Y) . The higher accuracy comes, however, at the price of computational costs which are substantially higher than those of the methods described in section 3. Nonetheless, combining the model with our previous point reduction techniques enables at least its computation for the real-world data set of section 5.

This section has, indeed, a somewhat different focus: On the one hand, it provides a “benchmark method” for a better assessment of the quality of fit produced by other methods for appropriately chosen test cases. On the other hand, it may be helpful for “quantifying the physical state” of a prematurely terminated growth process in terms of the achieved fit. As expected and exemplified in section 5, the new algorithm will only be practically useful if computation time is not the main concern.

The technique that we will now introduce is based on the paradigm of *support vector machines* from data analysis and requires knowledge of (a relevant part of) the grain scan (X, Y) . See, e.g., [31] for additional background information on machine learning.

Let again $G = (G_1, \dots, G_k)$ be the grain scan clustering. If there exists a diagram $\mathcal{P}(\mathcal{A}, S, \Gamma)$, compatible with G , its defining functions $h_i(x) = \|x - s_i\|_{\mathcal{A}_i}^2 + \gamma_i$ will separate the points of the clusters, i.e., for $j \in [n]$ and $i, \ell \in [k]$ with $\ell \neq i$,

$$h_i(x_j) \leq h_\ell(x_j) \quad \text{for each } x_j \in G_i.$$

Since, in general, real-world scans will not admit a perfectly fitting diagram, it is natural to relax these conditions and resort to the optimization problem

$$\begin{aligned} \min_{(\zeta_j)} \quad & \sum_{j=1}^n \zeta_j, \\ h_i(x_j) \leq \quad & h_\ell(x_j) + \zeta_j \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i), \\ \zeta_j \geq \quad & 0 \quad (j \in [n]). \end{aligned}$$

Here, the nonnegative error terms ζ_j allow for potential misclassification, and the objective minimizes the accumulated errors. As, however, we can multiply all matrices A_i and sizes γ_i by a common positive factor without changing the diagram, the infimum of the objective function will always be 0, independently of the fit of the diagram. There are various possibilities to break this invariance. Our approach reflects the assumption that points near the grain centroids are already correctly classified by the grain scans and classification errors occur only near the boundaries. This will be modeled in terms of the δ_i -interior for suitably chosen $\delta_i \in (0, \infty)$ introduced in the previous section, resulting in the optimization problem

$$\begin{aligned} \min_{(\zeta_j)} \quad & \sum_{j=1}^n \zeta_j, \\ h_i(x_j) \leq \quad & h_\ell(x_j) - 1 \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i(\delta_i)), \\ h_i(x_j) \leq \quad & h_\ell(x_j) + \zeta_j \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i \setminus G_i(\delta_i)), \\ \zeta_j \geq \quad & 0 \quad (j \in [n], x_j \in G_i \setminus G_i(\delta_i)). \end{aligned}$$

In this model, all points in the δ_i -interior of G_i are separated strictly. Note that the specific choice of 1 for the “margin” is arbitrary as long as it is positive. Let us point out that, while a positive value of ζ_j in the optimum indicates that the point x_j is misclassified, the objective function does not actually *count* the number of misclassified points. In fact, using indicator variables to express the classification error would result in an integer program which can no longer be solved for appropriate sizes of grain maps within acceptable time.

The above optimization problem has nonlinear constraints as the optimization extends over the entries in A_i , s_i , and γ_i and

$$h_i(x_j) = \|x_j - s_i\|_{A_i}^2 + \gamma_i = x_j^T A_i x_j - 2s_i^T A_i x_j + s_i^T A_i s_i + \gamma_i,$$

where only the points x_j are given. We will therefore apply a lifting technique to linearize the problem by introducing new variables for each relevant expression. First, we use the notation

$$x_j = \begin{pmatrix} (x_j)_1 \\ \vdots \\ (x_j)_d \end{pmatrix}, \quad A_i = \begin{pmatrix} (A_i)_{1,1} & \dots & (A_i)_{1,d} \\ \vdots & & \vdots \\ (A_i)_{d,1} & \dots & (A_i)_{d,d} \end{pmatrix}$$

and set

$$a_i = -2A_i s_i, \quad \alpha_i = s_i^T A_i s_i + \gamma_i$$

to encode all parameters in the *extended parameter vector*

$$\mathfrak{A}_i = \left(\alpha_i, a_i^\top, (A_i)_{1,1}, 2(A_i)_{1,2}, \dots, 2(A_i)_{d-1,d}, (A_i)_{d,d} \right)^\top \in \mathbb{R}^{1+d+\frac{1}{2}d(d+1)}.$$

Note that (A_i, s_i, γ_i) can easily be recovered from $\mathfrak{A}_i$. In fact, the entries of A_i can be read off directly, then $s_i = -1/2 A_i^{-1} a_i$ and, finally, $\gamma_i = \alpha_i - s_i^\top A_i s_i$. If we now introduce, for each point x_j , its *extended coefficient vector*

$$\mathfrak{X}_j = \left(1, (x_j)_1, \dots, (x_j)_d, (x_j)_1^2, (x_j)_1(x_j)_2, \dots, (x_j)_1(x_j)_d, \dots, (x_j)_{d-1}(x_j)_d, (x_j)_d^2 \right)^\top,$$

we can write the separation conditions in terms of the linear expression

$$h_i(x_j) = \mathfrak{A}_i^\top \mathfrak{X}_j.$$

Accordingly, we will address APD via the *direct linear programming model*

$$\begin{aligned} & \min_{(\zeta_j)} \sum_{j=1}^n \zeta_j, \\ \text{(DiLPM)} \quad & \mathfrak{A}_i^\top \mathfrak{X}_j - \mathfrak{A}_\ell^\top \mathfrak{X}_j + 1 \leq 0 \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i(\delta_i)), \\ & \mathfrak{A}_i^\top \mathfrak{X}_j - \mathfrak{A}_\ell^\top \mathfrak{X}_j - \zeta_j \leq 0 \quad (j \in [n], i, \ell \in [k], \ell \neq i, x_j \in G_i \setminus G_i(\delta_i)), \\ & \zeta_j \geq 0 \quad (j \in [n], x_j \in G_i \setminus G_i(\delta_i)). \end{aligned}$$

In the following we will also use the acronym DiLPM for any algorithm that solves (DiLPM).

There is one additional issue to address. In order to derive diagram parameters from a solution $(\mathfrak{A}_1^*, \dots, \mathfrak{A}_k^*)$ of (DiLPM), we need to guarantee that the obtained matrices $A_1^*, \dots, A_k^*$ are symmetric and positive definite. Otherwise, the computations may not result in an APD. While the symmetry of the matrices is built into the model, there is, however, no precaution taken that the A_i^* produced via DiLPM are indeed positive definite.

To obtain appropriate matrices, we will use the fact that the constraints in (4.1) are invariant under the operation which replaces all entries $(A_i)_{\ell,\ell}$ in $\mathfrak{A}_i$ by $(A_i)_{\ell,\ell} + \beta$ for any arbitrary $\beta \in \mathbb{R}$ while keeping all other entries as before. In fact, we simply add and subtract the term $\beta x_j^\top x_j$ in the first two groups of constraints. Hence, neither the constraints nor the objective function values of (DiLPM) change. In effect, this operation replaces the last $\frac{1}{2}d(d+1)$ entries in $\mathfrak{A}_i$ by $\bar{A}_i = A_i + \beta E_d$.

So, suppose that not all of the resulting matrices A_i are already positive definite. Then we can simply switch over to $\bar{A}_i = A_i + \beta E_d$ for all $i \in [k]$ and some suitable $\beta \in (0, \infty)$ so that all $\bar{A}_i$ become positive definite. We use the matrices $\bar{A}_i$ now for the subsequent computation of the sites and the sizes and set

$$\bar{s}_i = -1/2 \bar{A}_i^{-1} a_i, \quad \bar{\gamma}_i = \alpha_i - \bar{s}_i^\top \bar{A}_i \bar{s}_i.$$

Then $(\bar{A}_i, \bar{s}_i, \bar{\gamma}_i)$ ($i \in [k]$) yield, indeed, a desired diagram $\mathcal{P}(\bar{\mathcal{A}}, \bar{\mathcal{S}}, \bar{\Gamma})$ with the same objective function value in (DiLPM).

Let us close by pointing out that (DiLPM) can be combined with the image supports described in subsection 3.2. Indeed, for practical computations one can largely thin out X first

and, in particular, require strict separation only for data points in certain rings $G_i(\delta_i^-) \setminus G_i(\delta_i^+)$ with $\delta_i^- < \delta_i^+$. In section 5 we will employ such accelerating techniques but see that (DiLPM) is significantly more time-consuming than (P).

5. Computational results. We will now evaluate the introduced techniques with respect to computation time and accuracy using a 3D real-world grain scan from [24]. After briefly describing the computational setup, we will first assess the achievable degree of sparseness of the image supports for s-GBPD. This allows us to choose appropriate “sparseness parameters” to compare s-GBPD and DiLPM for a range of different quality criteria. This comparison will illustrate the degree of fit produced by the former at a much lower computational cost than the latter.

5.1. Computational setup. Our computational study is based on data provided by [24]; see also the description in [1]. It was obtained by a synchrotron microtomograph experiment conducted on a metastable beta titanium alloy (Ti β 21S). The material was scanned with a volume of $240\mu\text{m} \times 240\mu\text{m} \times 420\mu\text{m}$ at a resolution resulting in a voxel edge length of $0.7\mu\text{m}$. It comprises 591 grains, 211 of which are interior to the polycrystal, i.e., they are entirely surrounded by other grains.

We hence have a grain scan (X, Y) with

$$n = |X| = 339 \times 339 \times 599 = 68,837,679 \quad \text{and} \quad k = 591.$$

In the following we refrain from normalizing X and report all results for the original physical dimension stated above. All computations are carried out in three dimensions. The images depict 2D slices parallel to the plane of the first two coordinates. One such example is shown in Figure 1.

Let us point out that the larger blue patch in the left lower corner and the nearby tiny blue patch surrounded by the green area actually belong to the same grain. While this can be seen in the 3D image, its 50th slice depicts two components. So, let us emphasize again that



Figure 1. Slice no. 50 of the data set. Neighboring grains are colored in different colors.

all subsequent 2D images are for illustration purposes only, while all reported results have been fully computed in three dimensions. This contrasts with most computations in [1] and, as a matter of fact, 3D problems of the size considered here are far beyond the capability of the original GBPD-algorithm introduced there.

The reported running times were measured on a virtual machine with 10 virtual CPUs corresponding to five Intel Xeon Gold 6148 CPU @ 2.40 GHz CPUs and 45 GB of memory (i.e., one CPU processes two virtual CPUs). For better comparison, we solved all linear programs using the dual simplex algorithm on only one virtual CPU. Let us point out, however, that the sparse models, which we show to be sufficient, also run on standard machines with less than 16 GB of memory.

5.2. s-GBPD: Sparseness versus fit. The coresets constructions in Theorems 2 and 3 allow us to work with sample sizes that only depend on k and ϵ without much affecting the clustering cost. Grain-based interior removal can further reduce the size of the image support. We will now study experimentally the effect of sparseness on the quality of the generated diagram.

Relative fit and relative cluster weight error. In the following, we measure the quality of fit. More precisely, let $\mathcal{P} = (P_1, \dots, P_k)$ be the diagram obtained by the algorithm and set

$$\hat{C}_i = \{x_j \in X : x_j \in \text{int}(P_i)\} \quad (i \in [k]), \quad \hat{C} = (\hat{C}_1, \dots, \hat{C}_k).$$

Let us point out that $\hat{C}$ is not a clustering of X but only of $X \cap \bigcup_{i=1}^k \text{int}(P_i)$ (and this is why we use the notation $\hat{C}$ rather than C). We define the *relative fit* Φ_G of $\hat{C}$ as

$$\Phi_G(\hat{C}) = \frac{1}{n} \left| \bigcup_{i=1}^k G_i \cap \hat{C}_i \right|.$$

Note that the points on the cell boundaries are not assigned to any clustering and, hence, count as misclassified.

Also, we will compute the *relative cluster weight error* Ψ_G defined by

$$\Psi_G(\hat{C}) = \frac{1}{n} \sum_{i=1}^k |\kappa_i - \omega(\hat{C}_i)|.$$

Pencil coresets. In this setup, the sites S and the matrices $\mathcal{A}$ are given and we consider pencils issuing from the centroids of the G_i whose lines are distributed with respect to the ellipsoids $s_i + \{x : \|x\|_{A_i} \leq 1\}$.

While all computations are fully 3D, although this is hard to visualize, Figure 2 illustrates the construction in two dimensions with rays emanating from the centroids of the depicted 2D grains.

For about 100,000 points, we obtain already a relative fit of 0.93. Enlarging the image support improves Φ_G only in the third decimal digit. The relative cluster weight error Ψ_G benefits slightly more from increasing the density, from 0.025 for 100,000 to 0.02 for 200,000 points. Figure 3 depicts the dependency of the relative fit and the relative cluster weight error on the density of the sample support. While in Theorem 2, both the number of rays and the

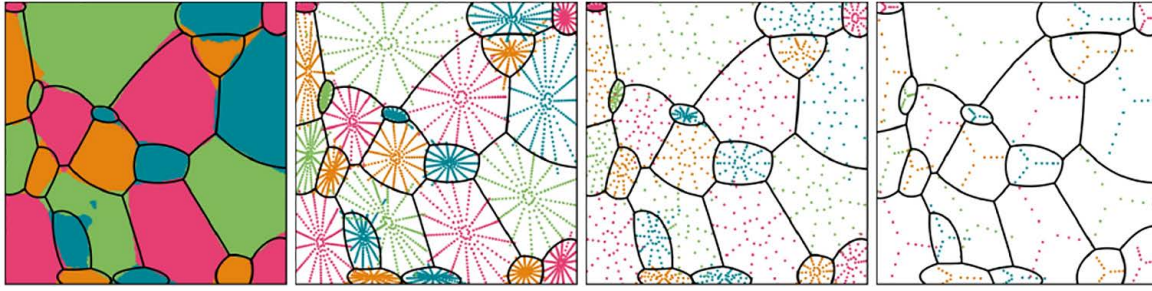


Figure 2. APDs from pencil coresets. From left to right: Computed diagrams (cell boundaries shown in black) based on decreasing image supports (the nonwhite and nonblack colored pixels in the respective images). The diagram in the left image uses the full image support (the full available data), while the others use image supports living on pencils of decreasing density.

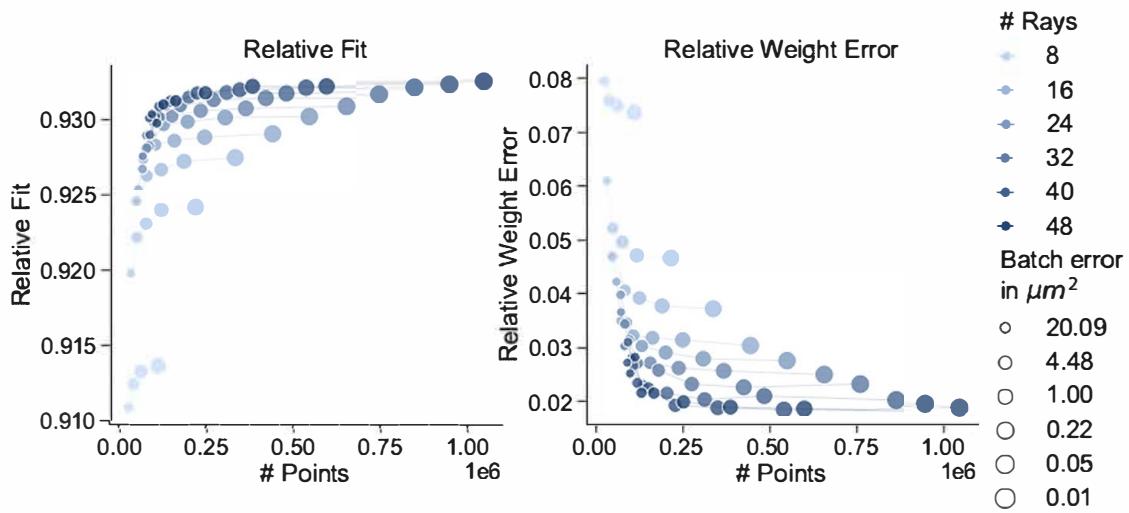


Figure 3. Relative fit (left) and relative cluster weight error (right) for pencil coresets evaluated for different cardinalities of image supports. The numbers of rays and the batch errors are signified by color and size. Also, the behavior for a fixed number of rays is indicated by connecting the corresponding points.

batch error $\sum_{x \in B} \|x - c(B)\|_A^2$ are functions of k and ϵ , we control these numbers in practice separately.

More precisely, first the number of rays is selected and each point of X is projected on the appropriate ray, as described in subsection 3.2. Then the points of each ray are partitioned into batches B and replaced by the batches' centroids $c(B)$. We do this dynamically so as to control the batch errors. Accordingly, Figure 3 also shows the influence of the batch error and the number of rays (indicated by the size and color of the points, respectively) on the relative fit and the relative weight error.

In general, the computational results suggest that, within a reasonable range, the number of rays may be more important than the batch error. This observation supports the intuition that the shape information about the grains is better captured by additional rays than by more points on each ray.

Resolution coresets. The results of a similar evaluation for the resolution coresets are illustrated in Figure 4.

Again, we obtain a relative fit of 0.93 for image supports above 100,000 points. However, the resolution coreset seems to cause a relative cluster weight error which is only half of that for pencil coresets, i.e., 0.01 instead of 0.025 for the latter. With increasing resolution, Ψ_G decreases further to 0.0022 for 1 million points. Figure 5 depicts the details.

The experimental results suggest that maintaining the grid structure in the image support yields, at the same level of sparsity, diagrams which are closer to the optimal diagram for the original instance.

Additional grain scan-based interior removal. In our practical evaluation of grain scan-based interior removal, we used a uniform setting, i.e., $\delta_i = \bar{\delta}$ for all $i \in [k]$, and employed

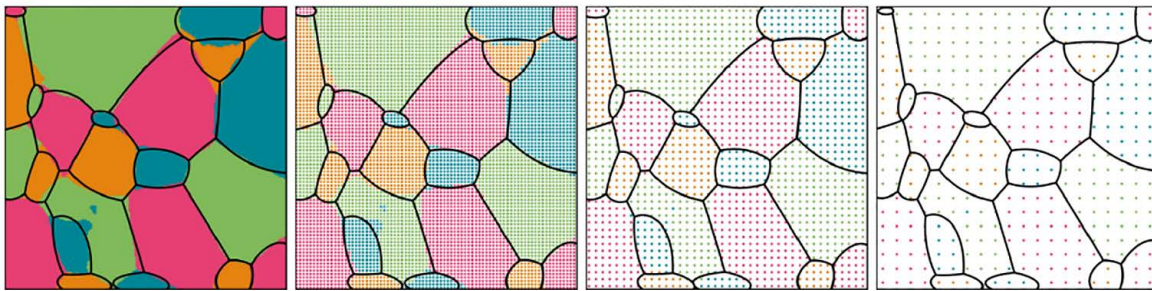


Figure 4. APDs from resolution coresets. From left to right: Computed diagrams (cell boundaries shown in black) based on decreasing image supports (the nonwhite and nonblack colored pixels in the respective images). The diagram in the left image uses the full image support (the full available data), while the others use image supports living on resolution coresets of decreasing density.

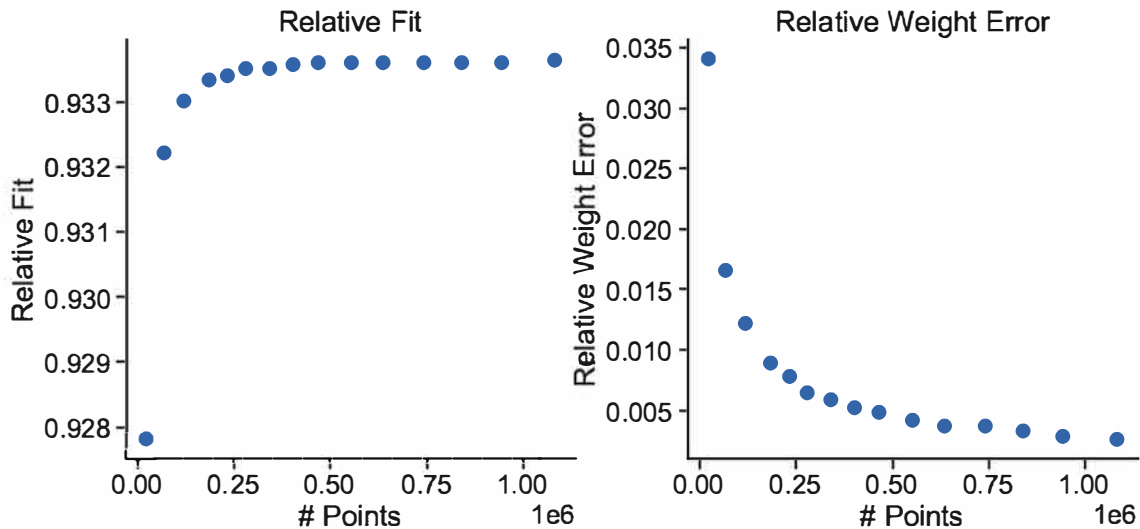


Figure 5. Relative fit (left) and relative cluster weight error (right) for resolution coresets evaluated for different cardinalities of image supports.

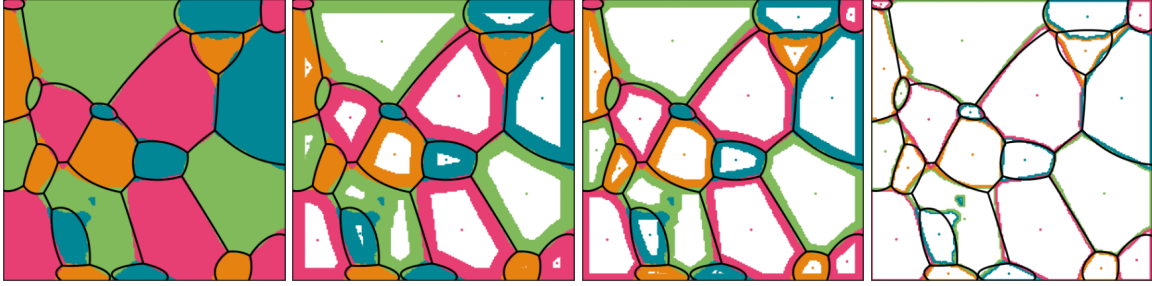


Figure 6. APDs for different degrees of grain scan-based interior removal (from left to right: none, and $\bar{\delta} = 10, 6, 2$). Shown are the computed diagrams (cell boundaries in black) based on the decreasing image supports (the nonwhite and nonblack colored pixels in the respective images).

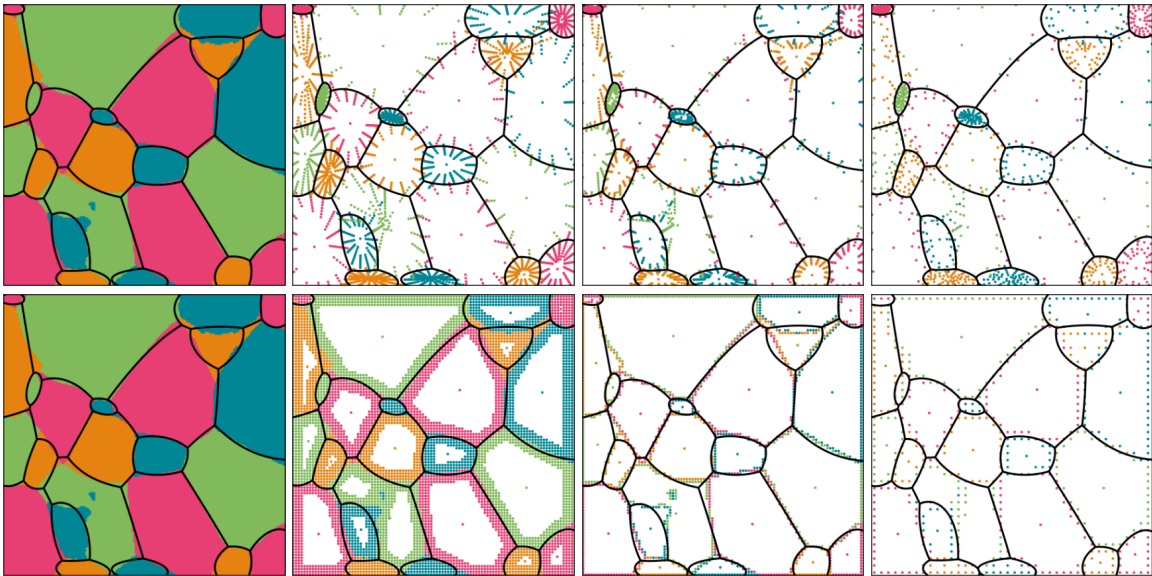


Figure 7. Pencil (top) and resolution coresets (bottom) combined with interior removal (from left to right: none, and $\bar{\delta} = 10, 6, 2$). Shown are the computed diagrams (cell boundaries in black) based on the decreasing image supports (the nonwhite and nonblack colored pixels in the respective images).

the grid graph distance. See Figure 6 for a visualization in two dimensions. (Here, and in Figure 7, we used the values $\bar{\delta} = 2, 6, 10$ to enhance the visual impression; the computations were done with a variety of different values, and we report the results for $\bar{\delta} = 2, 3, 4$ as these are most relevant.)

In order to be able to preserve the size constraints in (P) for the reduced image support, we represent each removed $\bar{\delta}$ -interior $G_i(\bar{\delta})$ by a single point at G_i 's centroid, weighted by the cardinality of the removed interior points. While, particularly for large grains, a considerable number of points may be omitted, the total length of the unmodified “grain boundaries” is typically quite large. Hence, as a sole thinning strategy, interior removal is insufficient for reducing the image support to render the linear program tractable in practice.

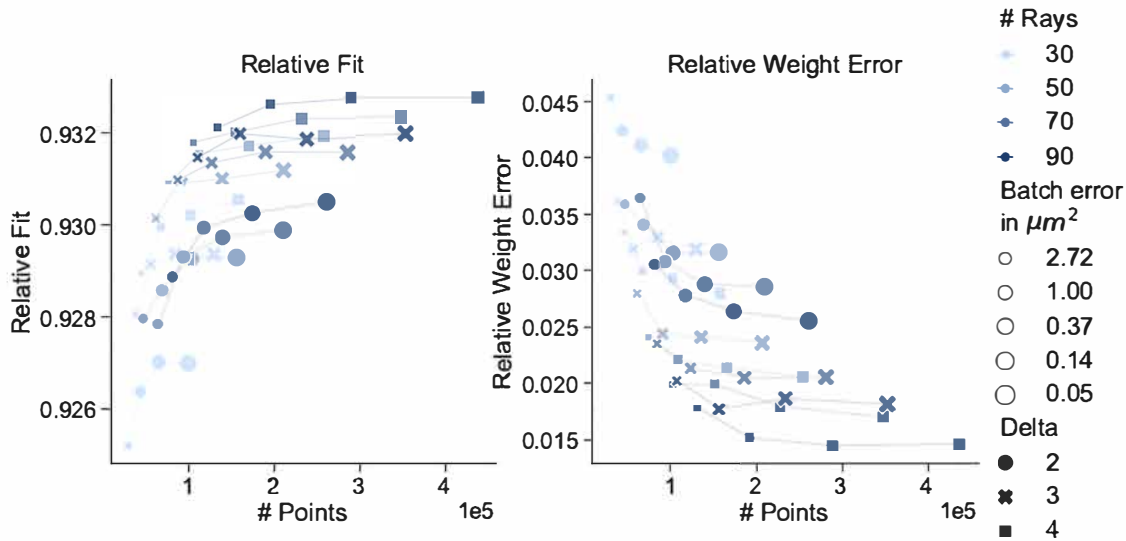


Figure 8. Combination of pencil coresets with interior removal: Dependence of $\Phi(G)$ and Ψ_G on the size of the resulting image support. Again, the color indicates the number of rays, the size refers to the batch error, while the shape of the symbol signifies the different values of $\bar{\delta}$.

When combined with the previous coreset strategies, grain scan-based interior removal can, however, add a significant additional contribution. Pencil and resolution coresets still yield many points in each grain's interior, which can be removed for further speeding up (P).

Figure 7 illustrates the combination of the coresets with interior removal in two dimensions. Again, the resulting diagrams are, as theory suggests, very similar to the original image.

In the experimental study, we investigated the effect of different parameter choices on the relative fit and the relative cluster weight error. Since we discussed the dependence of Φ_G and Ψ_G on the coreset parameters already, we focus on the dependence on the parameter $\bar{\delta}$. Recall that the $\bar{\delta}$ -interior $G_i(\bar{\delta})$ decreases with increasing $\bar{\delta}$. Fortunately, already a grid graph distance of $\bar{\delta} = 2$ produced excellent fits. Best results were obtained for $\bar{\delta} = 4$.

Figures 8 and 9 show the effect of different choices for the parameters on the relative fit and the relative cluster weight error.

In comparison, the combined thinning process based on resolution coresets outperformed that of pencil coresets: Its relative fit is higher, and its relative cluster weight error is smaller for all investigated coreset sizes. The difference is, however, generally only in the second or third digit.

All in all, using as few as 85,828 points was sufficient to obtain very high relative fit and low relative cluster weight error. Increasing the size of the image support to 1,000,000 points results in only very marginal additional improvements. In our setting, an average of only 145 points per grain sufficed for obtaining nearly the same quality of fit as the 116,476 points in the original grain scan.

5.3. s-GBPD versus DiLPM. While we showed in the previous subsection that the indirect clustering-based methods yield excellent accuracy for small sample sizes, we will now add

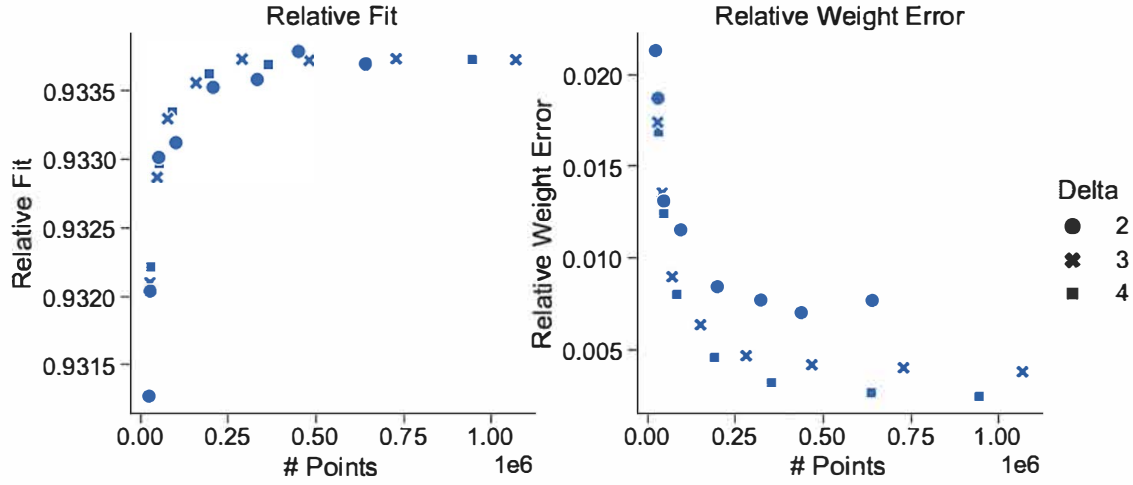


Figure 9. Combination of resolution coresets with interior removal: Dependence of $\Phi(G)$ and Ψ_G on the size of the resulting image support.

a comparison with the direct benchmark method DiLPM. Specifically, we computed $\mathcal{A}, S, \Gamma$ via DiLPM and derived the clustering $\hat{C}$ from the diagram $\mathcal{P}(\mathcal{A}, S, \Gamma)$. To decrease the number of constraints we reduced the image support with the above techniques to 85,828 points and used only distance constraints for neighboring grains in the scan.

In addition to the relative fit and the relative cluster weight error and, of course, the running time, we investigated various additional parameters. In particular, we computed the relative deviation of the centroids $c(G_i)$ and $c(\hat{C}_i)$ and also of the covariance matrices $\Sigma(G_i)$ and $\Sigma(\hat{C}_i)$. More precisely, the *relative centroid error* and the *relative covariance error* are defined as

$$\frac{1}{n} \sum_{i=1}^k \kappa_i \|c(G_i) - c(\hat{C}_i)\|_2 \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^k \kappa_i \|\Sigma(G_i) - \Sigma(\hat{C}_i)\|_2$$

In the latter case, the spectral norm $\|\cdot\|_2$ of a matrix measures the largest singular value of the covariance differences and, thus, essentially captures the difference in the largest singular vectors of the covariance matrices.

As characteristics of the grain topology affect the physical properties of polycrystals, we also address the cell combinatorics. In particular, we computed the percentage of *correct neighborhoods* of the grains. More precisely, we count a voxel as adjacent to a given one if it is contained in what is called the 26-neighborhood in discrete topology, i.e., if its centroid is among the 27 points which can be reached by at most one move in each coordinate direction of the grid graph. A grain is a neighbor of another grain if a voxel of the former is adjacent to a voxel of the latter. The neighborhood is correct if all grain neighbors in the ground truth are also neighbors in the representation.

We also computed the percentage of grains whose neighborhood has at most two errors, where an error is a missing or additional grain neighbor not present in the ground truth. In view of the results of the previous section, we focus on the resolution coresets with interior

Table 2
Comparison of s-GBPD and DiLPM for polycrystal representation.

Performance characteristics	s-GBPD	DiLPM
Relative fit Φ_G	0.9334	0.9559
Relative cluster weight error Ψ_G	0.01	0.02
Relative centroid error (in μm)	0.62	0.61
Relative covariance error (in μm^2)	15.53	8.72
Correct neighborhoods (in %)	35.70	63.45
Correct neighborhoods up to 1 error (in %)	72.08	91.71
Correct neighborhoods up to 2 errors (in %)	92.39	98.14
Computation time (in min)	2.60	627.00

removal as the specific version of s-GBPD, with image support of 85,828 points. Table 2 depicts these results.

While the Laguerre method from [24] achieves a relative fit Φ_G of 0.86 on this data set (see [24]), both our methods relying only on parts of the data improve the relative fit to a value of over 0.93. As expected, the direct model algorithm DiLPM performs better in most criteria but requires a significantly higher running time than s-GBPD. Since the differences are mostly marginal, the experimental findings indicate that we can compute a high precision 3D diagram representation of the grain scan in less than 3 minutes using s-GBPD. We expect that s-GBPD behaves similarly for other data sets as long as the resolution and the number of grains are of a similar magnitude.

6. Outlook. As we have seen, s-GBPD generates high-quality diagram representations of polycrystalline scans at speeds acceptable in many applications. If some applications require much faster algorithms, then—given the current state of the art—one needs to resort to methods that avoid optimization altogether by fixing the parameters to values that were found useful in empirical studies. Although some of these parameter choices can be interpreted in the context of Bayes classifiers [2], currently none of them comes with approximation guarantees.

Let us close with some remarks on the role the grain scan plays in applying s-GBPD. While we focused here on the situation that the whole grain scan (X, Y) is available, s-GBPD can be adjusted to require only certain measurements of the polycrystal. Grain volumes, centers, and second-order moments can, in many cases, be measured directly using tomographic techniques, and the described clustering approach can be solely based on such measurements. Both pencil and resolution coresets can still be constructed then. Grain scan-based interior removal, on the other hand, is (as emphasized correctly by its very name) based on the knowledge of (X, Y) .

It is, however, possible to remove the dependency on (X, Y) by adding an assumption that is common in *stochastic grain growth models*; see, e.g., [28, 10]. In fact, in such models, the underlying grain growth is often viewed as some realization of a stochastic process involving multivariate normal distributions $\mathcal{N}(\mu_i, \Sigma_i)$ for each grain. Their 3D mean vectors μ_i and 3×3 -covariance matrices Σ_i are determined by the measurements and correspond to $\mathcal{A}$ and

S. Under such an assumption, interior points can be removed from the coresets by removing points with high density measured by the corresponding multivariate density function. As points close to the centroids have high density, this approach removes the interior points. Hence, it is possible to obtain additional speed-ups even when only the above measurements are available.

Acknowledgments. We thank Henning Friis Poulsen for fruitful discussions and valuable comments on an earlier version of the present paper. We are also grateful to Henning Friis Poulsen and Allan Lyckegaard for providing the real-world data set used in this paper. Also, we would like to express our great appreciation to the anonymous referees for their valuable comments on a previous version on the paper.

REFERENCES

- [1] A. ALPERS, A. BRIEDEN, P. GRITZMANN, A. LYCKEGAARD, AND H. F. POULSEN, *Generalized balanced power diagrams for 3D representations of polycrystals*, Philos. Mag., 95 (2015), pp. 1016–1028, <https://doi.org/10.1080/14786435.2015.1015469>.
- [2] A. ALPERS, M. FIEDLER, P. GRITZMANN, AND F. KLEMM, *Dynamic Grain Models via Fast Heuristics for Diagram Representations*, <http://arxiv.org/abs/2204.06430>, 2022.
- [3] H. ALTENDORF, F. LATOURTE, D. JEULIN, M. FAESSEL, AND L. SAINTOYANT, *3D reconstruction of a multiscale microstructure by anisotropic tessellation models*, Image Anal. Stereol., 33 (2014), pp. 121–130, <https://doi.org/10.5566/ias.v33.p121-130>.
- [4] O. BACHEM, M. LUCIC, AND A. KRAUSE, *Practical Coresets Constructions for Machine Learning*, <http://arxiv.org/abs/1703.06476>, 2017.
- [5] J. BANHART, ED., *Advanced Tomographic Methods in Materials Research and Engineering*, Oxford University Press, Oxford, UK, 2008, <https://doi.org/10.1093/acprof:oso/9780199213245.001.0001>.
- [6] J. BARKER, G. BOLLERHEY, AND J. HAMAEEKERS, *A multilevel approach to the evolutionary generation of polycrystalline structures*, Comput. Mater. Sci., 114 (2016), pp. 54–63, <https://doi.org/10.1016/j.commatsci.2015.11.018>.
- [7] S. BEN-DAVID, N. EIRON, AND P. M. LONG, *On the difficulty of approximately maximizing agreements*, J. Comput. Systems Sci., 66 (2003), pp. 496–514, [https://doi.org/10.1016/S0022-0000\(03\)00038-2](https://doi.org/10.1016/S0022-0000(03)00038-2).
- [8] S. BORGWARDT, A. BRIEDEN, AND P. GRITZMANN, *An LP-based k-means algorithm for balancing weighted point sets*, European J. Oper. Res., 263 (2017), pp. 349–355, <https://doi.org/10.1016/j.ejor.2017.04.054>.
- [9] A. BRIEDEN, P. GRITZMANN, AND F. KLEMM, *Constrained clustering via diagrams: A unified theory and its application to electoral district design*, European J. Oper. Res., 263 (2017), pp. 18–34, <https://doi.org/10.1016/j.ejor.2017.04.018>.
- [10] S. N. CHIU, D. STOYAN, W. S. KENDALL, AND J. MECKE, *Stochastic Geometry and Its Applications*, 3rd ed., Wiley, Chichester, UK, 2013, <https://doi.org/10.1002/9781118658222>.
- [11] L. FEJES TÓTH, *Lagerungen in der Ebene, auf der Kugel und im Raum*, 2, extended and improved ed., Springer, Berlin, 1972, https://doi.org/10.1007/978-3-662-01206-2_7.
- [12] D. FELDMAN, M. SCHMIDT, AND C. SOHLER, *Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering*, SIAM J. Comput., 49 (2020), pp. 601–657, <https://epubs.siam.org/doi/10.1137/18M1209854>.
- [13] D. FELDMAN, M. SCHMIDT, AND C. SOHLER, *Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering*, in Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2013, pp. 1434–1453, <https://doi.org/10.1137/1.9781611973105.103>.
- [14] H. FICHTENBERGER, M. GILLÉ, M. SCHMIDT, C. SCHWIEGELSHOHN, AND C. SOHLER, *BICO: BIRCH meets coresets for k-means clustering*, in Algorithms—ESA 2013, Lecture Notes in Comput. Sci. 8125, Springer, Berlin, 2013, pp. 481–492, https://doi.org/10.1007/978-3-642-40450-4_41.

- [15] M. FIEDLER, *Constrained Clustering and Diagrams*, Ph.D. thesis, Technische Universität München, 2023.
- [16] M. FIEDLER AND P. GRITZMANN, *Coresets for Weight-Constrained Anisotropic Assignment and Clustering*, [arXiv:2203.10864v1](https://arxiv.org/abs/2203.10864v1), 2022.
- [17] M. FIEDLER AND P. GRITZMANN, *On the Stability of Weight-Balanced Anisotropic Clustering*, in preparation.
- [18] M. FIEDLER, P. GRITZMANN, AND F. KLEMM, *On Resolution Coresets for Constrained Clustering*, [arXiv:2204.00893](https://arxiv.org/abs/2204.00893), 2022.
- [19] P. GRITZMANN AND V. KLEE, *Computational convexity*, in Handbook of Discrete and Computational Geometry, J. E. Goodman, J. O'Rourke, and C. D. Toth, eds., CRC Press, Boca Raton, FL, 2017, pp. 937–968, <https://doi.org/10.1201/9781315119601>.
- [20] J. GUILLEMINOT, A. NOSHADRAVAN, C. SOIZE, AND R. G. GHANEM, *A probabilistic model for bounded elasticity tensor random fields with application to polycrystalline microstructures*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 1637–1648, <https://doi.org/10.1016/j.cma.2011.01.016>.
- [21] P. C. HANSEN, H. O. SØRENSEN, Z. SÜKÖSD, AND H. F. POULSEN, *Reconstruction of single-grain orientation distribution functions for crystalline materials*, SIAM J. Imaging Sci., 2 (2009), pp. 593–613, <https://doi.org/10.1137/080726021>.
- [22] S. HAR-PELED AND A. KUSHAL, *Smaller coresets for k -median and k -means clustering*, Discrete Comput. Geom., 37 (2007), pp. 3–19, <https://doi.org/10.1007/s00454-006-1271-x>.
- [23] M. KÜHN AND M. O. STEINHAUSER, *Modeling and simulation of microstructures using power diagrams: Proof of the concept*, Appl. Phys. Lett., 93 (2008), 034102, <https://doi.org/10.1063/1.2959733>.
- [24] A. LYCKEGAARD, E. M. LAURIDSEN, W. LUDWIG, R. W. FONDA, AND H. F. POULSEN, *On the use of Laguerre tessellations for representations of 3D grain structures*, Adv. Eng. Mater., 13 (2011), pp. 165–170, <https://doi.org/10.1002/adem.201000258>.
- [25] R. D. MACPHERSON AND D. J. SROLOVITZ, *The von Neumann relation generalized to coarsening of three-dimensional microstructures*, Nature, 446 (2007), pp. 1053–1055, <https://doi.org/10.1038/nature05745>.
- [26] J. MATOUŠEK, *Lectures on Discrete Geometry*, Grad. Texts in Math. 212, Springer, New York, 2002, <https://doi.org/10.1007/978-1-4613-0039-7>.
- [27] O. V. MISHIN, V. M. SEGAL, AND S. FERRASSE, *Quantitative microstructural characterization of thick aluminum plates heavily deformed using equal channel angular extrusion*, Metall. Mater. Trans. A, 43 (2012), pp. 4767–4776, <https://doi.org/10.1007/s11661-012-1287-1>.
- [28] A. OKABE, B. BOOTS, K. SUGIHARA, AND S. N. CHIU, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., Wiley, New York, 2000, <http://dx.doi.org/10.1002/9780470317013>.
- [29] L. PETRICH, O. FURAT, M. WANG, C. E. KRILL III, AND V. SCHMIDT, *Efficient fitting of 3D tessellations to curved polycrystalline grain boundaries*, Front. Mater., 8 (2021), 760602, <https://doi.org/10.3389/fmats.2021.760602>.
- [30] H. F. POULSEN, *Three-Dimensional X-Ray Diffraction Microscopy: Mapping Polycrystals and Their Dynamics*, Springer, Berlin, 2004, <https://doi.org/10.1007/b97884>.
- [31] B. SCHÖLKOPF AND A. SMOLA, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2002, <https://doi.org/10.7551/mitpress/4175.001.0001>.
- [32] A. SCHRIJVER, *Theory of Linear and Integer Programming*, Wiley, Chichester, UK, 1986.
- [33] O. ŠEDIVÝ, T. BRERETON, D. WESTHOFF, L. POLÍVKA, V. BENEŠ, V. SCHMIDT, AND A. JÄGER, *3D reconstruction of grains in polycrystalline materials using a tessellation model with curved grain boundaries*, Philos. Mag., 96 (2016), pp. 1926–1949, <https://doi.org/10.1080/14786435.2016.1183829>.
- [34] O. ŠEDIVÝ, J. M. DAKE, C. E. KRILL III, V. SCHMIDT, AND A. JÄGER, *Description of the 3D morphology of grain boundaries in aluminium alloys using tessellation models generated by ellipsoids*, Image Anal. Stereol., 36 (2017), pp. 5–13, <https://doi.org/10.5566/ias.1656>.
- [35] C. SOHLER AND D. P. WOODRUFF, *Strong coresets for k -median and subspace approximation: Goodbye dimension*, in Proceedings of the 59th Annual Symposium on Foundations of Computer Science, IEEE, 2018, pp. 802–813, <https://doi.org/10.1109/FOCS.2018.00081>.
- [36] A. SPETTL, T. WERZ, C. E. KRILL III, AND V. SCHMIDT, *Parametric representation of 3D grain ensembles in polycrystalline microstructures*, J. Stat. Phys., 154 (2014), pp. 913–928, <https://doi.org/10.1007/s10955-013-0893-7>.

- [37] A. SPETTL, T. BRERETON, Q. DUAN, T. WERZ, C. E. KRILL III, D. P. KROESE, AND V. SCHMIDT, *Fitting Laguerre tessellation approximations to tomographic image data*, Philos. Mag., 96 (2016), pp. 166–189, <https://doi.org/10.1080/14786435.2015.1125540>.
- [38] K. TEFERRA AND L. GRAHAM-BRADY, *Tessellation growth models for polycrystalline microstructures*, Comput. Mater. Sci., 102 (2015), pp. 57–67, <https://doi.org/10.1016/j.commatsci.2015.02.006>.
- [39] K. TEFERRA AND D. J. ROWENHORST, *Direct parameter estimation for generalised balanced power diagrams*, Philos. Mag. Lett., 98 (2018), pp. 79–87, <https://doi.org/10.1080/09500839.2018.1472399>.
- [40] H. TELLEY, T. M. LIEBLING, A. MOCELLIN, AND F. RIGHETTI, *Simulating and modelling grain growth as the motion of a weighted Voronoi diagram*, Mater. Sci. Forum, 94-96 (1992), pp. 301–306, <https://doi.org/10.4028/www.scientific.net/MSF.94-96.301>.
- [41] T. XU AND M. LI, *Topological and statistical properties of a constrained Voronoi tessellation*, Philos. Mag., 89 (2009), pp. 349–374, <https://doi.org/10.1080/14786430802647065>.
- [42] J. ZHANG, Y. ZHANG, W. LUDWIG, D. ROWENHORST, P. W. VOORHEES, AND H. F. POULSEN, *Three-dimensional grain growth in pure iron. Part I. Statistics on the grain level*, Acta Mater., 156 (2018), pp. 76–85, <https://doi.org/10.1016/j.actamat.2018.06.021>.

Dynamic grain models via fast heuristics for diagram representations

Andreas Alpers^a, Maximilian Fiedler^b, Peter Gritzmann^b and Fabian Klemm^b

^aDepartment of Mathematical Sciences, University of Liverpool, Liverpool, UK; ^bZentrum Mathematik, Technische Universität München, Garching bei München, Germany

ABSTRACT

The present paper introduces a mathematical model for analysing dynamic grain growth. In particular, we show how the characteristic measurements grain volumes, centroids, and central second-order moments at discrete moments in time can be quickly turned into a continuous description of the grain growth process in terms of geometric diagrams (which largely generalize the well-known Voronoi and Laguerre tessellations). We give a theoretical analysis of common optimization-free heuristics in terms of discriminant analysis and evaluate the computational behaviour of our algorithm on real-world data.

ARTICLE HISTORY

Received 14 April 2022

Accepted 10 February 2023

KEYWORDS

Power diagrams; generalized balanced power diagrams; tessellations; grains; polycrystals; grain growth

1. Introduction

Grain growth is an important field of study in materials science as the resulting grain structures strongly influence the mechanical and physical properties of metals, ceramics, and other polycrystalline materials [1, 2]. Although, starting with works of Smith, von Neumann, and Mullins in the 1950s [3–5], a number of different models have been introduced over the years, understanding and controlling grain growth remains challenging, both in theory and practice. For instance, the recent empirical study [6] demonstrates that neither the classical Hillert model [7] nor the MacPherson–Srolovitz model [8] capture anisotropic grain growth adequately. Even more drastically, [9] concludes that ‘a new model for grain boundary migration is needed to predict microstructure evolution’; see also [10–13].

The present paper introduces and analyses a novel tool which, when applied to real data, provides an empirical grain growth representation that can be used to analyse physical models and simulate grain growth. More precisely, we

CONTACT Andreas Alpers  andreas.alpers@liverpool.ac.uk  Department of Mathematical Sciences, University of Liverpool, Liverpool L69 7ZL, UK

© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

develop a mathematical interpolation/extrapolation model for quickly turning grain information acquired at discrete moments in time into a continuous description of the grain growth process. It employs diagram representations of grain maps (which will be formally introduced in Section 2) based on measured (or simulated) estimates of the grain volumes, centroids, and, if available, central second-order moments. Such characteristics can be obtained, for instance, by utilizing tomographic techniques [1, 14] or from otherwise available grain maps.

The diagram representations considered in this paper are anisotropic generalizations of previously used (isotropic) Laguerre and Voronoi tessellations [15–17] and allow for curved boundaries and non-convex grains. While these representations are based on a phenomenological model whose physical foundation is still not fully understood, they have consistently been reported to capture the grain boundaries in polycrystalline materials quite accurately; see [18–25].

Several sophisticated optimization algorithms are available for computing or approximating such diagrams with high accuracy, [18, 20], but they require relatively large computational cost. This can be a bottleneck, particularly in dynamic 3D studies. We therefore focus in the present paper on extremely fast optimization-free heuristics for fitting diagrams to measured data. Methods for the related but generally different task of generating synthetic grain structures that follow a given distribution are discussed in [26–30].

First, we use discriminant analysis to interpret, for the first time in literature, a heuristic (subsequently referred to as H1) which was introduced in [15] for the isotropic case and extended in [25] to the anisotropic case. Our analysis leads naturally to an even simpler and faster heuristic H2, proposed in [19], which, again for the first time, can now be explained to perform equally well under certain conditions. Also, the comparison of H1 and H2 and, in particular, our interpretation of the so-called size parameters, suggests that diagram parameters may provide additional insight into the stage of the growth process, and this will be used later on in the paper. Our theoretical foundation of the heuristics is then empirically verified. In fact, we apply H1 and H2 to real-world 3D grain map data in order to compare their quality. Then, we introduce an interpolation/extrapolation model to obtain a continuous description of the grain growth process based on measurements at only a few moments in time. Finally, we evaluate the computational behaviour of our algorithm on a real-world data set yielding an empirical grain growth model. Our study indicates that the growth process converges towards particularly simple diagrams, which seem to be energetically favourable.

The paper is organized as follows. Section 2 introduces the relevant diagrams and formalizes the concept of grain scans. Sections 3 and 4 address optimization-free heuristics, both, in terms of a theoretical analysis and a practical evaluation on real-world data. Section 5 introduces the interpolation/

extrapolation model which is capable of extending measurements at a few time steps to a continuous grain growth model very quickly. Further, the results of our empirical study on practical grain data are presented. Section 6 concludes with some final remarks.

2. Diagrams and grain scans

Adequate representations of grain maps can be valuable tools for understanding the physical principles of grain growth and the properties of the resulting material. In fact, the diagram representations studied in this paper require only a few parameters per grain and are thus relatively easy to process. As they exhibit geometric and combinatorial features of grain maps which are not readily available otherwise, such representations may contribute to predicting and, potentially, even controlling the forming of new materials.

Some or all parameters which specify a diagram can be estimated from certain tomographic measurements, and optimizing the remaining ones then constitutes an inverse problem; see [31]. Previous studies [18–25] indicate that, in particular, grain volumes, centroids, and central second-order moments govern decisive properties of grain maps.

Let us now formally introduce the appropriate class of diagrams. Let $d \in \mathbb{N}$ denote the *dimension* of space. Here, we are mainly interested in the 3 dimensional case. However, our approach works for general d , and, in fact, planar sections or surfaces of polycrystals are also relevant in practice. Further, let $k \in \mathbb{N}$ denote the number of different grains in the underlying grain map. An *anisotropic power diagram* is specified by the following data for each $i \in [k] := \{1, \dots, k\}$:

- a positive definite symmetric matrix $A_i \in \mathbb{R}^{d \times d}$,
- a *site* $s_i \in \mathbb{R}^d$, and a
- *size* $\gamma_i \in \mathbb{R}$.

For simpler reference, we collect these parameters in the two families $\mathcal{A} = \{A_1, \dots, A_k\}$ and $\Gamma = \{\gamma_1, \dots, \gamma_k\}$, where repetitions of elements are permitted, and the set $S = \{s_1, \dots, s_k\}$. The sites specify the positions of the grains while the matrices in $\mathcal{A}$ describe characteristics of their shapes. In fact, each grain is equipped with its own *ellipsoidal norm* $\|\cdot\|_{A_i}$ given through A_i by

$$\|x\|_{A_i} = \sqrt{x^\top A_i x}, \quad (x \in \mathbb{R}^d).$$

The sizes γ_i are used to control the grain volume.

Given $\mathcal{A}$, S , and Γ , the *anisotropic power diagram* (in the following often simply referred to as diagram)

$$\mathcal{P} = \mathcal{P}(\mathcal{A}, S, \Gamma) = \{P_1, \dots, P_k\},$$

is a decomposition of $\mathbb{R}^d$ into cells P_i defined by

$$P_i = \left\{ x \in \mathbb{R}^d : \|x - s_i\|_{A_i}^2 + \gamma_i \leq \|x - s_\ell\|_{A_\ell}^2 + \gamma_\ell, \forall \ell \in [k] \setminus \{i\} \right\}.$$

In particular, two different cells do not share interior points, i.e. $\text{int}(P_i) \cap \text{int}(P_\ell) = \emptyset$ for $i \neq \ell$. See [32, 33] for examples and further properties of different classes of diagrams.

Diagrams can be viewed as continuous (or resolution-independent) representations of polycrystalline samples. Conventionally, the samples are represented as voxel-based digital images at some resolution, called *grain maps*. More generally, given a set of points $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^d$ in the sample and labels, i.e. $y_1, \dots, y_n \in [k]$, with $k \in \mathbb{N}$ denoting again the number of different grains in the sample, we will call the labeled data set $\{(x_j, y_j) : j \in [n]\}$ a *grain scan* or *polycrystal scan*. In the following, we refer to it as (X, Y) with the understanding, that the indices of the labels in the family $Y = \{y_1, \dots, y_n\}$ correspond to the indices of the points in X . A grain scan therefore provides a partitioning of X into $G_i = \{x_j : (x_j, i) \in (X, Y)\}$, the (discretizations of the) *grains*.

We can measure the fit of the diagram in terms of the symmetric difference of G_i and $X \cap P_i$ for all $i \in [k]$. (Points on the boundary of two or more cells are assigned according to some preference criterion or, conservatively, can be viewed as classification error, but are essentially irrelevant in our context and hence generally discarded.)

Usual grain maps are special grain scans where the sample is assumed to be in the range $[0, 1]^d$ and points of X are identified with the points of $[0, 1]^d$ in the grid $\rho\mathbb{Z}^d$ for some *resolution* $\rho \in (0, 1)$. Note that, with $\kappa_i = |G_i|$, the term $\rho^d \kappa_i$ can be regarded as an approximation of the volume v_i of the i th grain.

If a grain scan is available the computation of a best fitting diagram requires the optimization of all three characteristics $\mathcal{A}$, S , and Γ . While this can indeed be done, see [20, 31], it is, however, too time-consuming, given current optimization technology, for dynamic 3D applications. Hence, we will in the following assume that $\mathcal{A}$ and S are available through measurements and Γ will be determined by a heuristic which may, additionally, use grain volume information, i.e. (approximations of) $v_1, \dots, v_k$. As such heuristics abstain from any optimization, they are computationally fast but potentially result in a higher misclassification error.

3. Optimization-free heuristics

This section considers two heuristics that derive all characteristic parameters $\mathcal{A}$, S , Γ for an anisotropic power diagram directly from given measurements. These heuristics, introduced in [15, 19, 25], avoid optimization routines altogether, are fast and, as we will see, already provide a reasonable classification accuracy. More specifically, we provide a new interpretation of the heuristics within the framework of discriminant analysis.

3.1. Choices for $\mathcal{A}$ and S

Discriminant analysis is a popular classification technique in statistics and provides a different perspective on the a priori choices for the characteristic parameters $(\mathcal{A}, S, \Gamma)$; see e.g. [34]. Within its realm, the grain scan is regarded as a sample of the multivariate random variable $(\mathcal{X}, \mathcal{Y})$ where $\mathcal{X}$ specifies a point in $[0, 1]^d$ and $\mathcal{Y}$ gives the index $i \in [k]$ of the corresponding grain. Discriminant analysis estimates the distribution of the multivariate random variable from samples and assigns a point to the class with the highest probability p . Hence, the decision boundary between two classes i, ℓ consists of those points x that have an equal conditional probability of lying in either class, i.e.

$$\{x \in \mathbb{R}^d : p(\mathcal{Y} = i \mid \mathcal{X} = x) = p(\mathcal{Y} = \ell \mid \mathcal{X} = x)\}.$$

For its estimation, discriminant analysis applies the maximum likelihood principle under the assumption that the members from each class are normally distributed. As each normal distribution $\mathcal{N}(\mu_i, \Sigma_i)$ is fully determined by its d -dimensional mean vector μ_i and its $(d \times d)$ -covariance matrix Σ_i , it is these characteristics that need to be estimated. It is well known that the maximum likelihood principle results in the choices of the sites S as the centroids and the matrices $\mathcal{A}$ as the inverse of the covariance matrices of the sample, i.e. more precisely,

$$\mu_i = c_i = \frac{1}{|G_i|} \sum_{x_j \in G_i} x_j,$$

$$A_i = (\Sigma(G_i))^{-1} = \left(\frac{1}{|G_i|} (x_j - c_i : x_j \in G_i)(x_j - c_i : x_j \in G_i)^T \right)^{-1}$$

for $i \in [k]$. Note that the latter requires that the matrices $\Sigma(G_i)$ have full rank d . This can be assumed here as, in the underlying model, grains that are degenerate to volume 0 do not matter. The above maximum likelihood setting is precisely the choice for $\mathcal{A}$ and S suggested in [18], and subsequently also used in [25]. Also, estimates for these parameters are available through tomographic measurement.

Let us point out that, in practice, discriminant analysis is often applied even when the underlying assumption of a normal distribution is only met approximately (as is the case here). This is justified as the procedure is generally reported to be quite robust; see, e.g. [35].

3.2. Choices for Γ

We will now turn to the third set of characteristics, i.e. the sizes Γ . We show first that the choice of Γ in heuristic H1 can also be interpreted within the paradigm

of discriminant analysis with a specific prior. Building on this interpretation, a different natural choice will lead to H2.

To begin with, let us briefly recall the volume argument of [15] and (in its anisotropic generality) [25] that leads to the specific choice for Γ in heuristic H1.

Let $A \in \mathbb{R}^{d \times d}$ be positive definite and symmetric. As A can be diagonalized there exist an orthogonal matrix $U \in \mathbb{R}^{d \times d}$ and a diagonal matrix $D = \text{diag}(\lambda_1, \dots, \lambda_d) \in \mathbb{R}^{d \times d}$ with $\lambda_1, \dots, \lambda_d > 0$ such that $A = UDU^T$. Then, using the abbreviation $D^{1/2} = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_d})$, we have for the unit ball $\mathbb{B}_A^d$ and $\rho \in (0, \infty)$

$$\begin{aligned} \rho \mathbb{B}_A^d &= \{x \in \mathbb{R}^d : \|x\|_A \leq \rho\} = \{x \in \mathbb{R}^d : x^T A x \leq \rho^2\} \\ &= \{x \in \mathbb{R}^d : x^T U D U^T x \leq \rho^2\} = \rho U D^{-1/2} \mathbb{B}_2^d. \end{aligned}$$

Hence, the volume $\text{vol}(\rho \mathbb{B}_A^d)$ of this ellipsoid is given by

$$\text{vol}(\rho \mathbb{B}_A^d) = \rho^d \text{vol}(U D^{-1/2} \mathbb{B}_2^d) = \rho^d \det(A)^{-1/2} \text{vol}(\mathbb{B}_2^d).$$

Regarding the ellipsoid $c_i + \rho \mathbb{B}_{A_i}^d$ as an approximation of the i th grain, it is now natural to choose γ_i according to the volume condition

$$\text{vol}\left(\left\{x \in \mathbb{R}^d : \|x\|_{A_i}^2 + \gamma_i \leq 0\right\}\right) = v_i$$

involving the available approximation v_i for the volume of the i th grain. A simple computation yields

$$\gamma_i = -\rho^2 = -\left(\frac{v_i}{\text{vol}(\mathbb{B}_{A_i}^d)}\right)^{2/d} = -\left(\frac{v_i \sqrt{\det(A_i)}}{\text{vol}(\mathbb{B}_2^d)}\right)^{2/d}$$

For $d = 3$, and with the specific setting of $A_i = (\Sigma(G_i))^{-1}$, this yields the choice

$$\gamma_i = -\left(\frac{3v_i}{4\pi\sqrt{\det(\Sigma(G_i))}}\right)^{2/3}$$

from [15, 25] (the heuristic H1). In order to interpret this choice via discriminant analysis, we determine which *probability distribution*, called *prior*, will lead to the same decision boundaries between the classes. So, let

$$\pi_1, \dots, \pi_k \in [0, \infty), \quad N = \sum_{i=1}^k \pi_i \neq 0.$$

Then $(\pi_1, \dots, \pi_k)$ specifies the a priori ‘belief’ about the *prior probability* π_i/N that a point (regardless of its position) belongs to cluster i , i.e.

$$\frac{\pi_i}{N} = p(\mathcal{Y} = i), \quad (i \in [k]).$$

The computation of the decision boundary is then facilitated via *Bayes’ Theorem*. As each class is assumed to be normally distributed, Bayes’ Theorem involves the conditional density functions

$$f_{\mathcal{X}|\mathcal{Y}=i}(x) = \frac{\det(A_i)^{1/2}}{(2\pi)^{d/2}} \cdot \exp\left(-\frac{1}{2}(x - c_i)^\top A_i(x - c_i)\right)$$

of the k estimated normal distributions $\mathcal{N}(\mu_i, \Sigma_i)$, i.e. with

$$\mu_i = c_i, \quad \Sigma_i = \Sigma(G_i) = A_i^{-1}.$$

In this situation, Bayes’ Theorem states that

$$p(\mathcal{Y} = i | \mathcal{X} = x) = \frac{f_{\mathcal{X}|\mathcal{Y}=i}(x) \pi_i}{\sum_{\ell=1}^k f_{\mathcal{X}|\mathcal{Y}=\ell}(x) \pi_\ell}.$$

Therefore, the maximum likelihood decision boundaries between two classes i, ℓ can be determined by solving the equation

$$f_{\mathcal{X}|\mathcal{Y}=i}(x) \cdot \pi_i = f_{\mathcal{X}|\mathcal{Y}=\ell}(x) \cdot \pi_\ell.$$

Taking the logarithm, using that $(x - c_i)^\top A_i(x - c_i) = \|x - c_i\|_{A_i}^2$, and simplifying, this condition turns into the equation

$$\begin{aligned} \|x - c_i\|_{A_i}^2 - \log(\det(A_i)) - 2 \log(\pi_i) &= \|x - c_\ell\|_{A_\ell}^2 - \log(\det(A_\ell)) \\ &\quad - 2 \log(\pi_\ell). \end{aligned}$$

Hence, these decision boundaries coincide with that of the diagram with the parameters $(\mathcal{A}, \mathcal{S}, \Gamma)$ if and only if

$$\gamma_i = \gamma_i^{(1)} = -\log(\det(A_i)) - 2 \log(\pi_i) + \gamma, \quad (i \in [k]),$$

where $\gamma \in \mathbb{R}$ is an arbitrary constant. (Recall that the diagrams are invariant under such common additions to the sizes.) For the above choice of the sizes in H1, we hence obtain

$$\pi_i = \pi_i^{(1)} = \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}} \cdot \exp\left(\frac{1}{2} \left(\frac{v_i \sqrt{\det(A_i)}}{\text{vol}(\mathbb{B}_2^d)}\right)^{2/d}\right), \quad (i \in [k]).$$

Since

$$\frac{1}{\sqrt{\det(A_i)}} = \sqrt{\det(\Sigma(G_i))} = \frac{1}{\text{vol}(\mathbb{B}_2^d)} \cdot \text{vol}(\mathbb{B}_{A_i}^d),$$

the first factor is proportional to the volume of the ellipsoid $\mathbb{B}_{A_i}^d$. The second factor is a ‘correction term’ which increases exponentially if the measured grain volume $\rho^d \kappa_i$ exceeds $\text{vol}(\mathbb{B}_{A_i}^d)$. The first term seems natural while the second is quite unusual.

Within the realm of discriminant analysis it appears reasonable to simply set $\gamma_i = 0$ for all clusters i . Then we obtain the priors

$$\pi_i = \pi_i^{(2)} = \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}}, \quad (i \in [k]),$$

which are proportional to the volumes of the ellipsoids corresponding to the covariance matrices of the grains. Thus, this simpler prior distribution encodes the assumption that the ellipsoid volumes already represent the grain volumes well. This justifies the choice

$$\gamma_1^{(2)} = \dots = \gamma_k^{(2)} = 0,$$

proposed in [19]. We will refer to the respective heuristic as H2.

In the next section we compare the behaviour of H1 and H2 on available real-world grain scans.

4. Comparison of H1 and H2

We compare the two heuristics H1 and H2 on a real-world 3D grain scan and indicate how their difference can be interpreted. This may be of particular interest for understanding the state of a grain growth process and will be taken up in Section 5.

4.1. Experimental evaluation

We compare the methods with a data set taken from [15] that has been obtained by a synchrotron micro-tomograph experiment conducted on a metastable beta titanium alloy (Ti β 21S). The material was scanned with the volume of $240\mu\text{m} \times 240\mu\text{m} \times 420\mu\text{m}$ at a resolution corresponding to a voxel size of $0.7\mu\text{m}$. This scan resulted in a 3D voxel image of size $339 \times 339 \times 599$ composed of a total of 591 grains. All our computations are carried out in 3D, i.e. $d = 3$.

In the following, we assess the quality of fit by means of different measures. Let $\mathcal{P} = \{P_1, \dots, P_k\}$ be a diagram (obtained by H1, H2, or, for that matter, any other algorithm). We set

$$\hat{C}_i = \{x_j \in X : x_j \in \text{int}(P_i)\} \quad (i \in [k]), \quad \hat{C}(\mathcal{P}) = \hat{C} = (\hat{C}_1, \dots, \hat{C}_k).$$

Note that $\hat{C}$ is not a clustering of X but only of $X \cap \bigcup_{i=1}^k \text{int}(P_i)$; this is indicated by writing $\hat{C}$ rather than C .

A natural measure how well $\mathcal{P}$ captures the grain scan $G = (G_1, \dots, G_k)$ is the *relative fit* or *accuracy* Φ_G of $\hat{C}$ defined by

$$\Phi_G(\hat{C}) = \frac{1}{n} \left| \bigcup_{i=1}^k (G_i \cap \hat{C}_i) \right|.$$

Let us point out that, as they are not assigned within $\hat{C}$, all points on the cell boundaries count as misclassified.

In Table 1 we will also report on the behaviour with respect to the *relative cluster weight error*

$$\Psi_G(\hat{C}) = \frac{1}{n} \sum_{i=1}^k |\kappa_i - |\hat{C}_i||,$$

and the relative deviation of the centroids $c(G_i)$ and $c(\hat{C}_i)$, and the covariance matrices $A_i^{-1} = \Sigma_i = \text{Cov}(G_i)$ and $\text{Cov}(\hat{C}_i)$, respectively. Formally, the *relative centroid error* and the *relative covariance error* are defined as

$$\frac{1}{n} \sum_{i=1}^k \kappa_i \|c(G_i) - c(\hat{C}_i)\|_2 \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^k \kappa_i \|\Sigma_i - \text{Cov}(\hat{C}_i)\|_2.$$

For the latter, we used the spectral norm. This matrix norm measures the largest singular value of the covariance differences and, thus, intuitively captures the difference in the largest singular vectors of the covariance matrices. Note that, as defined, the latter two measures depend on the original physical dimensions of the sample. We refrain from any additional normalization as we do not conduct inter-sample studies here.

We will also address combinatorial features. More precisely, we consider the percentage of grains with a correct neighbourhood, i.e. with all grain neighbours in the ground truth being also neighbours in the representation. Here, a grain is a neighbour of another grain if a voxel of the former grain is adjacent (face-connected, edge-connected or vertex-connected) to a voxel of the later grain. We also provide this percentage when one and, respectively, two errors in the neighbourhood are allowed, i.e. when there is either one additional or one missing voxel.

Table 1. Comparison of H1 and H2.

Performance characteristics	H1	H2
Accuracy (in %)	92.98652	92.98658
Relative cluster weight error	0.02869	0.02869
Relative centroid error (in μm)	0.64346	0.64347
Relative covariance error (in μm^2)	15.25372	15.25376
Correct neighbourhoods (in %)	42.30118	42.30118
Correct neighbourhoods up to 1 Error (in %)	80.54146	80.54146
Correct neighbourhoods up to 2 Errors (in %)	94.92386	94.92386
Running time (in s)	1.49641	0.00000

The results of our evaluation are shown in [Table 1](#).

Let us point out that the displayed running times involve the heuristics but, for a fairer comparison, do not include the computation of the performance criteria. Hence, the entry 0 for the running time of H2 refers to the fact that all parameters defining $\mathcal{P}$ are given upfront. Also, while we exercised appropriate care for the implementations, we do not claim that our code is fully optimized.

In terms of fit, we observe that both heuristics perform extremely well, in fact, almost identical, on the given real-world data set. We will now offer an explanation for the latter behaviour.

4.2. Similarity and difference

While both heuristics, H1 and H2, employ the same matrices $\mathcal{A}$ and sites S , they generally differ in the sizes Γ of the resulting diagrams. Hence, the nearly identical performance depicted in [Table 1](#) for all applied quality criteria may come as a surprise. As observed before, the priors underlying H1 and H2 denoted $\pi_i^{(1)}$ and $\pi_i^{(2)}$, as before, differ by a factor that depends exponentially on the deviation of $\text{vol}(\mathbb{B}_{A_i}^d)$ from the measured grain volume v_i .

Now, let α_i be a positive real number such that

$$v_i = \alpha_i \cdot \text{vol}(\mathbb{B}_{A_i}^d) = \alpha_i \cdot \frac{\text{vol}(\mathbb{B}_2^d)}{\sqrt{\det(A_i)}}.$$

Then, for each $i \in [k]$,

$$\begin{aligned} \pi_i^{(1)} &= \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}} \cdot \exp\left(\frac{1}{2} \left(\frac{v_i \sqrt{\det(A_i)}}{\text{vol}(\mathbb{B}_2^d)}\right)^{2/d}\right) = \frac{e^{\gamma/2}}{\sqrt{\det(A_i)}} \cdot \exp\left(\frac{1}{2} \alpha_i^{2/d}\right) \\ &= \pi_i^{(2)} \cdot \exp\left(\frac{1}{2} \alpha_i^{2/d}\right), \end{aligned}$$

and, expressed in terms of the sizes $\gamma_i^{(1)}$ used in H1, we obtain

$$\gamma_i^{(1)} = -\left(\frac{v_i}{\text{vol}(\mathbb{B}_{A_i}^d)}\right)^{2/d} = -\alpha_i^{2/d}. \quad (1)$$

Now, suppose that all factors α_i are identical, i.e. $v_i = \alpha \cdot \text{vol}(\mathbb{B}_{A_i}^d)$ for some positive real number α independent of i . Then, of course,

$$\pi_i^{(1)} = \pi_i^{(2)} \cdot \exp\left(\frac{1}{2} \alpha^{2/d}\right), \quad \gamma_i^{(1)} = -\alpha^{2/d}, \quad (i \in [k]).$$

Hence, the (normalized) priors for H1 and H2 coincide, which implies that the maximum likelihood decision boundaries for the two heuristics are the same.

Further, since diagrams are invariant under the addition of a constant to each size, the diagrams produced by H1 and H2 are identical.

We suspect that grain structures for which the volumes of the unit balls $\mathbb{B}_{A_i}^d$ with respect to $A_i = (\Sigma(G_i))^{-1}$ are nearly proportional to the grain volumes v_i might turn out to be energetically favourable. According to the above reasoning, the former may be quantified by the variance of the powers $\Gamma^{d/2} = (\gamma_1^{d/2}, \dots, \gamma_k^{d/2})$ computed in H1, i.e.

$$\text{var}(\Gamma^{d/2}) = E\left(\Gamma^{d/2} - E(\Gamma^{d/2})\right) = \frac{1}{k} \sum_{i=1}^k \left(\gamma_i^{d/2} - \frac{1}{k} \sum_{i=1}^k \gamma_i^{d/2} \right)^2.$$

As an example, we will in Section 5 compute the evolution of $\text{var}(\Gamma^{3/2})$ at different moments in time for a real-world 3D grain growth process. In terms of H1 and H2, we might expect that if the two heuristics perform differently in practice, the current grain structure will not be energetically minimal, and further growth will result in an improved structure.

We close this section with a ‘constructed’ theoretical example which sheds some additional light on the behaviour of the two heuristics. Since the general principle of the construction is the same in higher dimension and can, hence, already be illustrated in 2D we restrict the following description to the case $d = 2$.

As depicted in the top row of Figure 1, the image consists of two grains G_1 and G_2 ; one (orange [light gray]) is the square $[0, \beta] \times [0, \beta]$ for different values of $\beta \in (0, 1)$, the other (blue [gray]) is its complement in $[0, 1] \times [0, 1]$. Of course, one would not expect such samples to originate



Figure 1. A family of grain scans involving a parameter β to illustrate differences between H1 and H2. Top, from left to right: Grain scans for $\beta = 0.1$, $\beta = 0.5$, and $\beta = 0.9$. Grain G_1 (orange [light gray]) fills the square $[0, \beta] \times [0, \beta]$, G_2 (blue [gray]) fills the rest of the unit square. The bottom row shows the computed ellipses $c_i + \mathbb{B}_{A_i}^2$ with $A_i = (\Sigma(G_i))^{-1}$ for $i = 1, 2$.

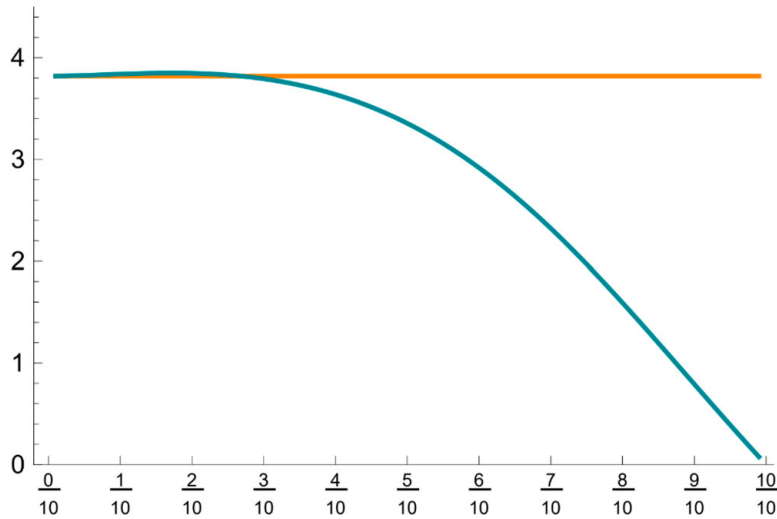


Figure 2. Ratios $v_i/\text{vol}(\mathbb{B}_{A_i}^2)$ of the areas of the grain and the corresponding ellipsoid for $i = 1$ (orange [light gray]) and $i = 2$ (blue [gray]) as a function of β .

from real-world grain scans. The bottom row of [Figure 1](#) shows the corresponding ellipses $c_i + \mathbb{B}_{A_i}^2$ with respect to $A_i = (\Sigma(G_i))^{-1}$.

As argued before, the variance of the ratios $v_i/\text{vol}(\mathbb{B}_{A_i}^2)$ is a relevant parameter. [Figure 2](#) therefore provides plots of

$$\frac{\beta^2}{\text{vol}(\mathbb{B}_{A_1}^2)} \quad \text{and} \quad \frac{1 - \beta^2}{\text{vol}(\mathbb{B}_{A_2}^2)}$$

as a function of β .

Note that these ratios differ considerably for larger values of β . Hence, we would expect that the heuristics H1 and H2 behave quite differently in these cases. This is confirmed by the resulting diagrams depicted in [Figure 3](#).



Figure 3. Diagrams for the family of grain scans from [Figure 1](#) obtained via H1 (top row) and H2 (bottom row).

The behaviour for $\beta = 0.9$ suggests that H1 tries to preserve the areas of the grains, while H2 seems to favour their shapes. If the relevant volume ratios are close, as is the case for small β , both heuristics achieve both aims and perform similarly.

Figure 4 shows the ratios of the grain volume v_i and the areas of the diagram cells under the two models H1 and H2, respectively.

5. Dynamic grain models

We now describe a simple interpolation/extrapolation model that is capable of quickly turning information acquired at a few moments in time into a continuous description of the grain growth process. To demonstrate the potential of the model, we then report on the results of an empirical evaluation on a real-world grain scan time series. Finally, we study for this empirical model the correlation between the diagram size parameters Γ and the disappearance of grains in the process.

5.1. The model

Since well-chosen diagrams have been demonstrated to yield high-quality representations of grain structures of many types of polycrystalline materials, it is natural to exploit their rather low-dimensional parameter space for deriving continuous models from data measured at discrete times. Recall that a diagram is fully characterized by the triple $(\mathcal{A}, S, \Gamma)$. In order to simulate the dynamic evolution of grain structures over time, we can fit a polynomial (or any other desired type of function) for each diagram parameter extracted from a discrete-time series polycrystal data set. These functions then represent the continuous change in diagram parameters during the growth process. Understanding the evolution of the diagram parameters over time may help predict changes in the material during an annealing process and may provide further insights into the material characteristics.

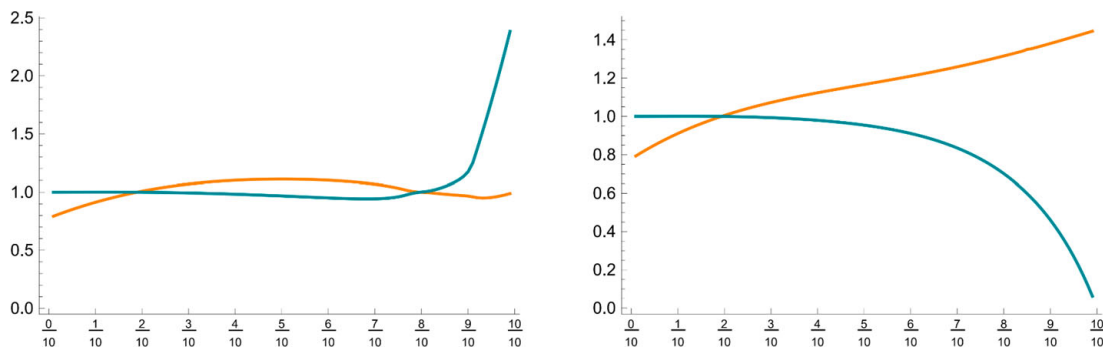


Figure 4. Ratio of the area v_i of the original grain and the area of the corresponding diagram cell obtained by H1 (left) and H2 (right) for $i = 1$ (orange [light gray]) and $i = 2$ (blue [gray]) as a function of β .

Suppose, that 3D-grain scans (or at least appropriate characteristic measurements) are available at the discrete times $t_0, t_1, \dots, t_m$. Then we compute, for each $t \in \{t_0, \dots, t_m\}$, the parameters of an anisotropic power diagram $\mathcal{P}_t = \mathcal{P}(\mathcal{A}_t, S_t, \Gamma_t)$ which represent the grain scans. Subsequently, we select a class $\mathcal{F}$ of parameterized functions and determine their parameters that best fit the data $\mathcal{A}_t, S_t, \Gamma_t, t \in \{t_0, \dots, t_m\}$. In our experimental setting we choose polynomials of a small degree to avoid overfitting, and we measure the fit in terms of least-squares deviation.

In 3D, the interpolation/extrapolation involves a class $\mathcal{F}$ consisting of 10 functions for each grain, three for the coordinates of the diagram sites, three for the rotation part of the grain's covariance matrix, three for the length of the principal axis of the covariance matrix, and one for the size parameter. Of course, in order to set up the computation of the functions, we need to 'track' each grain, i.e. we need to identify which grain corresponds to which at the different moments t in time.

During the grain growth process, grains may disappear. We assume, however, that no new grains emerge. In the experimental series that will be described in more detail below, about 40% of all initially present grains disappeared throughout the process. Hence we distinguish between such *non-persistent* grains and *persistent* grains that exist throughout the observed time interval. Of, course, for non-persistent grains, the functions are fitted only as long as the grain 'definitely exists'. More specifically, if a previously existing grain disappears from the measurement, we choose as last diagram parameters the covariance matrix from the previous time step and compute a size parameter such that its grain volume is essentially zero. If, on the other hand, for some moment in time our prediction generates a covariance matrix and a size parameter which result in a grain volume close to zero, we exclude the grain from further consideration.

The diagrams $\mathcal{P}_t$ can, in principle, be computed by any suitable method. In addition to the obvious criteria that any employed algorithm should be computationally fast and produce diagrams of appropriate accuracy, one has to pay attention to its behaviour concerning the invariances within the parameter space to avoid corruption of the interpolation/extrapolation. As we will be using the heuristic H1 described in Section 3, all parameters are, however, uniquely determined by the measurements, and we can hence, in the following, ignore this issue.

5.2. The data

For our experimental study, we use the data set from [6, 36] (details of the data processing chain, from experiment to grain scan, can be found there and, in particular, in their supplementary materials sections). The data set, obtained by a diffraction contrast tomography (DCT) [37, 38] experiment at beamline ID11 at the European Synchrotron Radiation Facility (ESRF), comprises 15

3D-grain scans (voxelized grain maps) of a 99.9% pure polycrystalline iron sample taken at different moments in time of an annealing process. More precisely, the material was first cold-rolled and annealed for 30 min at 700°C to fully recrystallize before it was first scanned (time t_0). Subsequently, the sample was annealed at a constant temperature of 800°C for 10 min leading to grain growth, cooled to stabilize, and scanned again (time t_1). This was repeated another 13 times with annealing periods of 5 min (time $t_2, \dots, t_{14}$). The effect of the growth process is already visible by the number of grains in the sample. While the first scan exhibits 1327 grains, the last one contains only 776, and the number of interior grains strictly decreases from 387 to 189; see [6, Table 1] for more details. The identification of grains over the 15 moments in time is already provided by in [6, 36].

For this data set, i.e. for each 3D-grain scan at $t \in \{t_0, t_1, \dots, t_{14}\}$, we computed an anisotropic power diagram $\mathcal{P}_t = \mathcal{P}(\mathcal{A}_t, S_t, \Gamma_t)$ representing the respective grain scan. For the computations, we used H1 as we were also interested in featuring the temporal development of the corresponding size parameters in view of the relation of H1 and H2 elaborated in Section 3.

5.3. Evaluation

We apply a cross-validation to assess the quality of the model. To compute the accuracy at time $t_\ell \in \{t_0, \dots, t_{14}\}$, we exclude the data for t_ℓ and fit the functions of $\mathcal{F}$ on the data for the remaining times $\{t_0, \dots, t_{14}\} \setminus \{t_\ell\}$. Then, by evaluating these functions at time t_ℓ , we obtain the diagram parameters $\mathcal{A}'_\ell, S'_\ell, \Gamma'_\ell$, and hence the diagram $\mathcal{P}'_\ell = \mathcal{P}(\mathcal{A}'_\ell, S'_\ell, \Gamma'_\ell)$. Computing $\Phi_G(\hat{C}(\mathcal{P}'_\ell))$, the accuracy of $\mathcal{P}'_\ell$ representing the 3D-grain scan G at t_ℓ , gives us finally the accuracy (of the model) at time t_ℓ . In other words, we perform a leave-one-out cross validation using (in all combinations) one observation for validation and 14 observations for fitting.

The results for polynomials of degree 1, 2, and 3 are depicted in Figure 5 in yellow [light gray], orange [gray], and red [dark gray], respectively. Let us point out that the piecewise linear curves which connect the 15 data points for each colour are only drawn for a better perception of the trends over time.

We observe that the degree of the employed polynomials does not seem to have a strong impact on the fit. It would be interesting to link this observation to the ‘uniform setup’ in terms of the actions over time of the underlying grain growth process.

As an additional reference the blue [black] markers give an upper ‘orientation’ for the accuracy that one may expect from the model. In fact, the blue [black] markers depict the accuracy of H1 at each of the 15 time steps. The curves are dashed at both ends to signify that the model changes from interpolation to extrapolation at these extremes. Also, note that the earlier annealing times differ from the latter ones.

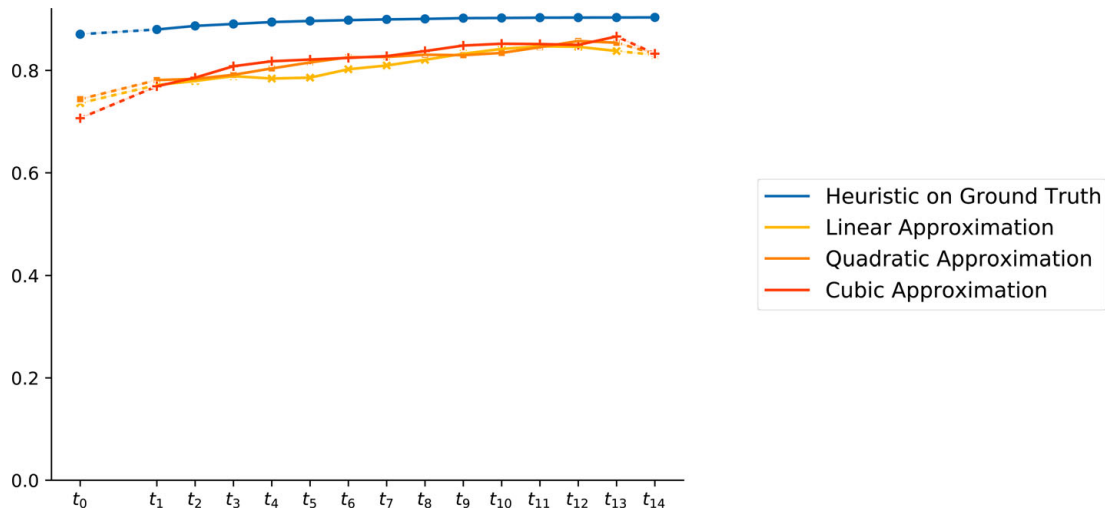


Figure 5. Accuracy (within the cross-validation paradigm) of the interpolation/extrapolation model plotted for each $t_0, \dots, t_{14}$ using polynomials of degree 1, 2, and 3, respectively for the fit.

Figure 5 shows that the accuracy of our models is generally high. With quadratic polynomials, we obtain an initial accuracy of 74%, which increases over time to approximately 86% for later time steps. Note that this is not much below the accuracy obtained by H1 on the ground truth. The general trend that accuracy increases with time (and hence prediction becomes easier) for this data set is in perfect agreement of the findings in [6, 36], which show that over time the system approaches the self-similar regime.

Figure 6 shows a fixed 2D slice of the 3D polycrystal at various moments in time and the diagrams resulting from our model (in the cross-validation scheme described above).

A discussion of the question how much the fit can be further improved by optimizing over all three sets of parameters that define the anisotropic power diagrams can be found in [31].

5.4. Size variation

In the remaining part of this section, we study how the growth process affects the size parameters Γ_t (recall that they need not correlate with grain sizes and they should not, at this point, be confused with often similarly denoted interfacial energy terms; we have provided an interpretation in terms of scaling factors in Equation (1)). As Figure 5 shows, the accuracy of the model increases with the duration of the process. This may be seen as an indication that the employed diagram structures are energetically favourable. Complementing the analysis of Section 4, we will now empirically study how the size parameters Γ of the diagrams change as the grains grow.

Let us begin by recalling the interpretation for the similarities and differences of H1 and H2. In fact, based on the analysis of Section 4 we suspected that for

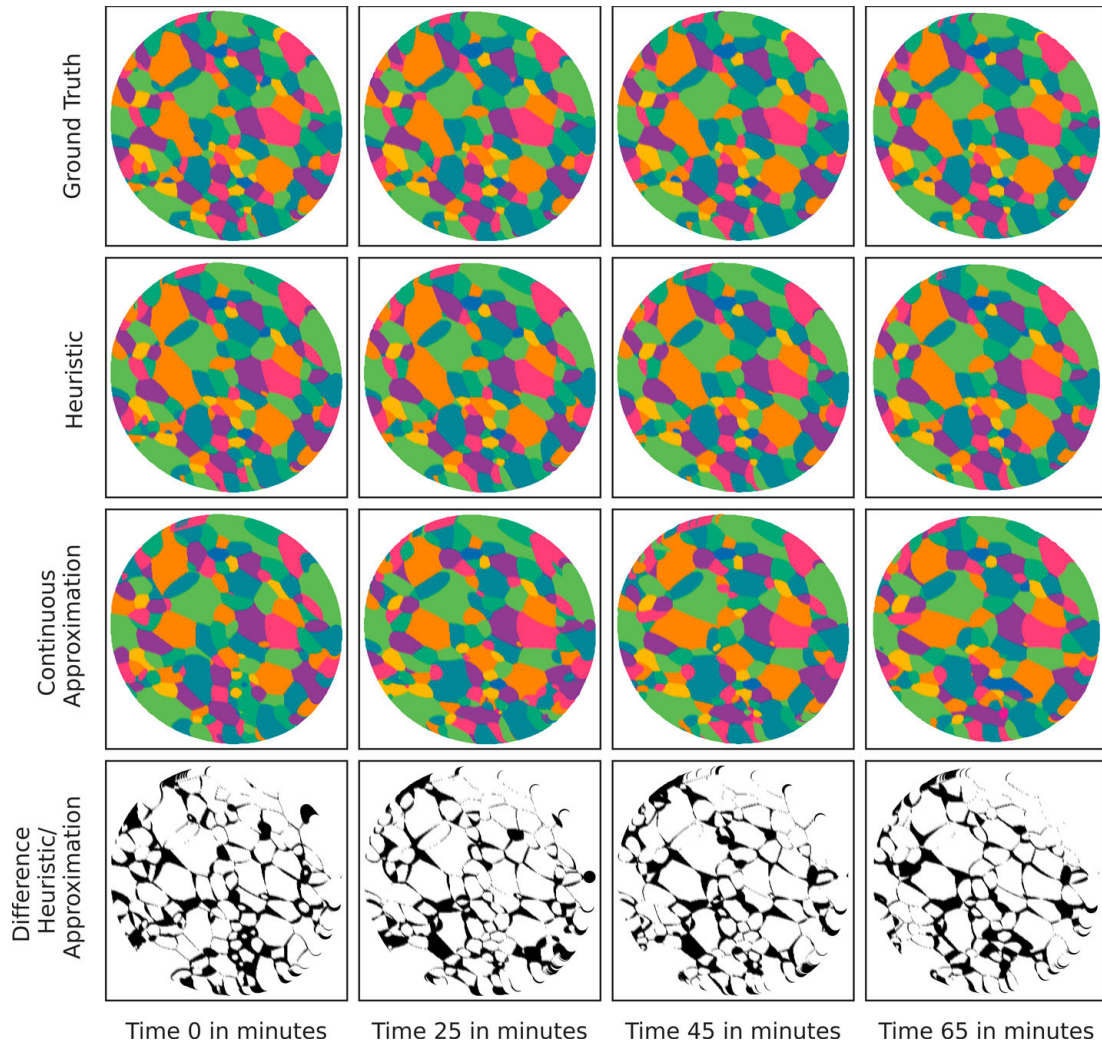


Figure 6. Visual comparison of diagrams obtained from full data (available for $t_0, \dots, t_{14}$) and the interpolation/extrapolation model that uses all data except for the respective moment in time. A fixed 2D slice through the 3D data set at times t_0, t_5, t_9 and t_{13} (top row); the corresponding diagram representation obtained via H1 (2nd row) and, respectively, via the interpolation/extrapolation model that uses only data from the other moments in time and best-fitting quadratic polynomials (3rd row); symmetric difference between the images from the 2nd and 3rd row (bottom row).

persistent grains the variances of the powers $\Gamma_t^{3/2}$ of the sizes chosen by H1 should be significantly smaller than for non-persistent grains (as the latter might not be energetically favourable). As Tables 2 and 3 show, these variances are indeed all small and significantly smaller for the persistent grains. This means, on the one hand, that the scaling factors α_i from Equation (1) do not vary much, and we might as well have employed the simpler heuristic H2.

Table 2. Variance of $\Gamma_t^{3/2}$ for $\ell \in \{0, 1, \dots, 14\}$ of persistent grains.

	$t_{\ell+0}$	$t_{\ell+1}$	$t_{\ell+2}$	$t_{\ell+3}$	$t_{\ell+4}$
$\ell = 0$	$6.409 \cdot 10^{-8}$	$6.214 \cdot 10^{-8}$	$5.130 \cdot 10^{-8}$	$3.016 \cdot 10^{-6}$	$2.544 \cdot 10^{-7}$
$\ell = 5$	$6.719 \cdot 10^{-6}$	$5.839 \cdot 10^{-6}$	$1.057 \cdot 10^{-5}$	$7.025 \cdot 10^{-4}$	$9.815 \cdot 10^{-7}$
$\ell = 10$	$2.286 \cdot 10^{-4}$	$7.425 \cdot 10^{-5}$	$5.749 \cdot 10^{-5}$	$1.185 \cdot 10^{-3}$	$1.823 \cdot 10^{-5}$

Table 3. Variance of $\Gamma_{t_\ell}^{3/2}$ for $\ell \in \{0, 1, \dots, 14\}$ of non-persistent grains.

	$t_{\ell+0}$	$t_{\ell+1}$	$t_{\ell+2}$	$t_{\ell+3}$	$t_{\ell+4}$
$\ell = 0$	$7.570 \cdot 10^{-3}$	$1.040 \cdot 10^{-2}$	$4.915 \cdot 10^{-3}$	$1.974 \cdot 10^{-2}$	$2.993 \cdot 10^{-3}$
$\ell = 5$	$1.278 \cdot 10^{-2}$	$1.834 \cdot 10^{-2}$	$2.790 \cdot 10^{-2}$	$4.158 \cdot 10^{-3}$	$8.959 \cdot 10^{-3}$
$\ell = 10$	$2.608 \cdot 10^{-2}$	$2.804 \cdot 10^{-2}$	$2.955 \cdot 10^{-2}$	$3.541 \cdot 10^{-3}$	–

On the other hand, in terms of the α_i , non-persistent grains are less ‘regular’ than persistent grains.

A closer look at Table 2 shows that, perhaps unexpectedly, the variance of $\Gamma_t^{3/2}$ does not decrease monotonously in time, not even for persistent grains. One might have anticipated that over time the behaviour of the volumes of the unit balls $\mathbb{B}_{A_i(t)}^3$ with respect to $A_i(t) = (\Sigma(G_i(t)))^{-1}$ become increasingly similar to that of the grain volumes $v_i(t)$. The actual behaviour might be explained by grain disappearance with neighbouring grains filling the newly available space such that it might require additional energy to reach a more regular mass distribution.

5.5. Grain disappearance

Next, we use a standard logistic regression model for predicting the probability $p_i(t)$ of the binary response variable which signifies whether grain i disappears (response value 1) at the next observed moment $t + 1$ or not (response value 0). As predictor variables we use the grain’s volume and its size parameter as computed by H1. Hence, we estimate three parameters, factors β_1 and β_2 for each of the two variables and one parameter β_0 for the intercept, i.e. we use the regression model

$$p_i(t) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 v_i(t) + \beta_2 \gamma_i(t))}}.$$

Note that the parameters β_0 , β_1 , and β_2 are independent of t . Hence, the model is identical for all time steps in the sense that it depends only on the grain characteristic. Our *null hypothesis* is that neither the size nor volume parameter correlate with grain disappearance. As a significance level we choose 0.05.

Table 4 depicts the computed coefficients and p -values. Let us, in addition, point out that the McFadden’s Pseudo R^2 value is 0.322, indicating a good model fit. As it turns out, we can reject the null hypothesis at a significance

Table 4. Coefficients estimated by maximum likelihood estimation and their p -values for the described logistic regression model.

	Coefficient	p -value
β_0 (intercept)	−4.0537	<0.0001
β_1 (volume)	−0.0010	<0.0001
β_2 (size parameter)	−1.0924	<0.0001

level of 0.05; the small p-value for β_2 suggests that the diagram size parameter Γ_t is statistically significant in this model for predicting grain disappearance.

We believe it is promising to investigate this further, not least because it agrees with recent experimental evidence from [39, 40], which demonstrates that the volume parameter alone is not a good predictor of grain growth (or shrinkage).

6. Final remarks

As we have seen, measurements at discrete times of a grain growth process can be turned into a continuous description of the process using diagram representations. Our findings about the increasing fit of anisotropic diagrams, the variance of $\Gamma_t^{3/2}$, and the relevance of the $\Gamma_t^{3/2}$ for predicting grain disappearance indicate that, in the course of the growth process, the grain scans tend towards particularly ‘simple’ diagrams with essentially uniform size parameters.

Since all computations are fast, and can, if necessary, generally even be speeded up by using H2 rather than H1, we believe that the described model has the potential of becoming a practical tool for simulation and further studies of grain growth in practice. A next step is to evaluate the model more extensively on various types of grain scan data. It would be desirable if a consolidating foundation of the model can be achieved based on first principles.

Finally, let us point out that the model allows for some flexibility to incorporate additional insight. It is possible, for instance, to incorporate available characteristics of grain growth into the curve fitting process.

Acknowledgments

We thank Henning Friis Poulsen for fruitful discussions and valuable comments on an earlier version of the present paper. We are also grateful to Henning Friis Poulsen and Allan Lyckegaard for providing the real-world data set used in this paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- [1] H.F. Poulsen, *Three-dimensional X-ray Diffraction Microscopy: Mapping Polycrystals and Their Dynamics*, Springer, Berlin, 2004.
- [2] A. Rollett, G.S. Rohrer, and J. Humphreys, *Recrystallization and Related Annealing Phenomena*, 3rd ed., Elsevier, Amsterdam, 2017.
- [3] C.S. Smith, *Grain shapes and other metallurgical applications of topology*, in *Metal Interfaces: a seminar on metal interfaces held during the 33rd National Metal*

- Congress and Exposition*, October 13-19, American Society for Metals, Detroit (Cleveland), 1952. pp. 65–108.
- [4] J. von Neumann, *Discussion of article [3] by C.S. Smith*, in *Metal Interfaces: a seminar on metal interfaces held during the 33rd National Metal Congress and Exposition*, October 13-19, American Society for Metals, Detroit (Cleveland), 1952. pp. 108–110.
 - [5] W.W. Mullins, *Two-dimensional motion of idealized grain boundaries*, J. Appl. Phys. 27 (1956), pp. 900–904.
 - [6] J. Zhang, Y. Zhang, W. Ludwig, D. Rowenhorst, P.W. Voorhees, and H.F. Poulsen, *Three-dimensional grain growth in pure iron. Part I. Statistics on the grain level*, Acta Mater. 156 (2018), pp. 76–85.
 - [7] M. Hillert, *On the theory of normal and abnormal grain growth*, Acta Metall. 13 (1965), pp. 227–238.
 - [8] R.D. MacPherson and D.J. Srolovitz, *The von Neumann relation generalized to coarsening of three-dimensional microstructures*, Nature 446 (2007), pp. 1053–1055.
 - [9] A. Bhattacharya, Y.-F. Shen, C.M. Hefferan, S.F. Li, J. Lind, R.M. Suter, C.E. Krill III, and G.S. Rohre, *Grain boundary velocity and curvature are not correlated in Ni polycrystals*, Science 374 (2021), pp. 189–193.
 - [10] J. Han, S.L. Thomas, and D.J. Srolovitz, *Grain-boundary kinetics: A unified approach*, Prog. Mater. Sci. 98 (2018), pp. 386–476.
 - [11] J.H.X. Wang, J. Zhang, J. Luo, Z. Zhang, and Z. Shen, *A general mechanism of grain growth-I. Theory*, J Materiomics 7 (2021), pp. 1007–1013.
 - [12] P.R. Rios and D. Zöllner, *Critical assessment 30: Grain growth – unresolved issues*, Mater. Sci. Technol. 34 (2018), pp. 629–638.
 - [13] D. Zöllner, P.R. Rios, and I. Zlotnikov, *Topological transitions: A topological random walk or pure geometric necessity?* Comput. Mater. Sci. 166 (2019), pp. 42–56.
 - [14] H.F. Poulsen and G.B.M. Vaughan, *Multigrain crystallography and three-dimensional grain mapping*, in *International Tables for Crystallography: Powder Diffraction*. Vol. H, International Union of Crystallography, Hoboken, 2019.
 - [15] A. Lyckegaard, E.M. Lauridsen, W. Ludwig, R.W. Fonda, and H.F. Poulsen, *On the use of Laguerre tessellations for representations of 3D grain structures*, Adv. Eng. Mater. 13 (2011), pp. 165–170.
 - [16] H. Telley, T.M. Liebling, and A. Mocellin, *The Laguerre model of grain growth in two dimensions I Cellular structures viewed as dynamical Laguerre tessellations*, Philos. Mag. B 73 (1996), pp. 409–427.
 - [17] H. Telley, T.M. Liebling, A. Mocellin, and F. Righetti, *Simulating and modelling grain growth as the motion of a weighted Voronoi diagram*, Mater. Sci. Forum 94-96 (1992), pp. 301–306.
 - [18] A. Alpers, A. Brieden, P. Gritzmam, A. Lyckegaard, and H.F. Poulsen, *Generalized balanced power diagrams for 3D representations of polycrystals*, Philos. Mag. 95 (2015), pp. 1016–1028.
 - [19] H. Altendorf, F. Latourte, D. Jeulin, M. Faessel, and L. Saintoyant, *3D reconstruction of a multiscale microstructure by anisotropic tessellation models*, Image Anal. Stereol. 33 (2014), p. 121.
 - [20] L. Petrich, O. Furat, M. Wang, C.E. Krill III, and V. Schmidt, *Efficient fitting of 3D tessellations to curved polycrystalline grain boundaries*, Front. Mater. 8 (2021), Article ID 760602.
 - [21] O. Šedivý, T. Brereton, D. Westhoff, L. Polívka, V. Beneš, V. Schmidt, and A. Jäger, *3D reconstruction of grains in polycrystalline materials using a tessellation model with curved grain boundaries*, Philos. Mag. 96 (2016), pp. 1926–1949.

- [22] O. Šedivý, J. Mullen Dake, C.E. Krill III, V. Schmidt, and A. Jäger, *Description of the 3D morphology of grain boundaries in aluminium alloys using tessellation models generated by ellipsoids*, Image Anal. Stereol. 36 (2017), p. 5.
- [23] A. Spetl, T. Brereton, Q. Duan, T. Werz, C.E. Krill III, D.P. Kroese, and V. Schmidt, *Fitting laguerre tessellation approximations to tomographic image data*, Philos. Mag. 96 (2016), pp. 166–189.
- [24] K. Teferra and L. Graham-Brady, *Tessellation growth models for polycrystalline microstructures*, Comput. Mater. Sci. 102 (2015), pp. 57–67.
- [25] K. Teferra and D.J. Rowenhorst, *Direct parameter estimation for generalised balanced power diagrams*, Philos. Mag. Lett. 98 (2018), pp. 79–87.
- [26] J. Barker, G. Bollerhey, and J. Hamaekers, *A multilevel approach to the evolutionary generation of polycrystalline structures*, Comput. Mater. Sci. 114 (2016), pp. 54–63.
- [27] D.P. Bourne, P.J.J. Kok, S.M. Roper, and W.D.T. Spanjer, *Laguerre tessellations and polycrystalline microstructures: A fast algorithm for generating grains of given volumes*, Philos. Mag. 100 (2020), pp. 2677–2707.
- [28] J. Guilleminot, A. Noshadravan, C. Soize, and R.G. Ghanem, *A probabilistic model for bounded elasticity tensor random fields with application to polycrystalline microstructures*, Comput. Methods Appl. Mech. Eng. 200 (2011), pp. 1637–1648.
- [29] M. Kühn and M.O. Steinhauser, *Modeling and simulation of microstructures using power diagrams: Proof of the concept*, Appl. Phys. Lett. 93 (2008), Article ID 034102.
- [30] T. Xu and M. Li, *Topological and statistical properties of a constrained Voronoi tessellation*, Philos. Mag. 89 (2009), pp. 349–374.
- [31] A. Alpers, M. Fiedler, P. Gritzmann, and F. Klemm, SIAM J. Imaging Sci. 16 (2023), pp. 223–249.
- [32] A. Brieden, P. Gritzmann, and F. Klemm, *Constrained clustering via diagrams: A unified theory and its application to electoral district design*, Eur. J. Oper. Res. 263 (2017), pp. 18–34.
- [33] P. Gritzmann and V. Klee, *Computational convexity*, in CRC Handbook on Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C.D. Toth, eds., 3rd extended ed., 2017, pp. 937–968.
- [34] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed., Springer, New York, 2009.
- [35] P.A. Lachenbruch and M. Goldstein, *Discriminant analysis*, Biometrics 35 (1979), p. 69.
- [36] J. Zhang, W. Ludwig, Y. Zhang, H.H.B. Sørensen, D.J. Rowenhorst, A. Yamanaka, P.W. Voorhees, and H.F. Poulsen, *Grain boundary mobilities in polycrystals*, Acta Mater. 191 (2020), pp. 211–220.
- [37] W. Ludwig, S. Schmidt, E.M. Lauridsen, and H.F. Poulsen, *X-ray diffraction contrast tomography: A novel technique for three-dimensional grain mapping of polycrystals. I. Direct beam case*, J. Appl. Crystallogr. 41 (2008), pp. 302–309.
- [38] G. Johnson, A. King, M.G. Honnicke, J. Marrow, and W. Ludwig, *X-ray diffraction contrast tomography: a novel technique for three-dimensional grain mapping of polycrystals. II. The combined case*, J. Appl. Crystallogr. 41 (2008), pp. 310–318.
- [39] A. Bhattachary, Y.-F. Shen, C.M. Hefferan, S. Fai Li, J. Lind, R.M. Suter, and G.S. Rohrer, *Three-dimensional observations of grain volume changes during annealing of polycrystalline Ni*, Acta Mater. 167 (2019), pp. 40–50.
- [40] V. Muralikrishnan, H. Liu, L. Yang, B. Conry, C.J. Marvel, M.P. Harmer, G.S. Rohrer, M.R. Tonks, R.M. Suter, C.E. Krill III, and A.R. Krause, *Observations of unexpected grain boundary migration in SrTiO₃*, Scr. Mater. 222 (2023), Article ID 115055.

Bibliography

- [Ach+20] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. *SLIC Superpixels*. <https://www.epfl.ch/labs/ivrl/research/slic-superpixels/>. 2020.
- [Ach+12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012), pp. 2274–2282.
- [AS17] R. Achanta and S. Ssstrunk. “Superpixels and Polygons Using Simple Non-iterative Clustering”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4895–4904.
- [AB09] M. R. Ackermann and J. Blmer. “Coresets and Approximate Clustering for Bregman Divergences”. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. Association for Computing Machinery (ACM), 2009, pp. 1088–1097.
- [AHPV05] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. “Geometric Approximation via Coresets”. In: *Combinatorial and Computational Geometry MSRI Publications* (2005), pp. 1–30.
- [AHPY08] P. K. Agarwal, S. Har-Peled, and H. Yu. “Robust Shape Fitting via Peeling and Grating Coresets”. In: *Discrete & Computational Geometry* 39.1-3 (2008), pp. 38–58.
- [ADK09] A. Aggarwal, A. Deshpande, and R. Kannan. “Adaptive Sampling for k-means Clustering”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 5687 LNCS. Springer, Berlin, Heidelberg, 2009, pp. 15–28.
- [AR14] C. C. Aggarwal and C. K. Reddy. *Data Clustering: Algorithms and Applications*. CRC Press, 2014.
- [AT17] E. Agustsson and R. Timofte. “NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* 2017-July (2017), pp. 1122–1131.

- [Alo+09] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. “NP-hardness of Euclidean Sum-of-squares Clustering”. In: *Machine Learning* 75.2 (2009), pp. 245–248.
- [Alp+15] A. Alpers, A. Brieden, P. Gritzmann, A. Lyckegaard, and H. F. Poulsen. “Generalized Balanced Power Diagrams for 3D Representations of Polycrystals”. In: *Philosophical Magazine* 95.9 (2015), pp. 1016–1028.
- [Alp+23a] A. Alpers, M. Fiedler, P. Gritzmann, and F. Klemm. “Dynamic grain models via fast heuristics for diagram representations”. In: *Philosophical Magazine* 103.10 (2023), pp. 948–968. arXiv: [2204.06430](#).
- [Alp+23b] A. Alpers, M. Fiedler, P. Gritzmann, and F. Klemm. “Turning Grain Maps into Diagrams”. In: *SIAM Journal on Imaging Sciences* 16.1 (2023), pp. 223–249. arXiv: [2205.06650](#).
- [Alt+14] H. Altendorf, F. Latourte, D. Jeulin, M. Faessel, and L. Saintoyant. “3D Reconstruction of a Multiscale Microstructure by Anisotropic Tessellation Models”. In: *Image Analysis & Stereology* 33.2 (2014), p. 121.
- [AMS07] S. Asharaf, M. N. Murty, and S. K. Shevade. “Multiclass Core Vector Machine”. In: *ACM International Conference Proceeding Series*. Vol. 227. New York, New York, USA: ACM Press, 2007, pp. 41–48.
- [Ass+19] S. Assadi, M. H. Bateni, A. Bernstein, V. Mirrokni, and C. Stein. “Coresets Meet EDCs: Algorithms for Matching and Vertex Cover on Massive Graphs”. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. Association for Computing Machinery, 2019, pp. 1616–1635. arXiv: [1711.03076](#).
- [Aur87] F. Aurenhammer. “Power Diagrams: Properties, Algorithms and Applications”. In: *SIAM Journal on Computing* 16.1 (1987), pp. 78–96.
- [AKL13] F. Aurenhammer, R. Klein, and D.-T. Lee. *Voronoi Diagrams and Delaunay Triangulations*. Singapore: World Scientific, 2013.
- [Awa+15] P. Awasthi, M. Charikar, R. Krishnaswamy, and A. K. Sinop. “The Hardness of Approximation of Euclidean k-means”. In: *Symposium on Computational Geometry* (2015), pp. 1–14.
- [BLK17] O. Bachem, M. Lucic, and A. Krause. “Practical Coreset Constructions for Machine Learning”. In: *arXiv* (2017). arXiv: [1703.06476](#).
- [BLC18] Z. Ban, J. Liu, and L. Cao. “Superpixel Segmentation Using Gaussian Mixture Model”. In: *IEEE Transactions on Image Processing* 27.8 (2018), pp. 4105–4117.

-
- [BFS24] S. Bandyapadhyay, F. V. Fomin, and K. Simonov. “On coresets for fair clustering in metric and Euclidean spaces and their applications”. In: *Journal of Computer and System Sciences* 142 (2024). arXiv: [2007.10137](#).
- [Ban08] J. Banhart, ed. *Advanced Tomographic Methods in Materials Research and Engineering*. Oxford University Press, Oxford, 2008.
- [BBH16] J. Barker, G. Bollerhey, and J. Hamaekers. “A Multilevel Approach to the Evolutionary Generation of Polycrystalline Structures”. In: *Computational Materials Science* 114 (2016), pp. 54–63.
- [BDW09] S. Basu, I. Davidson, and K. L. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. CRC Press, 2009, p. 472.
- [BL18] J.-P. Bauchet and F. Lafarge. “KIPPI: KInetic Polygonal Partitioning of Images”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 3146–3154.
- [Bay+18] C. Baykal, L. Liebenwein, I. Gilitschenski, D. Feldman, and D. Rus. *Data-dependent Coresets for Compressing Neural Networks with Applications to Generalization Bounds*. 2018. arXiv: [1804.05345](#).
- [Bā+02] M. Bădoiu, S. Har-Peled, P. Indyk, M. Bădoiu, S. Har-Peled, and P. Indyk. “Approximate Clustering via Core-sets”. In: *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing - STOC '02*. New York, New York, USA: ACM, 2002, pp. 250–257.
- [BDEL03] S. Ben-David, N. Eiron, and P. M. Long. “On the Difficulty of Approximately Maximizing Agreements”. In: *Journal of Computer and System Sciences* 66.3 (2003), pp. 496–514.
- [BWY06] J.-D. Boissonnat, C. Wormser, and M. Yvinec. “Curved Voronoi Diagrams”. In: *Effective Computational Geometry for Curves and Surfaces*. Springer Berlin Heidelberg, 2006, pp. 67–116.
- [BNS10] S. Boltz, F. Nielsen, and S. Soatto. “Earth Mover Distance on superpixels”. In: *IEEE International Conference on Image Processing*. 2010, pp. 4597–4600.
- [BBG17] S. Borgwardt, A. Brieden, and P. Gritzmann. “An LP-based k-means Algorithm for Balancing Weighted Point Sets”. In: *European Journal of Operational Research* 263.2 (2017), pp. 349–355. arXiv: [15334406](#).
- [BBG11] S. Borgwardt, A. Brieden, and P. Gritzmann. “Constrained Minimum-k-Star Clustering and its application to the consolidation of farmland”. In: *Operational Research* 11.1 (2011), pp. 1–17.

- [BBG14] S. Borgwardt, A. Brieden, and P. Gritzmann. “Geometric Clustering for the Consolidation of Farmland and Woodland”. In: *The Mathematical Intelligencer* 36.2 (2014), pp. 37–44.
- [Bou+20] D. P. Bourne, P. J. J. Kok, S. M. Roper, and W. D. T. Spanjer. “Laguerre Tessellations and Polycrystalline Microstructures: a Fast Algorithm for Generating Grains of Given Volumes”. In: *Philosophical Magazine* 100.21 (2020), pp. 2677–2707.
- [BDMI13] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. “Near-optimal Coresets for Least-squares Regression”. In: *IEEE Transactions on Information Theory* 59.10 (2013), pp. 6880–6892. arXiv: [1202.3505](#).
- [Bra+21] V. Braverman, D. Feldman, H. Lang, A. Statman, and S. Zhou. “Efficient Coreset Constructions via Sensitivity Sampling”. In: *Proceedings of The 13th Asian Conference on Machine Learning*. Vol. 157. PMLR, 2021, pp. 948–963.
- [Bra+16] V. Braverman, D. Feldman, H. Lang, A. Statman, and S. Zhou. “New Frameworks for Offline and Streaming Coreset Constructions”. In: *arXiv* (2016). arXiv: [1612.00889](#).
- [BMZ17] B. Brekhna, A. Mahmood, and Y. Zhou. “Robustness analysis of superpixel algorithms to image blur, additive Gaussian noise, and impulse noise”. In: *Journal of Electronic Imaging* 26.06 (2017), p. 1.
- [BG04] A. Brieden and P. Gritzmann. “A Quadratic Optimization Model for the Consolidation of Farmland by Means of Lend-Lease Agreements”. In: *Operations Research Proceedings 2003*. Springer, Berlin, Heidelberg, 2004, pp. 324–331.
- [BG10] A. Brieden and P. Gritzmann. “On Clustering Bodies: Geometry and Polyhedral Approximation”. In: *Discrete and Computational Geometry* 44.3 (2010), pp. 508–534.
- [BG12] A. Brieden and P. Gritzmann. “On Optimal Weighted Balanced Clusterings: Gravity Bodies and Power Diagrams”. In: *SIAM Journal on Discrete Mathematics* 26.2 (2012), pp. 415–434.
- [BG20] A. Brieden and P. Gritzmann. “Predicting Show Rates in Air Cargo Transport”. In: *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*. IEEE, 2020, pp. 1–9.
- [BG21] A. Brieden and P. Gritzmann. “Response Prediction: Gaining Reliable and Interpretable Insight from Small Study Data”. In: *submitted* (2021), p. 13.

-
- [Bri+98] A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovasz, and M. Simonovits. “Approximation of Diameters: Randomization Doesn’t Help”. In: *Proceedings 39th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc, 1998, pp. 244–251.
- [BGK17] A. Brieden, P. Gritzmann, and F. Klemm. “Constrained Clustering via Diagrams: A Unified Theory and its Application to Electoral District Design”. In: *European Journal of Operational Research* 263.1 (2017), pp. 18–34.
- [CG16] Y. Cai and X. Guo. “Anisotropic Superpixel Generation Based on Mahalanobis Distance”. In: *Computer Graphics Forum* 35.7 (2016), pp. 199–207.
- [CG11] G. D. Canas and S. J. Gortler. “Orphan-Free Anisotropic Voronoi Diagrams”. In: *Discrete & Computational Geometry* 46.3 (2011), pp. 526–541.
- [CCD16] J. G. Carlsson, E. Carlsson, and R. Devulapalli. “Shadow Prices in Territory Division”. In: *Networks and Spatial Economics* 16.3 (2016), pp. 893–931.
- [Che09] K. Chen. “On Coresets for k-Median and k-Means Clustering in Metric and Euclidean Spaces and Their Applications”. In: *SIAM Journal on Computing* 39.3 (2009), pp. 923–947.
- [Che+18] M. Chen, X. Wei, Q. Yang, Q. Li, G. Wang, and M.-H. Yang. “Spatiotemporal GMM for Background Subtraction with Superpixel Hierarchy”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1518–1525.
- [CSL20] Y.-Q. Chen, Z.-L. Sun, and K.-M. Lam. “An Effective Subsuperpixel-Based Approach for Background Subtraction”. In: *IEEE Transactions on Industrial Electronics* 67.1 (2020), pp. 601–609.
- [CXC14] Z. Chen, Y. Xiao, and J. Cao. “Approximation by piecewise polynomials on Voronoi tessellation”. In: *Graphical Models* 76.5 (2014), pp. 522–531.
- [Chi+17] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. “Fair Clustering through Fairlets”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 5036–5044.
- [Chi+13] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. 3rd. Chichester: Wiley, 2013.
- [Cla10] K. L. Clarkson. “Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm”. In: *ACM Transactions on Algorithms*. Vol. 6. 4. ACM PUB27 New York, NY, USA, 2010.

- [Coh+15] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu. “Dimensionality Reduction for k-Means Clustering and Low Rank Approximation”. In: *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. New York, NY, USA: ACM, 2015, pp. 163–172.
- [Das+08] A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney. “Sampling Algorithms and Coresets for lp Regression”. In: *SIAM Journal on Computing* 38.5 (2008), pp. 2060–2078. arXiv: [0707.1714](https://arxiv.org/abs/0707.1714).
- [Das07] S. Dasgupta. “The Hardness of k-means Clustering”. In: *Technical report CS2007-0890, University of California, San Diego* CS2008-091 (2007), p. 6.
- [DP21] A. Deshpande and R. Pratap. “Sampling-based Dimension Reduction for Subspace Approximation with Outliers”. In: *Theoretical Computer Science* 858 (2021), pp. 100–113.
- [Des+06] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. “Matrix Approximation and Projective Clustering via Volume Sampling”. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. Vol. 2. 1. Theory of Computing Exchange, 2006, pp. 1117–1126.
- [DZ16] L. Dong and J. Zhou. “Noise-robust SLIC Superpixel for Natural Images”. In: *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. IEEE, 2016, pp. 335–340.
- [Don+22] X. Dong, Y. Xiao, Z. Chen, J. Yao, and X. Guo. “GPU-based supervoxel segmentation for 3D point clouds”. In: *Computer Aided Geometric Design* 93 (2022), p. 102080.
- [Don+20] X. Dong, J. Han, D. Chen, J. Liu, H. Bian, Z. Ma, H. Li, X. Wang, W. Zhang, and N. Yu. “Robust Superpixel-Guided Attentional Adversarial Attack”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 12892–12901.
- [DL15] L. Duan and F. Lafarge. “Image Partitioning Into Convex Polygons”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3119–3127.
- [DL] L. Duan and F. Lafarge. *Varane Implementation*. <https://www.epfl.ch/labs/ivrl/research/slic-superpixels/>.
- [EZZ04] C. F. Eick, N. Zeidat, and Z. Zhao. “Supervised Clustering - Algorithms and Benefits”. In: *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* (2004), pp. 774–776.
- [Fav+20] J. D. Favreau, F. Lafarge, A. Bousseau, and A. Auvolet. “Extracting Geometric Structures in Images with Delaunay Point Processes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.4 (2020), pp. 837–850.

-
- [Fej72] L. Fejes Tóth. *Lagerungen in der Ebene, auf der Kugel und im Raum*. 2. extende. Berlin, Heidelberg, New York: Springer, 1972.
- [Fel+11] D. Feldman, M. Faulkner, A. Krause, and E. Zurich. *Scalable Training of Mixture Models via Coresets*. Tech. rep. 2011, pp. 2142–2150.
- [Fel+06] D. Feldman, A. Fiat, M. Sharir, and M. Sharif. *Coresets for weighted facilities and their applications*. Tech. rep. 1. New York, New York, USA, 2006, pp. 315–324.
- [FL11] D. Feldman and M. Langberg. “A unified framework for approximating and clustering data”. In: *Proceedings of the forty-third annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 2011, pp. 569–578. arXiv: [1106.1379](#).
- [Fel+10] D. Feldman, M. Monemizadeh, C. Sohler, and D. P. Woodruff. “Coresets and Sketches for High Dimensional Subspace Approximation Problems”. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. Association for Computing Machinery (ACM), 2010, pp. 630–649.
- [FSS13] D. Feldman, M. Schmidt, and C. Sohler. “Turning Big Data Into Tiny Data: Constant-Size Coresets for k-means, PCA and Projective Clustering”. In: *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. Vol. 49. 3. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2013, pp. 1434–1453. arXiv: [1807.04518](#).
- [FSS20] D. Feldman, M. Schmidt, and C. Sohler. “Turning Big Data Into Tiny Data: Constant-Size Coresets for k-Means, PCA, and Projective Clustering”. In: *SIAM Journal on Computing* 49.3 (2020), pp. 601–657. arXiv: [1807.04518](#).
- [Fen+19] Q. Feng, J. Hu, N. Huang, and J. Wang. “Improved PTAS for the Constrained k-means Problem”. In: *Journal of Combinatorial Optimization* 37.4 (2019), pp. 1091–1110.
- [Fic+13] H. Fichtenberger, M. Gillé, M. Schmidt, C. Schwiegelshohn, and C. Sohler. “BICO: BIRCH Meets Coresets for k-means Clustering”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8125 LNCS. Springer, Berlin, Heidelberg, 2013, pp. 481–492.
- [FA20] M. Fiedler and A. Alpers. “Power-SLIC: Fast Superpixel Segmentations by Diagrams”. 2020. arXiv: [2012.11772](#).
- [FG22a] M. Fiedler and P. Gritzmann. “Coresets for Weight-Constrained Anisotropic Assignment and Clustering”. In: *arXiv* (2022). arXiv: [2203.10864](#).
- [FG22b] M. Fiedler and P. Gritzmann. “On the Stability of Weight-Balanced Anisotropic Clustering”. 2022.

- [FGK22] M. Fiedler, P. Gritzmann, and F. Klemm. “On resolution coresets for constrained clustering”. In: *arXiv* (2022). arXiv: [2204.00893](https://arxiv.org/abs/2204.00893).
- [FKF16] J. Forsythe, V. Kurlin, and A. Fitzgibbon. “Resolution-Independent Superpixels Based on Convex Constrained Meshes Without Small Angles”. In: *Advances in Visual Computing. ISVC 2016*. Springer, 2016, pp. 223–233.
- [FS06] G. Frahling and C. Sohler. “A Fast k-means Implementation Using Coresets”. In: *Proceedings of the Annual Symposium on Computational Geometry*. Vol. 2006. Association for Computing Machinery (ACM), 2006, pp. 135–143.
- [FS05] G. Frahling and C. Sohler. “Coresets in Dynamic Geometric Data Streams”. In: *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing - STOC '05*. New York, New York, USA: ACM Press, 2005, p. 209.
- [FLF15] O. Freifeld, Y. Li, and J. W. Fisher. “A fast method for inferring high-quality simply-connected superpixels”. In: *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 2184–2188.
- [FVS09] B. Fulkerson, A. Vedaldi, and S. Soatto. “Class Segmentation and Object Localization With Superpixel Neighborhoods”. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 670–677.
- [GMW07] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, 2007.
- [Gou+08] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. “Multi-Class Segmentation with Relative Location Prior”. In: *International Journal of Computer Vision* 80.3 (2008), pp. 300–316.
- [Gou+14] S. Gould, J. Zhao, X. He, and Y. Zhang. “Superpixel Graph Label Transfer with Learned Distance Metric”. In: *Computer Vision – ECCV 2014*. Springer, 2014, pp. 632–647.
- [GK17] P. Gritzmann and V. Klee. “Computational Convexity”. In: *CRC Handbook on Discrete and Computational Geometry, 3rd extended ed., (Eds. J.E. Goodman, J. O’Rourke and C.D. Toth)*. 2017, pp. 937–968.
- [Gui+11] J. Guilleminot, A. Noshadran, C. Soize, Ghanem, and R. Ghanem. “A Probabilistic Model for Bounded Elasticity Tensor Random Fields With Application to Polycrystalline Microstructures”. In: *Computer Methods in Applied Mechanics and Engineering* 200.17-20 (2011), pp. 1637–1648.
- [Gur20] L. L. C. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2020.

-
- [HTS18] J. Han, S. L. Thomas, and D. J. Srolovitz. “Grain-Boundary Kinetics: A Unified Approach”. In: *Progress in Material Science* 98 (2018), pp. 386–476.
- [HZX21] Q. Han, N. Zhao, and J. Xu. “Recognition and location of steel structure surface corrosion based on unmanned aerial vehicle images”. In: *Journal of Civil Structural Health Monitoring* 11.5 (2021), pp. 1375–1392.
- [Han+09] P. C. Hansen, H. O. Sørensen, Z. Sükösd, and H. F. Poulsen. “Reconstruction of Single-Grain Orientation Distribution Functions for Crystalline Materials”. In: *SIAM Journal on Imaging Sciences* 2.2 (2009), pp. 593–613.
- [HPK07] S. Har-Peled and A. Kushal. “Smaller Coresets for k-Median and k-Means Clustering”. In: *Discrete & Computational Geometry* 37.1 (2007), pp. 3–19.
- [HPM04] S. Har-Peled and S. Mazumdar. “On Coresets for k-means and k-median Clustering”. In: *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing - STOC '04*. New York, New York, USA: ACM Press, 2004, p. 291.
- [HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2nd editio. Springer Series in Statistics. New York, NY: Springer New York, 2009.
- [Hil65] M. Hillert. “On the Theory of Normal and Abnormal Grain Growth”. In: *Acta Metallurgica* 13 (1965), pp. 227–238.
- [Hu+21] J. Hu, X. Wang, J. Zhang, J. Luo, Z. Zhang, and Z. Shen. “A General Mechanism of Grain Growth -I. Theory”. In: *Journal of Materiomics* 7 (2021), pp. 1007–1013.
- [HJV19] L. Huang, S. H. C. Jiang, and N. K. Vishnoi. “Coresets for Clustering with Fairness Constraints”. In: *Advances in Neural Information Processing Systems* 32 (2019). arXiv: [1906.08484](https://arxiv.org/abs/1906.08484).
- [Hua+18] L. Huang, S. H. Jiang, J. Li, and X. Wu. “Epsilon-Coresets for Clustering (with Outliers) in Doubling Metrics”. In: *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*. Vol. 2018-Octob. IEEE Computer Society, 2018, pp. 814–825.
- [HV20] L. Huang and N. K. Vishnoi. “Coresets for clustering in Euclidean spaces: Importance Sampling is Nearly Optimal”. In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. New York, NY, USA: ACM, 2020, pp. 1416–1429.
- [Jam+13] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. Vol. 103. Springer Texts in Statistics. New York, NY: Springer New York, 2013.

- [Jam+18] V. Jampani, D. Sun, M. Y. Liu, M. H. Yang, and J. Kautz. “Superpixel sampling networks”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11211 LNCS (2018), pp. 363–380. arXiv: [1807.10174](https://arxiv.org/abs/1807.10174).
- [Jia+14] Jianbing Shen, Yunfan Du, Wenguan Wang, and Xuelong Li. “Lazy Random Walks for Superpixel Segmentation”. In: *IEEE Transactions on Image Processing* 23.4 (2014), pp. 1451–1462.
- [JX12] Jie Wang and Xiaoqiang Wang. “VCells: Simple and Efficient Superpixels Using Edge-Weighted Centroidal Voronoi Tessellations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.6 (2012), pp. 1241–1247.
- [KZM20] X. Kang, L. Zhu, and A. Ming. “Dynamic Random Walk for Superpixel Segmentation”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 3871–3884.
- [Kan+04] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. “A Local Search Approximation Algorithm for k-means Clustering”. In: *Computational Geometry: Theory and Applications* 28.2-3 SPEC. ISS. (2004), pp. 89–112.
- [Kin84] L. J. King. *Central Place Theory*. Scientific geography series. SAGE Publications, 1984.
- [Kle20] F. Klemm. “Constrained Clustering via Generalized Voronoi Diagrams”. PhD thesis. Technische Universität München, 2020.
- [KS08] M. Kühn and M. O. Steinhauser. “Modeling and Simulation of Microstructures Using Power Diagrams: Proof of the Concept”. In: *Applied Physics Letters* 93.3 (2008), p. 034102.
- [KM20] V Kurlin and G Muszynski. “Persistence-based Resolution-Independent Meshes of Superpixels”. In: *Pattern Recognit. Lett.* 131.1 (2020), pp. 300–306.
- [LS03] F. Labelle and J. R. Shewchuk. “Anisotropic Voronoi Diagrams and Guaranteed-Quality Anisotropic Mesh Generation”. In: *Proceedings of the nineteenth annual symposium on Computational geometry*. New York, NY, USA: ACM, 2003, pp. 191–200.
- [LG79] P. A. Lachenbruch and M. Goldstein. “Discriminant Analysis”. In: *Biometrics* 35.1 (1979), p. 69.
- [Lev+09] A Levinshtein, A Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K Siddiqi. “TurboPixels: Fast Superpixels Using Geometric Flows”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.12 (2009), pp. 2290–2297.

-
- [Li+16] X Li, H He, R Wang, and J Cheng. “Superpixel-Guided Nonlocal Means for Image Denoising and Super-Resolution”. In: *Signal Process.* 124.6 (2016), pp. 173–183.
- [LC15] Z Li and J Chen. “Superpixel Segmentation using Linear Spectral Clustering”. In: *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1356–1363.
- [Liu+11] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. “Entropy rate superpixel segmentation”. In: *CVPR 2011*. IEEE, 2011, pp. 2097–2104.
- [Liu+16] Y.-J. Liu, C.-C. Yu, M. M.-J. Yu, and Y. He. “Manifold SLIC: A Fast Method to Compute Content-Sensitive Superpixels”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 40. 3. IEEE, 2016, pp. 651–659.
- [Liu+18] Y.-J. Liu, M. Yu, B.-J. Li, and Y. He. “Intrinsic Manifold SLIC: A Simple and Efficient Method for Computing Content-Sensitive Superpixels”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3 (2018), pp. 653–666.
- [Llo82] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137.
- [Lyc+11] A. Lyckegaard, E. M. Lauridsen, W. Ludwig, R. W. Fonda, and H. F. Poulsen. “On the Use of Laguerre Tessellations for Representations of 3D Grain Structures”. In: *Advanced Engineering Materials* 13.3 (2011), pp. 165–170.
- [LP11] A. Lyckegaard and H. F. Poulsen. “Data from grain Scan Measurements”. In: (2011).
- [Ma+21] D. Ma, Y. Zhou, S. Xin, and W. Wang. “Convex and Compact Superpixels by Edge-Constrained Centroidal Power Diagram”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 1825–1839.
- [MS07] R. D. MacPherson and D. J. Srolovitz. “The von Neumann Relation Generalized to Coarsening of Three-Dimensional Microstructures”. In: *Nature* 446 (2007), pp. 1053–1055.
- [Mac67] J. B. MacQueen. “Some methods of classification and analysis of multivariate observations”. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967, pp. 281–297.
- [MNV12] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. “The Planar k-means Problem is NP-hard”. In: *Theoretical Computer Science*. Vol. 442. Elsevier, 2012, pp. 13–21.

- [Mai+18] G. Maierhofer, D. Heydecker, A. I. Aviles-Rivero, S. M. Alsaleh, and C.-B. Schonlieb. “Peekaboo-Where are the Objects? Structure Adjusting Superpixels”. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3693–3697.
- [MMR19] K. Makarychev, Y. Makarychev, and I. Razenshteyn. “Performance of Johnson-Lindenstrauss transform for k-means and k-medians clustering”. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. New York, NY, USA: ACM, 2019, pp. 1027–1038. arXiv: [1811.03195](#).
- [MRR21] S. R. S. P. Malladi, S. Ram, and J. J. Rodriguez. “Image Denoising Using Superpixel-Based PCA”. In: *IEEE Transactions on Multimedia* 23 (2021), pp. 2297–2309.
- [Mar+01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. “A Database of Human Segmented Natural images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics”. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. IEEE Comput. Soc, 2001, pp. 416–423.
- [Mat02] J. Matoušek. *Lectures on Discrete Geometry*. Ed. by J. Matoušek. Vol. 212. Graduate Texts in Mathematics. New York, NY: Springer New York, 2002.
- [MG15] M. Menze and A. Geiger. “Object Scene Flow for Autonomous Vehicles”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3061–3070.
- [MSF12] O. V. Mishin, V. M. Segal, and S. Ferrasse. “Quantitative Microstructural Characterization of Thick Aluminum Plates Heavily Deformed Using Equal Channel Angular Extrusion”. In: *Metallurgical and Materials Transactions A* 43 (2012), pp. 4767–4776.
- [Mor05] G. Mori. “Guiding model search using segmentation”. In: *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*. IEEE, 2005, 1417–1423 Vol. 2.
- [Mul56] W. W. Mullins. “Two-dimensional Motion of idealized Grain Boundaries”. In: *Journal of Applied Physics* 27 (1956), pp. 900–904.
- [Mun+18] A. Munteanu, C. Sohler, C. Schwiegelshohn, and D. P. Woodruff. “On Coresets for Logistic Regression”. In: *Advances in Neural Information Processing Systems*. Vol. 2018-Decem. 2018, pp. 6561–6570. arXiv: [1805.08571](#).

-
- [Mus+22] B. Mussay, D. Feldman, S. Zhou, V. Braverman, and M. Osadchy. “Data-Independent Structured Pruning of Neural Networks via Coresets”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2022), pp. 7829–7841. arXiv: [2008.08316](#).
- [Neu52] J. von Neumann. “Discussion: Shape of Metal Grains”. In: *Metal Interfaces* (1952), pp. 108–110.
- [Now33] W. Nowacki. “Der Begriff „Voronischer Bereich“”. In: *Zeitschrift für Kristallographie - Crystalline Materials* 85.1-6 (1933), pp. 331–332.
- [NOW76] W. NOWACKI. “Über Allgemeine Eigenschaften von Wirkungsbereichen”. In: *Zeitschrift für Kristallographie - Crystalline Materials* 143.1-6 (1976), pp. 360–386.
- [Oka+00] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations*. Ed. by A. Okabe, B. Boots, K. Sugihara, S. N. Chiu, and D. G. Kendall. 2nd. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2000.
- [Pan+22] X. Pan, Y. Zhou, Y. Zhang, and C. Zhang. “Fast Generation of Superpixels With Lattice Topology”. In: *IEEE Transactions on Image Processing* 31 (2022), pp. 4828–4841.
- [Pet+21] L. Petrich, O. Furat, M. Wang, Krill III Carl E., and V. Schmidt. “Efficient Fitting of 3D Tessellations to Curved Polycrystalline Grain Boundaries”. In: *Frontiers in Materials* 8 (2021), p. 760602.
- [Phi17] J. M. Phillips. “Coresets and sketches”. In: *Handbook of Discrete and Computational Geometry, Third Edition* (2017), pp. 1269–1288. arXiv: [1601.00617](#).
- [PT18] J. M. Phillips and W. M. Tai. “Improved Coresets for Kernel Density Estimates”. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. Association for Computing Machinery, 2018, pp. 2718–2727. arXiv: [1710.04325](#).
- [Pou04] H. F. Poulsen. *Three-Dimensional X-Ray Diffraction Microscopy: Mapping Polycrystals and their Dynamics*. Springer, Berlin, 2004.
- [PV19] H. F. Poulsen and G. B. M. Vaughan. “Multigrain Crystallography and Three-Dimensional Grain Mapping”. In: *International Tables for Crystallography: Powder Diffraction*. Vol. H. International Union of Crystallography, Hoboken, 2019.
- [RM03] Ren and Malik. “Learning a Classification Model for Segmentation”. In: *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, 10–17 vol.1.

- [RZ18] P. R. Rios and D. Zöllner. “Critical Assessment 30: Grain Growth - Unresolved Issues”. In: *Material Science and Technology* 34 (2018), pp. 629–638.
- [Roe+16] J. Roels, J. De Vylder, J. Aelterman, S. Lippens, Y. Saeys, and W. Philips. “Superpixel Quality in Microscopy Images: The Impact of Noise & Denoising”. In: *Proc. XIV Mediterranean Conf. Medical and Biological Engineering and Computing (MEDICON)*. 2016, pp. 258–263.
- [RRH04] A. Rollett, G. S. Rohrer, and J. Humphreys. *Recrystallization and Related Annealing Phenomena*. 3rd editio. Elsevier, 2004.
- [Sai82] N. Saitô. “Asymptotic Regular Pattern of Epidermal Cells in Mammalian Skin”. In: *Journal of Theoretical Biology* 95.3 (1982), pp. 591–599.
- [SFS12] A. Schick, M. Fischer, and R. Stiefelhagen. “Measuring and evaluating the compactness of superpixels”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012, pp. 930–934.
- [SSS20] M. Schmidt, C. Schwiegelshohn, and C. Sohler. “Fair Coresets and Streaming Algorithms for Fair k-means”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11926 LNCS (2020), pp. 232–251.
- [SSS18] M. Schmidt, C. Schwiegelshohn, and C. Sohler. “Fair Coresets and Streaming Algorithms for Fair k-Means Clustering”. In: *arXiv* (2018). arXiv: [1812.10854](https://arxiv.org/abs/1812.10854).
- [SS02] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.
- [Sch86] A. Schrijver. *Theory of Linear and Integer Programmin*. Wiley & Sons, Chichester: Springer, 1986.
- [Šed+16] O. O. Šedivý, T. Brereton, D. Westhoff, L. Polívka, V. Beneš, V. Schmidt, and A. Jäger. “3D Reconstruction of Grains in Polycrystalline Materials Using a Messellation Model With Curved Grain Boundaries”. In: *Philosophical Magazine* 96.18 (2016), pp. 1926–1949.
- [Šed+17] O. Šedivý, J. M. Dake, C. E. Krill III, V. Schmidt, and A. Jäger. “Description of the 3D Morphology of Grain Boundaries in Aluminium Alloys Using Tessellation Models Generated by Ellipsoids”. In: *Image Analysis & Stereology* 36.1 (2017), p. 5.
- [SARS20] P. Sellars, A. I. Aviles-Rivero, and C. Schönlieb. “Superpixel Contracted Graph-Based Learning for Hyperspectral Image Classification”. In: *IEEE Trans. Geosci. Remote Sens.* 58.6 (2020), pp. 4180–4193.

-
- [Sha+95] M. Sharir, P. K. Agarwal, Sharir Micha, and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [Shu+11] Shu Wang, Huchuan Lu, Fan Yang, and M.-H. Yang. “Superpixel tracking”. In: *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1323–1330.
- [Smi52] C. S. Smith. “Grain Shapes and Other Metallurgical Applications of Topology”. In: *Metal Interfaces* (1952), pp. 65–133.
- [SW18] C. Sohler and D. P. Woodruff. “Strong Coresets for k-Median and Subspace Approximation: Goodbye Dimension”. In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018, pp. 802–813. arXiv: [1809.02961](#).
- [Spe+16] A. Spetttl, T. Brereton, Q. Duan, T. Werz, C. E. Krill, D. P. Kroese, and V. Schmidt. “Fitting Laguerre Tessellation Approximations to Tomographic Image Data”. In: *Philosophical Magazine* 96.2 (2016), pp. 166–189.
- [Spe+14] A. Spetttl, T. Werz, C. E. Krill, and V. Schmidt. “Parametric Representation of 3D Grain Ensembles in Polycrystalline Microstructures”. In: *Journal of Statistical Physics* 154.4 (2014), pp. 913–928.
- [SHL18] D. Stutz, A. Hermans, and B. Leibe. “Superpixels: An Evaluation of the State-of-the-art”. In: *Computer Vision and Image Understanding* 166 (2018), pp. 1–27.
- [Suz20] T. Suzuki. “Superpixel Segmentation Via Convolutional Neural Networks with Regularized Information Maximization”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 2020-May. IEEE, 2020, pp. 2573–2577. arXiv: [2002.06765](#).
- [TGB15] K. Teferra and L. Graham-Brady. “Tessellation Growth Models for Polycrystalline Microstructures”. In: *Computational Materials Science* 102 (2015), pp. 57–67.
- [TR18] K. Teferra and D. J. Rowenhorst. “Direct Parameter Estimation for Generalised Balanced Power Diagrams”. In: *Philosophical Magazine Letters* 98.2 (2018), pp. 79–87.
- [Tel+92] H. Telley, T. Liebling, A. Mocellin, and F. Righetti. “Simulating and Modelling Grain Growth as the Motion of a Weighted Voronoi Diagram”. In: *Materials Science Forum* 94-96.1 (1992), pp. 301–306.
- [Tel+96] H. Telley, T. m. Libeling, A. A. Mocellin, T. M. Liebling, and A. A. Mocellin. “The Laguerre Model of Grain Growth in two Dimensions II. Examples of Coarsening Simulations”. In: *Philosophical Magazine B* 73.3 (1996), pp. 409–427.

- [TKH05] I. W. Tsang, J. T. Kwok, and P. U. Hk. *Core Vector Machines: Fast SVM Training on Very Large Data Sets* Pak-Ming Cheung. Tech. rep. 2005, pp. 363–392.
- [TKZ06] I. W. H. Tsang, J. T. Y. Kwok, and J. M. Zurada. “Generalized Core Vector Machines”. In: *IEEE Transactions on Neural Networks* 17.5 (2006), pp. 1126–1140.
- [Tu+18] W.-C. Tu, M.-Y. Liu, V. Jampani, D. Sun, S.-Y. Chien, M.-H. Yang, and J. Kautz. “Learning Superpixels with Segmentation-Aware Affinity Loss”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 568–576.
- [URF19] R. Uziel, M. Ronen, and O. Freifeld. “Bayesian Adaptive Superpixel Segmentation”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 8469–8478.
- [Van+15] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool. “SEEDS: Superpixels Extracted Via Energy-Driven Sampling”. In: *International Journal of Computer Vision* 111.3 (2015), pp. 298–314.
- [Wan+17] M. Wang, X. Liu, Y. Gao, X. Ma, and N. Q. Soomro. “Superpixel segmentation: A benchmark”. In: *Signal Processing: Image Communication* 56.April (2017), pp. 28–39.
- [Wor08] C. Wormser. “Generalized Voronoi Diagrams and Applications”. PhD thesis. Université Nice Sophia Antipolis, 2008.
- [XL09] T. Xu and M. Li. “Topological and Statistical Properties of a Constrained Voronoi Tessellation”. In: *Philosophical Magazine* 89.4 (2009), pp. 349–374.
- [Yan+15] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li. “Object detection by labeling superpixels”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 5107–5116.
- [Yan+20] F. Yang, Q. Sun, H. Jin, and Z. Zhou. “Superpixel Segmentation With Fully Convolutional Networks”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 13961–13970. arXiv: [2003.12929](https://arxiv.org/abs/2003.12929).
- [Ye+19] Z. Ye, R. Yi, M. Yu, Y.-J. Liu, and Y. He. “Fast Computation of Content-Sensitive Superpixels and Supervoxels Using Q-Distances”. In: *2019 IEEE / CVF International Conference on Computer Vision (ICCV)*. Vol. 2019-Octob. IEEE, 2019, pp. 3769–3778.
- [Zha+21] J. Zhang, A. I. Aviles-Rivero, D. Heydecker, X. Zhuang, R. Chan, and C.-B. Schönlieb. “Dynamic spectral residual superpixels”. In: *Pattern Recognition* 112 (2021), p. 107705.

-
- [Zha+18] J. Zhang, Y. Zhang, W. Ludwig, D. Rowenhorst, P. W. Voorhees, and H. F. Poulsen. “Three-dimensional Grain Growth in Pure Iron. Part I. Statistics on the Grain Level”. In: *Acta Materialia* 156 (2018), pp. 76–85.
- [ZGZ20] Y. Zhang, Q. Guo, and C. Zhang. “Simple and Fast Image Superpixels Generation With Color and Boundary Probability”. In: *The Visual Computer* (2020).
- [ZCL19] Z. Zhang, J. Chen, and Z. Liu. “SLIC segmentation method for full-polarised remote-sensing image”. In: *The Journal of Engineering* 2019.20 (2019), pp. 6404–6407.
- [ZP17] Y. Zheng and J. M. Phillips. “Coresets for Kernel Regression”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Vol. Part F1296. New York, NY, USA: Association for Computing Machinery, 2017, pp. 645–654. arXiv: [1702.03644](#).
- [Zhu+21] L. Zhu, Q. She, B. Zhang, Y. Lu, Z. Lu, D. Li, and J. Hu. “Learning the Superpixel in a Non-iterative and Lifelong Manner”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2021), pp. 1225–1234. arXiv: [2103.10681](#).
- [ZK07] C. L. Zitnick and S. B. Kang. “Stereo for Image-Based Rendering using Image Over-Segmentation”. In: *International Journal of Computer Vision* 75.1 (2007), pp. 49–65.