New Methods for Applying Topological Data Analysis to Materials Science

by

Teresa Heiss

December, 2024

A thesis submitted to the
Graduate School
of the
Institute of Science and Technology Austria
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Committee in charge:

Maria Ibáñez, Chair
Herbert Edelsbrunner
Uli Wagner
Dmitriy Morozov

The thesis of Teresa Heiss, titled *New Methods for Applying Topological Data Analysis to Materials Science*, is approved by:

Supervisor: Herbert Edelsbrunner, ISTA, Klosterneuburg, Austria

Signature: _____

Committee Member: Uli Wagner, ISTA, Klosterneuburg, Austria

Signature: _____

Committee Member: Dmitriy Morozov, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Signature: _____

Defense Chair: Maria Ibáñez, ISTA, Klosterneuburg, Austria

Signature: _____

Signed page is on file

ISTA Thesis, ISSN: 2663-337X

ISBN: 978-3-99078-052-7

I hereby declare that this thesis is my own work and that it does not contain other people's work without this being so stated; this thesis does not contain my previous work without this being stated, and the bibliography contains all the literature that I used in writing the dissertation.

I accept full responsibility for the content and factual accuracy of this work, including the data and their analysis and presentation, and the text and citation of other work.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other university or institution.

I certify that any republication of materials presented in this thesis has been approved by the relevant publishers and co-authors.

Signature: _____

Teresa Heiss
December, 2024

Signed page is on file

Abstract

Many chemical and physical properties of materials are determined by the material's shape, for example the size of its pores and the width of its tunnels. This makes materials science a prime application area for geometrical and topological methods. Nevertheless many methods in topological data analysis have not been satisfyingly extended to the needs of materials science. This thesis provides new methods and new mathematical theorems targeted at those specific needs by answering four different research questions. While the motivation for each of the research questions arises from materials science, the methods are versatile and can be applied in different areas as well.

The first research question is concerned with image data, for example a three-dimensional computed tomography (CT) scan of a material, like sand or stone. There are two commonly used topologies for digital images and depending on the application either of them might be required. However, software for computing the topological data analysis method persistence homology, usually supports only one of the two topologies. We answer the question how to compute persistent homology of an image with respect to one of the two topologies using software that is intended for the other topology.

The second research question is concerned with image data as well, and asks how much of the topological information of an image is lost when the resolution is coarsened. As computer tomography scanners are more expensive the higher the resolution, it is an important question in materials science to know which resolution is enough to get satisfying persistent homology. We give theoretical bounds on the information loss based on different geometrical properties of the object to be scanned. In addition, we conduct experiments on sand and stone CT image data.

The third research question is motivated by comparing crystalline materials efficiently. As the atoms within a crystal repeat periodically, crystalline materials are either modeled by unmanageable infinite periodic point sets, or by one of their fundamental domains, which is unstable under perturbation. Therefore a fingerprint of crystalline materials is needed, with appropriate properties such that comparing the crystals can be eased by comparing the fingerprints instead. We define the density fingerprint and prove the necessary properties.

The fourth research question is motivated by studying the hole-structure or connectedness, i.e. persistent homology or merge trees, of crystalline materials. A common way to deal with periodicity is to take a fundamental domain and identify opposite boundaries to form a torus. However, computing persistent homology or merge trees on that torus loses some of the information materials scientists are interested in and is additionally not stable under certain noise. We therefore decorate the merge tree stemming from the torus with additional information describing the density and growth rate of the periodic copies of a component within a growing spherical window. We prove all desired properties, like stability and efficient computability.

Acknowledgements

Throughout my PhD and in general throughout my life I have had the amazing privilege to be surrounded by super supportive people, whom I owe a lot. I don't think I will ever be able to pay back that debt directly to these people, but I hope I will be able to pay it back to society by trying to support future generations of scientists the way I have been supported.

First of all I want to thank my PhD supervisor, Herbert Edelsbrunner, for all the invaluable support he provided me during my PhD. Thank you for believing in me, often even more than I believed in myself. I like how you always communicate with me on eye-level even when I was just a bachelor's student. I am so lucky to have had a PhD supervisor with such an extensive knowledge in the field, such a gift for coming up with the right intuition, and who additionally does not shy away from also working on details, instead of having time for the big picture only (which would be perfectly understandable as well). I learned so much from you. In addition, I am very thankful for the freedom you gave me and for financing my many business trips where I met great scientists all over the world, which is also such a privilege.

I also want to thank the rest of my thesis committee, for accompanying me on this journey and providing valuable advice. In particular, I want to thank Dmitriy Morozov, for all the great scientific input, encouragement, and help. It has always been so much fun to meet you at Dagstuhl, Berkeley, Kyoto, or ISTA and talk to you about research, coming up with so many good ideas together every time. I hope we will continue to have many opportunities like that.

Many thanks to my collaborators of finished and on-going projects! I learned a lot from each and every one of you. And I enjoyed working on the projects together. I also want to warmly thank Lisbeth Fajstrup and Yasu Hiraoka for hosting my research visits in such a welcoming way.

A person who has facilitated my path into academia tremendously is Hubert Wagner. I am so thankful for you for so many things, I probably will not even succeed in listing all of them. Together with Herbert, your enthusiasm for Computational Topology and your fantastic teaching was what got me to love Computational Topology in the first place. During my internship at ISTA, when I had no idea what to choose as a research project, you basically adopted me and introduced me to the exciting topic of streaming algorithms for persistent homology and Euler characteristic curves. You supervised me with such a didactic skill, letting me discover things on my own, but when I was on the wrong tracks, helping me to realize that myself and letting me adjust my course of action myself. I remember the time of that internship was one of the happiest times of my life. You showed me what research is like, and I liked it a lot. But your tremendous help did not end there. I wrote my first research paper with you and you spent a lot of time patiently teaching me how to write research papers. You also helped me with my first review. Throughout the years, even after you left ISTA, I turned to you whenever I needed help, was it making career decisions, or you

listening to my prep talk for an important job application and giving me feedback for hours, thinking together how to change the talk. And apart from all the scientific skills you taught me (Computational Topology, how to do research, some insights on how to write good code, how to write papers, how to write reviews, how to apply for jobs, and so on), you also taught me how to dance Salsa, for which I am very thankful for as well. You are also a great role model for me, for example showing me that science does not need to be the only focus in the life of a scientist. And on top of all of that, you are simply a great friend.

I am also very thankful for the Applied Algebraic Topology Research Network (AATRN), in particular to Henry Adams. AATRN could not wish for a better executive director. You are so kind and caring, for example in making sure that each of us co-directors gets visibility—which has probably helped my career a lot—often doing the most boring tasks yourself. I learned so many things from you and from the other co-directors, Sara, Elchanan, Hanka, and Bastian and from our advisory committee. I am so lucky to be part of this wonderful group. I am also very thankful for all the AATRN members, who contribute with exciting talks, interviews, tutorials, posters, and so on, and who show us that they appreciate our efforts.

A big thank you also goes to every single member of the Edelsbrunner group over the years. You all have been super supportive, and we have had so much fun together. Many of you have been role models for me, and I learned so much from all of you and received so much help from all of you. In particular, lots of love to Elizabeth, who is such an amazing friend and such a stable pillar in my life. Without you my PhD would have been a lot less fun and harder. Also a big thank you to Chiara Martyka, Sophie Rosenmeier, and Dmitrii Vasilev. It has been a pleasure to help supervise your master thesis / internships. Supervising you was probably the most fun part of my whole PhD. I also want to thank my prime cohort at ISTA for all the fun we had together.

Going back in time to before my PhD studies, I received a lot of support already then which lead to my PhD. I am very thankful for my professors at TU Vienna during my bachelor's studies. In particular, I'm thinking of Martin Goldstern, Reinhard Winkler, Harald Woracek, Joachim Schöberl, and Helmut Pottmann.

I am also extremely grateful for the friends I made while studying at TU Vienna: Lisa Weijler, Daniel Ringsmuth, Bernd Schwarzenbacher, Johannes Schürz, Peter Stroppa and later Theresa, Georg Hofstätter and later Sophie, Katharina Riederer, Therese Fehrer and later Odysseas, Daniel Herold and later Sophie, Jyoti Zeilinger, Raffaela Heily and later Daniel, Tobias Danzcul, Nici Kastanek, Philipp Münz, Michael Neunteufel, Daniel Hainschink, and more. I loved how we all helped each other so much during our studies—together we were so strong—not only in the obvious way of explaining bits of mathematics to each other, but in so many other ways: For example each and everyone of you imprinted forever on my personality. I learned so much from you, not only about mathematics, but also about the world, friendships, and who I want to be. And we had so much fun together back then, cooking Kasknöpfle and playing Achtung Kurve and so on, and are still having great fun together these days. Some of you told me that you are thankful to me for explaining you some mathematical concepts, but I think you might be overlooking how much explaining things to you benefited myself. Without you, I probably would not have had the motivation to look at the lecture notes after the lecture and go through the parts that were a little too fast for me. But with you asking about them, I was suddenly super motivated. I hardly ever had to do something on my own, because I always had you all as amazing company. Only during my PhD I realized how much I missed always having someone there to work on the math problem together. Additionally, I improved my explanation skills every single time I

explained something. I also simply liked to see the happy look on your faces, when you understood something you didn't understand before. That's such a joyous feeling. And all of it boosted by confidence, which I'm also very grateful for. I want to thank you all for all of that. And I want to thank Daniel Hainschink, Georg Hofstätter and several others, to be there to make it clear to me early on that there is always someone who is clearly better in mathematics than yourself, and that this is not a problem; instead you can learn a lot from these people.

Going back even further, I am extremely thankful to my family, especially my wonderful brother, Jakob Heiss and my wonderful parents, who showered me with unconditional love, making me the happy and confident person I am. All three of them also showed me from a young age how beautiful and fun mathematics is, sparking that interest, and they tried to keep gender-stereotypes away from me. I am so lucky to have been born into such a loving and supportive family. Additionally, I was very lucky to have academics as parents which made everything easier, and even more so that my brother and I chose the same bachelor's degree, so that I could always ask for his help, when I didn't understand something. Jakob is such an amazingly talented mathematician and I am very proud to be his sister. Additionally, I am very grateful for my loving grand-parents, aunts and uncles, and in particular my cousins, who are particularly dear to me. While my cousin Marie's death was the saddest event of my life, I will always keep happy memories of her.

Most importantly, I want to thank Peter Synak (soon Peter Heiss-Synak), for being the love of my life, for supporting me so greatly throughout my PhD, and for soon moving to Australia with me. You are amazing and I am extremely happy that we are together!

Funding. I was supported by the European Research Council (ERC) Horizon 2020 project "Alpha Shape Theory Extended" No. 788183 and by the Pöttinger Scholarship. In addition, I am very thankful for having been able to attend the second Workshop for Women in Computational Topology in July 2019, funded by the Mathematical Sciences Institute at ANU, the US National Science Foundation through the award CCF-1841455, the Australian Mathematical Sciences Institute and the Association for Women in Mathematics. Two of the projects presented in this thesis started there. One of them reached completion thanks to funding from the MSRI Summer Research in Mathematics program awarded to me and my collaborators in 2020.

About the Author

Teresa Heiss completed her BSc in Technical Mathematics at the Vienna University of Technology (TU Wien) in August 2017, directly before joining ISTA in September 2017. Already during her bachelor's degree, she took a course and an internship on Computational Geometry and Topology at ISTA and wrote her first research paper, which started her passion for the subject, in particular for persistent homology. While she is both interested in theory and applications, she enjoys most to develop new theory motivated by applications that cannot be satisfyingly solved with the existing theory. Searching for such applications, she discovered her interest for materials science. In addition to the four publications described in this thesis, Teresa co-authored three more publications during her PhD. Teresa presented her research at many top international conferences and workshops. Throughout her PhD studies, Teresa has enjoyed being engaged in service activities, for example having been a founding and executive committee member of the Austrian Association of Women in Mathematics (A^2WiM), having been the student representative of the mathematics track at ISTA, and most of all, being a co-director of the Applied Algebraic Topology Research Network (AATRN).

List of Collaborators and Publications

The following publications appear in this thesis:

1. Bea Bleile, Adélie Garin, Teresa Heiss, Kelly Maggs, and Vanessa Robins. The persistent homology of dual digital image constructions. In Ellen Gasparovic, Vanessa Robins, and Katharine Turner, editors, *Research in Computational Topology 2*, pages 1–26. Springer International Publishing, Cham, 2022,
   referred to as [BGH$^+$22] in the rest of the thesis

2. Teresa Heiss, Sarah Tymochko, Brittany Story, Adélie Garin, Hoa Bui, Bea Bleile, and Vanessa Robins. The impact of changes in resolution on the persistent homology of images. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3824–3834, 2021, Copyright © 2021 IEEE,
   referred to as [HTS$^+$21b] in the rest of the thesis,

3. Herbert Edelsbrunner, Teresa Heiss, Vitaliy Kurlin, Philip Smith, and Mathijs Wintraecken. The density fingerprint of a periodic point set. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik,
   referred to as [EHK$^+$21] in the rest of the thesis

4. Herbert Edelsbrunner and Teresa Heiss. Merge trees of periodic filtrations. *arXiv preprint arXiv:2408.16575*, 2024,
   referred to as [EH24] in the rest of the thesis.

Chapters 3, 4, 5, and 6 of this thesis are based on publications 1, 2, 3, and 4 in this order, one chapter per publication. Additionally, Chapter 2 of this thesis combines the background sections of all four.

Contributions. In publication 1 all authors developed the theory together, with Adélie Garin taking a leading role.

In publication 2, I took the lead in developing most of the theoretical bounds (see Section 4.3.1 of this thesis) except the theoretical bound of Section 4.3.2, which was Vanessa Robins' idea. Sarah Tymochko implemented all the software for the experiments (see Section 4.4 of this thesis). Other than that, every author contributed to each part of the project.

In publication 3, the definition of the density fingerprint, along with the idea to compute it via Brillouin zones, stems from Herbert Edelsbrunner, Vitaliy Kurlin, and Philip Smith.

Mathijs Wintraecken and I took a lead in proving the properties (continuity and generic completeness, see Sections 5.2 and 5.3.1). Additionally, I took the lead in proving bounds on how many density functions are needed to distinguish crystals (see Section 5.3.3). Philip Smith contributed greatly, for example, by implementing the software. Vitaliy Kurlin and Philip Smith additionally contributed the application (see Section 5.5). Herbert Edelsbrunner did the majority of the writing. Additionally, we helped each other where necessary.

In publication 4, the definition of the shadow monomial (Definition 6.1.2), which is the key to defining the periodic merge tree, was the idea of Herbert Edelsbrunner. The algorithm was developed by the two of us together and we analysed its running time together (see Section 6.2). I developed the definitions and proofs of Section 6.3 about properties, Section 6.4 about the periodic barcode, and Section 6.5 illustrating the concepts with an example.

Table of Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

This thesis provides several new geometrical and topological methods tailored towards the needs of materials science applications, which will hopefully in the future lead to new insights in materials science. Physical and chemical properties of materials are often determined by the shape of the material. Hence, geometry and topology, the two mathematical fields of shape, are particularly suitable for studying materials. However, some of the existing geometrical and topological methods do not take into account the particularities of materials science applications. Therefore, this thesis provides new methods to overcome these limitations.

Topological Data Analysis (TDA), the study of shape and in particular holes, has proven useful in numerous applications [DSG07, PC14, HNH+16, RNS+17]. A particularly popular TDA method is *persistent homology*, which measures the hole-structure of an object at different length scales. This is achieved by deciding how the object should grow (a so-called *filter function* assigns every part of the growing object the time when it is being added), and then tracking the appearance and disappearance of every hole and every connected component in this sequence of different states in the growth process. The hole-structure of an object can reveal its functional properties. For example, the tunnel-shaped hole of Gramicidin proteins in the membrane of a biological cell are responsible for channeling ions in and out of the cell [EÖ19]. In addition, the hole-structure of an object can be seen as a descriptor carrying information about shape, which can be used to distinguish it from other objects, even in situations where the holes are not the focus of attention. In applications spanning from medical sciences [ARC14, CSMR+23] to cosmology [PEv+16, PAB+19], computing the hole-structure of data and potentially passing the output to a machine learning model has led to relevant insights.

In particular, TDA has established itself as a suitable method to gain insight into materials science [RWS11, HNH+16, LBD+17, KHM20]. This is because physical and chemical properties of materials are often determined by the shape of the material, for example by the size of pores or the width of tunnels through the material. For example, the ability of nanoporous materials to adsorb methane depends on the shape and size of the pores [LBD+17, KHM20]. As materials science is such a promising application area for TDA methods, it is important to tailor TDA methods to the needs of these applications.

However, there has not yet been enough attention on solving the problems encountered when trying to apply TDA methods to materials science data. The disciplines materials

science and TDA are far enough removed so that most TDA experts are not aware of all the problems materials scientists face when trying to use TDA methods on their data. In spite of many successful applications, there are still unsolved problems, some of which are addressed in this thesis. The problems arise from a variety of different materials science applications, from studying buckets of sand on the level of sand grains to studying crystalline materials on the level of atoms. Each of these applications faces its own problems that need to be addressed by providing new TDA theory.

This thesis provides methods to solve four different problems occurring when using TDA methods on materials science data and other data. For each of the four research questions, there is one chapter of the thesis devoted to it (Chapters 3–6). The research is motivated by specific applications in materials science that exhibit these problems. As other disciplines encounter similar problems, the methods described in this thesis can be applied outside materials science as well, for example in medical science and cosmology. The data is usually given either as a digital image or as a point set and both data types are featured in this thesis.

The first two research questions are concerned with image data. The application that motivated this research is studying physical properties of porous materials, like the rate of fluid flow through the material. For this micro-CT (micro Computed Tomography) x-ray scanners are used to produce a three-dimensional high-resolution image of the material. As the material typically consists of two phases, solid grains and between them pores filled by fluid, the image is segmented into a binary image [WS13]. Taking sublevel sets of the (two-sided) Euclidean distance transform of that image corresponds to shrinking and enlarging the pore space [RWS11, DRS14]. Hence, its persistence homology describes size, shape, and connectivity of the pores, which are known to determine physical properties such as permeability and trapping capacity [HHA+13, HRS19]. In addition, the methods presented could be used to analyze other images, for example medical CT images.

In order to topologically study digital images, the first question necessary to answer is how the building blocks of an image—called pixels for 2-dimensional images and voxels in general dimension—are supposed to connect to each other, in other words, which digital topology to choose.

Research Question 1: Which Digital Topology for Image Data?

As there are different choices which digital topology to use for image data, different TDA software developers choose differently. We show how to use software implemented using one topology in applications where another digital topology is needed, with hardly any extra effort. That such a result exists, might not be surprising, but there are some important details needing careful consideration.

For digital images, there is no canonical topology, instead it is necessary to choose one of several digital topologies depending on the application. Two of the most-commonly used digital topologies are called indirect and direct adjacency. For 2-dimensional images, the indirect adjacency is known as 8-connectivity and direct adjacency as 4-connectivity. This is because in indirect adjacency, each voxel of the foreground has $3^d - 1$ neighbors (including diagonal ones), whereas in direct adjacency, it is has only the $2d$ direct neighbors. The difference becomes visible when there is a thin diagonal black line on white background and

a thin diagonal white line on black background, see Figure 1.1. Indirect adjacency considers the thin black line connected, while breaking the thin white line into many disconnected pieces. In other words it favors connectivity of the black foreground. Direct adjacency favors connectivity of the white background: the thin white line stays connected, but the thin black line breaks into disconnected pieces. Hence, in applications where thin dark features on light background are more important, indirect adjacency should be used. Whereas direct adjacency keeps better track of thin light features on dark background, or in other words, thin light cracks between dark parts of the image.

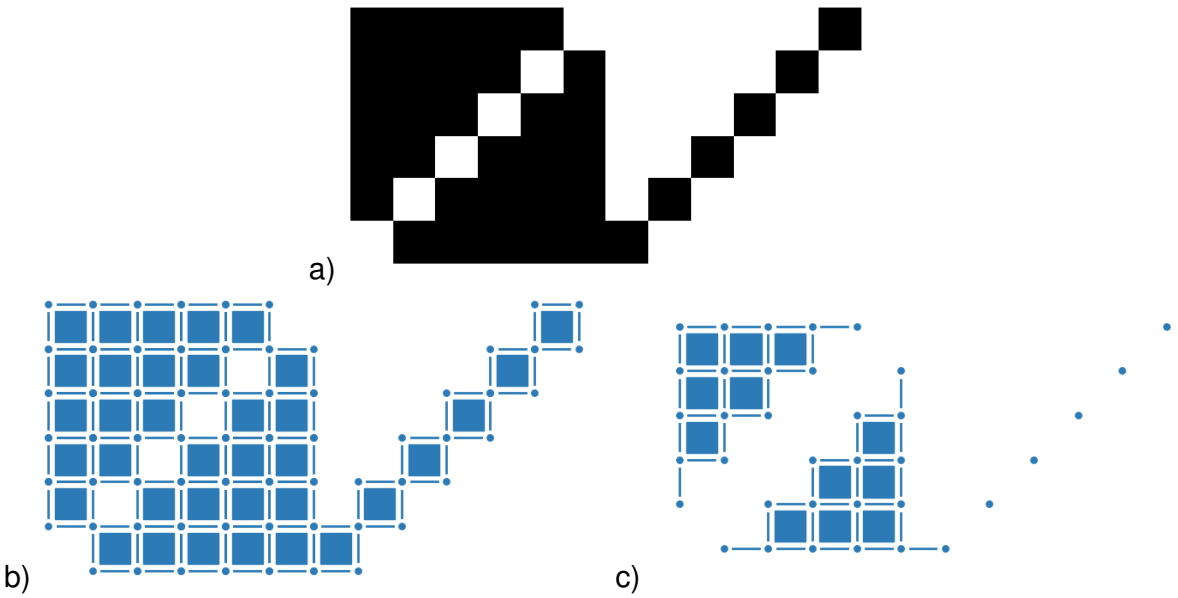

Figure 1.1: a) a digital image with a thin black and a thin white line. b) Indirect adjacency: The cubical complex constructed from the image using the T-construction yields indirect adjacency. The black line stays connected, the white line gets broken apart and thus does not separate the two black triangles. c) Direct adjacency: The cubical complex constructed from the image using the V-construction yields direct adjacency. The black line gets broken apart in six disconnected dots, while the white line separates the two black triangles.

When working with indirect or direct adjacency, note that it is not possible to choose the same topology for the foreground and background. The diagonal connections of the foreground in indirect adjacency automatically block diagonal connections in the complement, and thus cause the background to be connected via direct adjacency. Similarly, when using direct topology for the foreground, the lack of diagonal connections, causes diagonal connections in the complement so that the background is connected via indirect topology.

The image itself, an array of grey values, does not have a topology. Instead, we choose the topology by choosing how to represent the image as a cubical complex. A specific construction that regards voxels as top-dimensional cubical cells—we call it *T-construction*—yields the indirect adjacency topology. Whereas another construction that regards voxels as vertices—we call it *V-construction*—yields the direct adjacency topology.

However, when designing a persistent homology algorithm for image data, the software developer has to decide for a digital topology without knowing for which applications the software will later be used. Once this choice is made, it is not easy to change the implementation to use another digital topology. This is because different digital topologies require

using different cubical complex constructions. For this reason, software with certain advantages is currently only available for one digital topology. For example, while the software cubicle [Wag18, Wag23], which uses indirect topology, can compute persistent homology of particularly large images (e.g., processing several billions of voxels on a regular laptop) using a streaming approach, no such software seems to exist for direct topology. Contrarily, the software diamorse [DFR15, RWS11, DRS15], which has the advantage of saving time by omitting persistence pairs of low persistence, uses direct adjacency only. The question arises: How to use the software of choice with its unique advantages if its digital topology does not fit a given application?

The goal is to describe how software intended for one digital topology can be used to produce results for the opposite one. The gist of the idea here is to flip the grey values of the image, so that the foreground becomes the background and vice versa. Hence, a software that favors connectivity of the background of the flipped image, favors connectivity of the foreground of the original image. It is not surprising that there is some kind of correspondence between the persistent homology of the foreground and the background, because Alexander duality [Ale15, Hat02] provides a correspondence between the homology of a set (in our case the foreground of an image) and the homology of its complement (the background of the image).

Note that there are three relevant details that need to be taken care of. Firstly, when stating the duality between the persistent homology of the original image and the flipped image, it is important that opposite (not same) digital topologies are used for the two images. Secondly, Alexander duality is usually stated between a set and its complement on the sphere, whereas we would like to take the complement within the image domain, a rectangle. Thirdly, there is a dimension shift between finite and infinite lifetime persistence pairs; see Theorem 3.1.4. All three of these details are sometimes overlooked, leading to wrong claims in the literature.

In Chapter 3, we describe in detail how software intended for one digital topology can be used to produce results for the other. As we rigorously prove our claims, it is certain that we took all the necessary details into account. While the result is not particularly surprising, we believe it is important that we provide this resource for other scientists, so that they do not need to reinvent the wheel every time, potentially overlooking the important details.

For proving our theorems, we chose a general approach, first proving the relationship between dual filtrations in general in Section 3.1. This is followed by Sections 3.2, 3.3, and 3.4, where we prove how images can be modified to fit this setting. The general result can be used in different settings as well, for example the filtration of (weighted) Alpha complexes [Ede93] is dual to a filtration of the (weighted) Voronoi tessellation, describing the homology of the complements of shrinking balls.

Research Question 2: Which Resolution for Image Data?

Continuing with the image setting, apart from choosing a digital topology, a resolution needs to be chosen. Micro-CT scanners are more expensive the higher the resolution. Hence, Research Question 2 asks how high the resolution needs to be to approximate topological information satisfyingly.

For this we distinguish between two models. In the first, easier model, we approximate a function $\mathrm{f}\colon \mathbb{R}^d \to \mathbb{R}$ by a grayscale digital image consisting of voxels, which are cubes of

size r^d. The gray value of a voxel is the average of the function f over the cube. We chose indirect adjacency for the image to compute its persistent homology and compare it to the persistent homology of the original function f. For this, we use a standard distance between persistence diagrams, the bottleneck distance. In Corollary 4.2.3, not surprisingly, we prove that when f is L-Lipschitz continuous, the bottleneck distance is bound by $Lr\sqrt{d}$. In other words, the bound is linear in the voxel length r, meaning simply the smaller the voxel length, the better the approximation of the topology.

The more interesting model is the second. We assume there is an object $X \subseteq \mathbb{R}^d$, approximated by a binary image $X(r,t)$. In order to get information about the hole-structure of the object at different length scales, we would like to use persistent homology of the (two-sided) Euclidean distance transform of the object X. However, in applications, X is not available, only its approximation $X(r,t)$. Hence, we compute persistent homology of the discrete Euclidean distance transform of the image $X(r,t)$, and bound its bottleneck distance to the persistent homology of the (continuous) Euclidean distance transform of the set X. Note that this setting is different from the first model: If r is chosen large enough that a small connected component or hole of X is not captured by $X(r,t)$, the distance transforms will be vastly different, leading to bottleneck distances orders of magnitude larger than the voxel length.

For the second model, we give theoretical bounds and experimental results. Our three theoretical bounds are based on three different geometric characteristics of the object X. The bound of Corollary 4.3.9 is based on the reach of the boundary of X, measuring how far the object can be shrunk or enlarged before two of its fronts meet [Blu67]. Theorem 4.3.1 is based on a geometric characteristic, introduced by us: the leash of X with respect to a certain radius—in our setting the diagonal $\sqrt{d}r$ of the pixel. It measures how long the leash of a dog would need to be if attached to a ball of radius $\sqrt{d}r$ so that the dog can visit all parts of the set X while the ball needs to be fully inside X. The geometric characteristic of Theorem 4.3.12 compares how far $X(r,t)$ is from the two most extreme ways of thresholding: $X(r,t)$ consists of the voxels which are covered more than 50 per cent by X, and the two extreme versions would be allowing only those voxels covered 100 per cent by X or all voxels that are covered more than 0 per cent by X.

As this research is motivated by studying buckets of sand or stone, where those three geometric characteristics are often not known, we provide a numerical case study of micro-CT images of such porous materials, segmented into two phases, solid and void. For this, we consider the given high-resolution binary image as ground truth, and track how the bottleneck distance to the ground truth changes when the resolution is being reduced. In addition, we illustrate our analysis methods on a simple synthetic example. Theorem 4.3.1 and the synthetic examples might suggest that as the resolution decreases, components and holes are being lost in $X(r,t)$ at different length-scales of the material, leading to several sudden increases in the leash and thus in the bottleneck distance. This would lead to plateaus between two such increases when plotting the bottleneck distances to the highest resolution image. The lowest resolution within a plateau might be a good compromise between cost and accuracy, because increasing the resolution more only provides a small improvement in accuracy. However, in our analysis of the given micro-CT images these plateaus are not as pronounced and instead seem to vanish in noise.

Research Question 3: Finding a Fingerprint for Periodic Point Data

In Crystal Structure Prediction (CSP), materials scientists try to simulate which possible different crystalline materials a certain molecule can form. Crystalline materials are either periodic or quasi-periodic, but in this thesis, we focus on the more common periodic setting, meaning that the set of atoms in the crystal is translation invariant under a full-dimensional lattice (the integer linear combinations of a basis of $\mathbb{R}^3$). With supercomputers, materials scientists in CSP choose a so-called unit cell (a parallelepiped) with periodic boundary conditions, insert several copies of the molecule, and simulate how the forces on the atoms move the molecules until they converge into a periodic configuration. Repeating this experiment for millions of different unit cells and starting positions of the molecules, they need to find out which few of these configurations have potential to be stable enough in practice to produce this crystal in a laboratory.

Obstacles in this endeavor are the numerous duplicates or near-duplicates in the millions of simulations. For example, if a simulation has created atom positions M inside a certain unit cell, then a different simulation starting with two copies of that unit cell next to each other and twice as many molecules might converge to two copies of M (or a slight perturbation thereof) inside the twice as large unit cell. Note that these simulations represent the same crystal because periodically repeating either of them yields the same periodic point set (or a slight perturbation thereof). Hence, supercomputing time is wasted if both simulations are individually tested for their potential as stable crystals.

Motivated by this application, we introduce the density fingerprint of a periodic point set in $\mathbb{R}^3$ to facilitate fast comparison between crystalline materials by comparing their density fingerprints instead. The density fingerprint of a periodic point set A is the collection of k-th density functions for all non-negative integers k. The k-th density function maps a positive radius r to the proportion of the volume of the regions of $\mathbb{R}^3$ covered by exactly k balls around the points of A. For the comparison between fingerprints to give information about the comparison between crystals, we need certain properties to hold, which we prove:

- Invariance: The idea is that the fingerprint of a crystal should always be the same, no matter how the crystal is rotated or translated. Indeed, the density fingerprint is invariant under all isometries of $\mathbb{R}^3$ which also includes reflections; see Lemma 5.1.2.
- Lipschitz continuity: In order to avoid false negatives, we need to be certain that when the distance between fingerprints is large, the distance between the corresponding crystals is also large. Indeed, the density fingerprint map is Lipschitz continuous. The Lipschitz constant depends on the smallest distance between points and the largest void between points; see Theorem 5.2.2.
- Generic Completeness: We consider false positives as less problematic than false negatives, but nevertheless want to have an informative fingerprint that hardly ever maps two different crystals to the same fingerprint. Indeed, we prove that outside a measure zero subset of the space of periodic point sets, non-isometric periodic point sets never have the same density fingerprint; see Theorem 5.3.1. While this does not prove anything inside the measure zero subset, so far there is no known example of non-isometric crystals with the same density fingerprint, apart from an example in dimension one that we did not manage to extend to higher dimensions and thus it

seems irrelevant given that for applications crystals in three dimensions matter, not one.

- Efficient Computability: While theoretically the conditions above could be strengthened, compromises need to be made to achieve efficient computability. Indeed, the first $k+1$ density functions of the density fingerprint of a periodic point set can be evaluated for a constant number of radii, in time $O(mk^3)$, where m is the number of points per unit cell; see Section 5.4.3. This opens the question how many density functions need to be computed to get enough distinguishing power, which we approach by providing a lower bound in Lemma 5.3.3.

Research Question 4: Extending Persistent Homology to the Periodic Setting

Continuing with the periodic setting, we focus on computing the hole-structure of crystals or other materials that are simulated as periodic. The hole-structure is crucial for estimating properties of the material, like conductivity (which is relevant in battery research [CDS+23]) or the ability to adsorb methane molecules or other green-house gases [LBD+17, KHM20]. The shape and size of tunnels and pores (which are holes of dimension 1 and 2) indicates whether Lithium ions can tunnel through the battery or whether methane or carbon-dioxide molecules can be captured in pores. Note that by regarding the complement of the material, i.e. the space in which the Lithium ions or methane molecules can roam around, both questions can be phrased via connected components (which are holes of dimension 0). Indeed, in battery research we desire connections between the anode and the cathode within the complement of the material, corresponding to paths that a Lithium ion can take. When wanting to trap carbon-dioxide molecules, it seems promising to study connected components of the complement that become connected to the boundary for a certain stage in the growth process.

Ideally, we would want to compute persistent homology of all dimensions of a periodic filter function of the periodic material. However, the infinitely many repetitions in a periodic filter cause the filter to lack a requirement (tameness) for the definition of persistent homology. An alternative is taking the quotient with respect to the periodic repetitions and working on the resulting quotient complex on the torus. However, this is neither stable under certain perturbations nor does it provide the full information; see the Related Work Section in Chapter 6. While it is thus not clear how to satisfyingly define persistent homology for holes of arbitrary dimension, the 0-dimensional case is easier. The *merge tree*, which tracks which components merge with each other at which time in the growth process, is defined also for periodic filters, in spite of not being tame. However, merge trees of periodic filters consists of infinitely many branches, making them impractical.

We introduce the *periodic merge tree* (see Definition 6.1.4), which compresses the infinite merge tree into a finite tree, by removing redundancies, while keeping track of how many branches of the merge tree have been compressed into one. When finitely many branches get compressed into one, we label the compressed branch by that finite number. In the more common infinite case, we label the compressed branch by what we call a *shadow monomial*, giving the density and growth rate of the infinitely many components (corresponding to the branches) seen within a spherical window with growing radius.

While this description uses the infinite merge tree, our algorithm computing the periodic merge tree efficiently performs the computations within only a small finite fundamental domain of the periodic filter. This leads to a running time that is log-linear in the number of edges and vertices per fundamental domain; see Theorem 6.2.4.

We define a pseudo-distance (see Definition 6.3.4) between equivalence classes of periodic merge trees, and show stability with respect to that pseudo-distance (see Theorem 6.3.7). As the pseudo-distance is hard to compute in practice, we introduce a second descriptor derived from the periodic merge tree, the *periodic* 0*-th persistence barcode* (see Section 6.4.1), with a distance that is easier to compute (see Definition 6.4.1). We prove that this descriptor is stable as well (see Theorem 6.4.6) and can easily be computed from the periodic merge tree.

Our methods can be applied to any periodic filter in any dimension. The most common ways to derive a periodic filter from different periodic data types are:

- (Weighted) Alpha filtrations of periodic point sets: Modeling each atom of a periodic material as a point or weighted point with different weights for different atom types yields a (weighted) periodic point set. The (weighted) Delaunay mosaic and filter of that (weighted) periodic point set are periodic as well. The sublevel sets of the (weighted) Delaunay filter are called (weighted) Alpha complexes and have the same persistent homology and merge tree as the union of growing balls around the points.

- Periodic digital images or piecewise constant functions: When simulating periodic functions, like the electron density of a periodic material, it can be useful to discretize space into voxels, and model the function as piecewise constant on the voxels. The result is a periodic grayscale image, which can be the input filter for our methods.

- Piecewise-linear or smooth periodic functions: Instead of modeling a function as piecewise constant, we can model it as piecewise linear or smooth. We can construct a graph with a filter, consisting of the minima and saddle points together with their function values. As they correspond to changes of the connected components of the sublevel sets, the periodic merge tree of this filter tracks the connected components of the sublevel sets of the original function.

Additionally to periodic structures, like crystals, these methods can be applied to structures that are not periodic, but simulated with periodic boundary conditions. Molecular dynamics simulations of aperiodic materials, like glass, often use periodic boundary conditions to avoid boundary effects at the boundary of the simulated cube [CDS$^+$23]. Similarly, cosmological simulations of the universe use periodic boundaries [ABL$^+$13]. The trick here is that an infinite periodic structure does not have any boundary and thus no boundary effects. If the cube is chosen large enough, the simulation is locally realistic in every point, while globally the assumption of periodicity is of course not realistic. Nevertheless, studying global behaviour of these periodic simulations can be meaningful. For example, when the cube size is large enough, the periodic simulation approximates the aperiodic material well when it comes to the percolation threshold (i.e. the largest radius of a ball that can tunnel from one side of the material to the opposite side) [NZ01]. Note that the percolation threshold can be read off the periodic merge tree of the filter that models the complements of the shrinking balls around the atoms, namely as the first so-called catenation event.

1.1 Connections Between Research Questions

On the method side, our solutions to Research Questions 1, 3, and 4 have in common that we remove unwanted boundary effects. In Research Question 1, the two cubical complex constructions corresponding to the two different digital topologies are dual to each other, except at the boundary. We remove this unwanted boundary effect by adding additional cells at the boundaries to make them perfectly dual. Later we need to subtract the effect of these additional cells on the persistent homology. In Research Question 4, the naïve approach of computing persistent homology of a large enough chunk of the periodically repeated data would yield unwanted boundary effects. The method we suggest instead, the periodic merge tree, is carefully constructed to evade boundary effects. Similarly, in Research Question 3, boundary effects would violate the invariance property. The solution here is to use a descriptor that does not have boundary effects, namely the volume of certain intersections of balls, instead of a descriptor like homology that would have boundary effects. This theme of evading boundary effects will therefore be a red thread through the thesis.

On the application side, all four research questions have in common that they arose from situations where existing topological data analysis methods were not enough to tackle the various materials science problems. Different materials science applications look at materials on a different scale. For example, the application motivating Research Questions 1 and 2 studies the hole-structure of micro-CT images of buckets of sand or stone on the length scale of individual sand grains. Whereas in Research Question 3, we study the material on the length scale of individual atoms. The method presented in Research Question 4, can be applied to any length scale, and to both image data (typically appearing for large length scales, like in Research Questions 1 and 2) and point cloud data (for example modeling every atom as a point, like in Research Question 3).

Research Questions 3 and 4 can be combined fruitfully in the following way. In Research Question 3, to avoid boundary effects, we had to use the simple descriptor measuring volumes of intersections of balls, instead of more sophisticated descriptors, like persistent homology. This comes with the disadvantage that the information necessary to reconstruct the point set from its fingerprint is not well accessible inside the fingerprint. We prove in Theorem 5.3.1 that the changes of derivative of the density functions carry the necessary information to reconstruct almost all periodic point sets from their fingerprints. However, since in practice we do not compute a closed form of the function but instead only sample the function at many discrete points, the changes of derivatives are not accessible in this discrete version of the density fingerprint. It would be more straight forward to directly only track the moments where the derivative of the density function changes, instead of tracking all the many discrete moments between changes. Persistent homology of a construction called the order-k Delaunay mosaics of a periodic point set aims at tracking exactly those discrete changes. However, persistent homology contrary to the density fingerprint has boundary effects if applied naïvly. We therefore need to find a version of persistent homology without boundary effects for periodic data. A first step towards this is to solve the case of 0-dimensional persistent homology, i.e. Research Question 4.

A connection between Research Questions 1, 2, and 4 is the usage of image data as an input for persistent homology. The method suggested in Research Question 4 can handle a variety of periodic input data, in particular periodic images. Image data (also called cubical data), does not necessarily need to stem from an image per se, but also occurs in situations where a function is being approximated by a piecewise constant function, using cubes as

pieces. For example, the electron density of a material, which is the probability distribution of the locations of electrons, is used as an informative descriptor of a material for estimating material properties [TAS+23]. To approximate it, values are simulated along a grid, and the resulting array of numbers can be viewed as an image. For a periodic material, the electron density is simulated on the torus, yielding an image whose opposite boundaries should be thought of as adjacent. Such an image can be the input for Research Questions 1, 2, or 4. In the case of such an image on the torus, Research Question 1, simplifies because the two cubical complex constructions are already perfectly dual to each other on the torus, without the need of adding cells at the boundary, because the torus does not have any boundary. However, the results of Research Question 1 do not apply to the infinite image consisting of periodic copies of the original image, i.e., the setting of Research Question 4. This is due to the assumption of finite complexes in the theorem about persistence of dual filtrations (Theorem 3.1.4). Indeed, in the infinite periodic setting in $\mathbb{R}^2$, a death in 0-homology can be dual to a death in 0-homology (see Figure 7.1 in Chapter 7), which is different to Theorem 3.1.4, where deaths can only be dual to births. Research Question 2 stays very similar in the periodic setting and can be combined with Research Question 4: First choosing the resolution using the bounds provided in Chapter 4 and then computing the periodic 0-th persistence barcode from Chapter 6.

CHAPTER 2

Mathematical Background

This chapter combines the background sections of [BGH+22, HTS+21b, EHK+21, EH24].

2.1 Cell Complexes and Filtrations

To allow computers to handle complicated shapes, we break the shape into building blocks. Those building blocks could be simplices, cubes, or more generally, cells. We give an unnecessarily general definition, given that we will restrict ourselves in Chapters 3 and 4 to regular CW-complexes or even cubical complexes and in Chapter 6 to graphs (with possible self-loops and possibly multiple edges between the same pair of vertices). The following definition takes Hatcher's definition of a Δ-complex [Hat02, p.103] (which is more general than a simplicial complex, for example by allowing the two endpoints of an edge to coincide), but replaces simplices by convex polytopes (to also include regular CW-complexes, in particular cubical complexes).

A *cell complex*, K, is a topological space $|K|$ (for example a subset of $\mathbb{R}^d$) equipped with continuous maps $\tau_i : P_i \to |K|$, $i \in I$, where P_i is a closed convex p_i-dimensional polytope, and I is an index set. We require the following.

- We call the image of a map $\tau_i : P_i \to |K|$, a *(closed) p_i-dimensional cell* and the image of its restriction to the interior of the polytope, an *open p_i-dimensional cell*.
- The restriction of each map τ_i to the interior of the polytope is a homeomorphism onto its image (the open p_i-dimensional cell).
- Every point of $|K|$ is in exactly one open cell.
- For each $i \in I$, every facet of P_i can be identified with a P_j with $j \in I$ such that $p_j = p_i - 1$ and τ_i restricted to that facet is τ_j. If τ_j appears in this context an odd number of times for τ_i, we call the cell corresponding to τ_j a *face* of the cell corresponding to τ_i.
- A set in $A \subseteq |K|$ is open if and only if the preimage $\tau_i^{-1}(A)$ is open in P_i for every $i \in I$.

Depending on the context, we treat K as a combinatorial object (a collection of cells with face relations), or as a geometric object (the set $|K|$ of points in the union of these cells,

dropping the vertical lines for convenience). The *dimension* $\dim(K)$ of the cell complex K is the maximum dimension of cells in K. The (closed or open) 0-dimensional cells, which are points, are referred to as the *vertices*, and the open 1-dimensional cells as the *edges* of K.

Note that this definition allows for infinitely many cells but in this thesis, K will always be either finite (Chapters 3 and 4) or locally finite (Chapter 6). In addition, this definition allows, for example, the two endpoints of an edge to coincide. When we want to avoid such situations, we restrict ourselves to regular cell complexes. A cell complex is *regular* if the closure of each k-cell is homeomorphic to the closed k-dimensional ball D^k. In Chapters 3 and 4, all cell complex will be assumed to be regular in addition to finite.

Let K be a cell complex with cells τ and σ. If τ is a face of σ, then σ is a *coface* of τ, written as $\tau \preceq \sigma$. The *codimension* of a pair of cells $\tau \preceq \sigma$ is the difference in dimension, $\dim(\sigma) - \dim(\tau)$. If τ is a face of σ of codimension 1, we call it a *facet* of σ, and write $\tau \lhd \sigma$. A function $f : K \to \mathbb{R}$ on the cells of K is *monotonic* if $f(\sigma) \leq f(\tau)$ whenever $\sigma \preceq \tau$.

Definition 2.1.1. A *filtered (cell) complex* (K, f) is a cell complex K together with a monotonic function $f : K \to \mathbb{R}$ on the cells. We call f a *filter* on K.

For finite cell complexes K, a linear ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$ of the cells in K, such that $\sigma_i \preceq \sigma_j$ implies $i \leq j$, is *compatible* with the filter f when

$$f(\sigma_0) \leq f(\sigma_1) \leq \ldots \leq f(\sigma_n).$$

Note that the monotonicity condition implies that, for $r \in \mathbb{R}$, the *sublevel set*

$$K_r := f^{-1}(-\infty, r]$$

is a subcomplex of K. The nested sequence of sublevel sets is called a *filtration*. The value $f(\sigma)$ determines when a cell σ enters the filtration. The definition of a compatible ordering also implies that each step in the sequence

$$\emptyset \subset \{\, \sigma_0 \,\} \subset \{\, \sigma_0, \sigma_1 \,\} \subset \cdots \subset \{\, \sigma_0, \sigma_1, \ldots, \sigma_n \,\} = K$$

is a subcomplex, and every sublevel set $K_r = f^{-1}(-\infty, r]$ appears somewhere in this sequence: $f^{-1}(-\infty, r] = f^{-1}(-\infty, f(\sigma_i)] = \{\, \sigma_0, \sigma_1, \ldots, \sigma_i \,\}$ for $i = \max\{\, i = 0, \ldots, n \mid f(\sigma_i) \leq r \,\}$.

2.1.1 Dual Cell Complexes and Filtrations

Definition 2.1.2. The d-dimensional cell complexes K and K^* are *combinatorially dual* if there is a bijection $K \to K^*, \sigma \mapsto \sigma^*$ between the sets of cells such that

1. (Dimension Reversal) $\dim(\sigma^*) = d - \dim \sigma$ for all $\sigma \in K$.
2. (Face Reversal) $\sigma \preceq \tau \iff \tau^* \preceq \sigma^*$ for all $\sigma, \tau \in K$.

Definition 2.1.3. Two filtered complexes (K, f) and (K^*, g) are *dual filtered complexes* if K and K^* are combinatorially dual to one another and if there exists a linear ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$ of the cells in K that is compatible with f and its dual ordering $\sigma_n^*, \sigma_{n-1}^*, \ldots, \sigma_0^*$ is compatible with g.

Proposition 2.1.4. *Let K and K^* be combinatorially dual. Suppose two filters $f : K \to \mathbb{R}$ and $f^* : K^* \to \mathbb{R}$ satisfy $f^*(\sigma^*) = -f(\sigma)$. Then (K, f) and (K^*, f^*) are dual filtered complexes.*

2.2 Periodic Setting

Before introducing periodic point sets, we introduce two more general concepts. We recall that $A \subseteq \mathbb{R}^d$ is *locally finite* if any compact subset of $\mathbb{R}^d$ contains only finitely many points of A. It is a *Delone set* [DLS98] if there exist $r, R > 0$ such that every open ball of radius r contains at most one point of A and every closed ball of radius R contains at least one point of A. In other words, no two points of A can be closer than $2r$ and no point of $\mathbb{R}^d$ can be further from A than R. We refer to the largest such r as the *packing radius* and the smallest such R as the *covering radius* of A. A Delone set is necessarily infinitely large and its points are in a sense evenly spread out over the entire Euclidean space.

In Chapters 5 and 6, we are specifically interested in point sets, complexes and filters that are periodically repeated copies of a finite point set, complex or filter. The natural language for such a setting is that of lattices in Euclidean space, discussed e.g. in [Cas97, Zhi15]. Let $u_1, u_2, \ldots, u_p$ be linearly independent vectors in $\mathbb{R}^d$. The set of all integer combinations is the *lattice* spanned by these vectors:

$$\Lambda = \Lambda(u_1, u_2, \ldots, u_p) = \left\{\lambda_1 u_1 + \lambda_2 u_2 + \ldots + \lambda_p u_p \mid \lambda_i \in \mathbb{Z} \text{ for } 1 \leq i \leq p\right\}, \tag{2.1}$$

we call $u_1, u_2, \ldots, u_p$ a *basis* of Λ and $p = \dim \Lambda$ the *dimension* of the lattice. If $p = d$, we call the lattice *full* and write U for the $d \times d$ matrix whose columns are the basis vectors. The *span* of the vectors, denoted $\operatorname{span}(\Lambda)$, is the set of all real combinations of the vectors, which is a p-dimensional linear subspace of $\mathbb{R}^d$. Similarly, the *unit cell*, denoted $\operatorname{Unit} U$, of the vectors is the set of real combinations with coefficients in $[0, 1)$. The same lattice can be spanned by different bases, but the dimension, the span, and the (p-dimensional) volume of the unit cell are always the same. We write $\operatorname{vol}_p(\Lambda)$ for the latter and note that for $p = d$, it is the determinant of U. Assuming the lattice is full, recall that the columns of U are the basis vectors. Since $U^{-1}U$ is the identity matrix, U^{-1} maps Λ to the standard integer lattice, $\mathbb{Z}^d$, and U^{-1} applied to a lattice vector gives the integer coordinates of that vector with respect to the basis given by U.

Given a lattice Λ, a set A is Λ-*periodic* if it is translation invariant under translations of Λ, i.e. for every $u \in \Lambda$, $A + u = A$. If Λ is full and A is additionally locally finite, then A is a *periodic point set*. As any unit cell of Λ is compact, there are only finitely many points of Λ inside, and we call that finite point set, a *motif* M. The periodic point set A can then be written as the Minkowski sum $A = M + \Lambda$. If Λ is the largest lattice with the property that the periodic point set A is Λ-periodic, then we call any unit cell of Λ *primitive*. Note that periodic point sets are Delone sets.

It is important to keep in mind that the basis of a lattice and therefore the primitive unit cell are not unique. This is illustrated in Figure 2.1, which shows three of the infinitely many bases of the hexagonal lattice: a together with $b - a$, b, or $b + a$. Applying Niggli's algorithm for the *Niggli reduced cell* [Nig28] to this particular lattice, there is an ambiguity between the bases $\{a, b\}$ and $\{a, b - a\}$, because the projections of b and $b - a$ onto the line of a both have length $\frac{1}{2}\|a\|$. The tie can be broken by preferring b, but this causes a discontinuity in the construction of the reduced unit cell.

Similarly, for a d-dimensional lattice Λ, a locally finite complex, $K \subseteq \mathbb{R}^d$, is Λ-*periodic* if $\sigma \in K$ and $u \in \Lambda$ implies $\sigma + u \in K$. And a filter on a Λ-periodic complex K is a Λ-*periodic* filter, $F\colon K \to \mathbb{R}$, if it satisfies $F(\sigma) = F(\sigma + u)$ for every $\sigma \in K$ and $u \in \Lambda$.

Periodicity means that information is repeated infinitely often. To avoid infinitely large descriptors, it can therefore be useful to reduce redundancy by building quotients. The

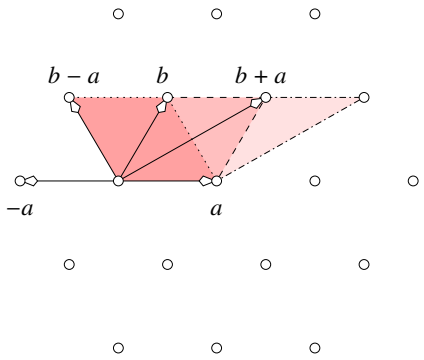

Figure 2.1: The bases $\{a, b - a\}$, $\{a, b\}$, $\{a, b + a\}$ of the hexagonal lattice, and the corresponding unit cells.

quotient, $\mathbb{R}^d/\Lambda$, is obtained by identifying points $x, y \in \mathbb{R}^d$ whenever $y - x \in \Lambda$. This quotient with topology inherited from the d-dimensional Euclidean space is usually referred to as the *d-dimensional torus*. Similarly, we can identify cells of K if they are translates of each other by a vector of Λ. The resulting *quotient complex*, K/Λ, is a complex on the d-torus consisting of finitely many cells because K is locally finite and the d-torus is compact.[1] We write $F/\Lambda\colon K/\Lambda \to \mathbb{R}$ for the *quotient filter*, which is a convenient representation of $F\colon K \to \mathbb{R}$. Unfortunately, this representation is not unique since, for example, K and F are also periodic with respect to 2Λ, or really any sublattice of Λ.

2.2.1 Rigid Motions and Isometries

Important classes of transformations between periodic point sets in $\mathbb{R}^3$ are rigid motions and isometries. Rather than a fixed set in $\mathbb{R}^3$, we often consider the class of sets that are equivalent under a particular type of transformation. For example, a *rigid motion* is a map $\mathbb{R}^3 \to \mathbb{R}^3$ that is composed of a rotation and a translation. It preserves distances between pairs of points as well as orientations of ordered triplets of points. An *isometry* is a rigid motion possibly composed with a reflection, and so preserves distances but not necessarily orientations. This is a relevant group of transformations to this thesis as we model crystals by isometry classes of periodic point sets in Chapter 5.

2.3 Persistent Homology

In this section, we provide a brief outline of the definitions and results of persistent homology. The reader who is not familiar with the theory of persistent homology should refer to [EH08, ZC05] for a more comprehensive introduction.

When working with data, standard topological quantities can be highly sensitive to noise and small geometric fluctuations. Persistent homology addresses this problem by examining a collection of spaces, indexed by a real variable often representing an increasing length scale. These spaces are modelled by a cell complex K with a filter function $f : K \to \mathbb{R}$ assigning to each cell the scale at which this cell appears.

[1] Note, however, that even a periodic complex that is simplicial can have a non-simplicial quotient, which includes the possibilities of more than two edges connecting the same two vertices and an edge connecting a single vertex back to itself.

2.3.1 Definition

Given a filtered complex (K, f), we obtain inclusions $f^{-1}(-\infty, r] \to f^{-1}(-\infty, s]$ of sublevel sets for $r \leq s$. Assume f is *tame*, i.e. all sublevel sets have finitely generated homology groups and there are only finitely many thresholds r, where the homology groups change. Applying degree-k homology with coefficients in $\mathbb{Z}/2\mathbb{Z}$ to these inclusions yields linear maps between vector spaces

$$H_k(f^{-1}(-\infty, r]) \to H_k(f^{-1}(-\infty, s]).$$

The resulting functor $H_k(f) : (\mathbb{R}, \leq) \to \mathrm{Vec}_{\mathbb{Z}/2\mathbb{Z}}$ from the poset category $(\mathbb{R}, \leq)$ to the category of vector spaces over the field $\mathbb{Z}/2\mathbb{Z}$ is called a *persistence module*, for details see [CDSGO16].

As discussed in [CDSGO16], Gabriel's Theorem from representation theory implies that the persistence module $H_k(f)$ decomposes into a sum of persistence modules consisting of $\mathbb{Z}/2\mathbb{Z}$ for $r \in [b, d)$ connected by identity maps, and 0 elsewhere, called *interval modules* $\mathbb{I}_{[b,d)}$:

$$H_k(f) \cong \bigoplus_{l \in L} \mathbb{I}_{[b_l, d_l)}.$$

Each interval summand $\mathbb{I}_{[b_l,d_l)}$ represents a degree-k homological feature that is *born* at $r = b_l$ and *dies* at $r = d_l$. Intervals with $d_l = \infty$ are called *essential* and correspond to features that never die.

The *degree-k persistence diagram* of f is the multiset

$$\mathrm{Dgm}^k(f) = \{\, [b_l, d_l) \mid l \in L \,\}$$

and we call the pair (b_l, d_l) a *(persistence) point* in the diagram $\mathrm{Dgm}^k(f)$. We write $[b_l, d_l)_k \in \mathrm{Dgm}^k(f)$ to denote the homological degree of an interval and define the *persistence diagram* of f as the disjoint union over all degrees:

$$\mathrm{Dgm}(f) = \bigsqcup_{k=0}^{\dim(K)} \mathrm{Dgm}^k(f).$$

Writing $\mathrm{Dgm}_{\mathbf{F}}(f)$ for the multiset of finite intervals with $d_l < \infty$, and $\mathrm{Dgm}_\infty(f)$ for the remaining essential ones, we obtain $\mathrm{Dgm}(f) = \mathrm{Dgm}_{\mathbf{F}}(f) \sqcup \mathrm{Dgm}_\infty(f)$.

2.3.2 Computation

To compute the persistence diagram $\mathrm{Dgm}(f)$ of a finite filtered complex (K, f), we choose an ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$ of the cells in K that is compatible with f. Choosing a compatible ordering can be viewed as breaking ties between cells σ_i and σ_j with $f(\sigma_i) = f(\sigma_j)$. Without breaking the tie, they would appear at the same step in the nested sequence of sublevel sets $\left(f^{-1}(\infty, r]\right)_{r \in \mathbb{R}}$. For the following computations however, we must add exactly one cell at every step:

$$\emptyset \subset \{\, \sigma_0 \,\} \subset \{\, \sigma_0, \sigma_1 \,\} \subset \cdots \subset \{\, \sigma_0, \sigma_1, \ldots, \sigma_{n-1} \,\} \subset \{\, \sigma_0, \sigma_1, \ldots, \sigma_n \,\} = K.$$

When adding the cells one step at a time, a cell of dimension k causes either the birth of a k-dimensional feature or the death of a $(k-1)$-homology class [DE95], that is, each

cell is either a *birth* or a *death cell*. A pair (σ_i, σ_j) of cells where σ_j kills the homological feature created by σ_i is called a *persistence pair*. A persistence pair (σ_i, σ_j) corresponds to the interval $[f(\sigma_i), f(\sigma_j)) \in \mathrm{Dgm}_{\mathbf{F}}(f)$. Note that this interval can be empty, namely if $f(\sigma_i) = f(\sigma_j)$. Empty intervals are usually neglected in the persistence diagram. A birth cell σ_i with no corresponding death cell is called *essential*, and corresponds to the essential interval $[f(\sigma_i), \infty) \in \mathrm{Dgm}_{\infty}(f)$.

Recall that presentations for the standard homology groups are found by studying the image and kernel of integer-entry matrices that represent the boundary maps taking oriented chains of dimension k to those of dimension $(k-1)$ [Mun84]. In persistent homology, we work with the $\mathbb{Z}/2\mathbb{Z}$ *total boundary matrix* D, which is defined by $D_{i,j} = 1$ if $\sigma_i \lhd \sigma_j$ and 0 otherwise. Define

$$r_D(i,j) = \operatorname{rank} D_i^j - \operatorname{rank} D_i^{j-1} - \operatorname{rank} D_{i+1}^j + \operatorname{rank} D_{i+1}^{j-1}$$

where $D_i^j = D[i:n, 0:j]$ is the lower-left sub-matrix of D attained by deleting the first rows up to $i-1$ and the last columns starting from $j+1$. With this we get the persistence pairs from the following theorem.

Theorem 2.3.1 (Pairing Uniqueness Lemma [CSEM06])**.** *Given a linear ordering of the cells in a filtered cell complex (K, f), (σ_i, σ_j) is a persistence pair if and only if $r_D(i,j) = 1$.*

The ranks are usually computed by applying the column reduction algorithm [EH08] to obtain the reduced matrix R and using the property that $\operatorname{rank} D_i^j = \operatorname{rank} R_i^j$ under the operations of the algorithm. The persistence pairs can then be read off easily since $r_R(i,j) = 1$ if and only if the ith entry of the jth column of the reduced matrix is the lowest 1 of this column. However, in this thesis, we can work directly with r_D.

Corollary 2.3.2. *If $r_D(i,j) \neq 1$ and $r_D(j,i) \neq 1$ for all j then the cell σ_i is essential.*

Proof. The fact that every cell is either a birth or a death cell implies that σ_i must be an unpaired birth or death cell. However, as every filtration begins as the empty set, there are no unpaired death cells. □

2.3.3 Bottleneck Distance

Let f, f' be two functions with persistence diagrams $\mathrm{Dgm}^k(f)$ and $\mathrm{Dgm}^k(f')$. The *bottleneck distance* between $\mathrm{Dgm}^k(f)$ and $\mathrm{Dgm}^k(f')$ is defined using matchings γ on persistence points in the two diagrams,

$$\mathrm{d_B}(\mathrm{Dgm}^k(f), \mathrm{Dgm}^k(f')) = \min_{\gamma} \max_{x} \|x - \gamma(x)\|_{\infty}.$$

Note that a matching, γ, is permitted to pair any persistence point (b,d) in either diagram with any point (c,c) on the diagonal of the other diagram, i.e. imagine both diagrams to have infinitely many points on the diagonal additionally to the persistence points. The stability theorem of persistent homology [CSEH07] now tells us that if f and f' are tame functions that are point-wise close, then their persistence diagrams are also close:

$$\mathrm{d_B}(\mathrm{Dgm}^k(f), \mathrm{Dgm}^k(f')) \leq \|f - f'\|_{\infty}.$$

CHAPTER 3

The Persistent Homology of Dual Digital Image Constructions

This chapter is based on [BGH+22]. Therefore, the CC-BY license of this thesis does not apply to this chapter, instead the license of the original Springer book chapter applies. Additionally, an extended abstract of it was published in the Young Researchers Forum of the Symposium on Computational Geometry [GHM+20].

To compute the persistent homology of a grayscale digital image one needs to build a simplicial or cubical complex from it. For cubical complexes, the two commonly used constructions (corresponding to direct and indirect digital adjacencies) can give different results for the same image. The two constructions are almost dual to each other, and we use this relationship to extend and modify the cubical complexes to become dual filtered cell complexes. We derive a general relationship between the persistent homology of two dual filtered cell complexes, and also establish how various modifications to a filtered complex change the persistence diagram. Applying these results to images, we derive a method to transform the persistence diagram computed using one type of cubical complex into a persistence diagram for the other construction. This means software for computing persistent homology from images can now be easily adapted to produce results for either of the two cubical complex constructions without additional low-level code implementation.

Prior work. At its most basic level, the algebraic relationship between the persistent homology of two dual filtered cell complexes is similar to that between persistent homology and persistent relative cohomology. The latter corresponds to taking the anti-transpose of the boundary matrix [DSMVJ11] or equivalently, leaving the boundary matrix as is and applying the row reduction algorithm instead of the column-reduction algorithm [DSMVJ11, EÖ20]. Lemma 3.1.2 shows that the same relationship applies to the boundary matrices of dual filtered cell complexes. Therefore, Theorem 3.1.4 can be viewed as a translation of the known bijection between the persistence pairs of persistent homology and relative persistent cohomology into the setting of dual filtrations. This theorem is the first step towards establishing the mapping between persistence diagrams of the T- and the V-construction of images in Section 3.4. Furthermore, it applies more generally to the persistent homology of dual filtered cell complexes without using the connection to persistent relative cohomology.

The symmetry theorem of extended persistence diagrams [CSEH09] is also closely related to our Theorem 3.1.4 and Corollary 3.1.5. Given X a manifold without boundary and a

function $f : X \to \mathbb{R}$, the extended persistent homology sequence starts with a filtration by sublevel sets $f^{-1}(-\infty, s]$ and continues with the relative homology of the pair $(X, f^{-1}[r, \infty))$, where s is an increasing threshold and r is a decreasing one. In [CSEH09] the symmetry theorem follows from a duality result for the extended persistence diagrams of a simplicial complex K whose underlying space is X with f defined on the vertices of K. As observed in [ES12], the cubical complex constructions used in digital image analysis do not satisfy the duality theorem of extended persistence because the Partition Lemma of [CSEH09] fails for cubical complexes. The authors of [ES12] overcome this by constructing a simplicial complex from the digital image that consistently reflects the connectivity of both sub- and super-level sets, and use this to obtain the expected duality in the extended persistence diagram. In contrast, we work with the existing widely-implemented cubical complex constructions of digital images and establish results relating homology sequences of dual filtered complexes rather than the homology and relative homology sequences of sub- and super-level set filtrations of a single complex. This permits a simple high-level algorithm to transform between two regular (not extended) persistence diagrams computed from a digital image.

Outline. Our results are aimed at both pure and applied mathematicians who want to understand and use the relationship between the persistent homology of dual filtered cell complexes and particularly the two standard constructions of cubical complexes from digital images.

Section 3.1 establishes the relationship between persistence diagrams of two dual filtered cell complexes. In Section 3.2, we describe and formalise the two standard cubical complexes used in topological computations on digital images. We explain how these two complexes (the T- and V-constructions described in Chapter 1) must be extended and modified to form dual filtered cell complexes with underlying space homeomorphic to the d-sphere. The effects these modifications have on persistence diagrams are derived in Section 3.3. For the investigation of one of these effects we use the long exact sequence of a filtered pair of cell complexes arising from the category theoretic view of persistence modules.

The last Section 3.4 states the results for persistence diagrams of digital images and explains how to compute the persistence diagram of the T-construction by simple manipulation of a persistence diagram computed using the V-construction, and vice versa. This gives a practical method for adapting the output from existing software packages that use one or the other construction to obtain the persistence diagram for the dual construction.

3.1 The Persistent Homology of Dual Filtered Complexes

In this entire chapter we assume all cell complexes to be finite and regular.

Recall again that in standard homology and cohomology the coboundary map is the adjoint of the boundary map. Hence, given a consistent choice of bases for the chain and cochain groups, their matrix representations are related simply by taking the transpose. In [DSMVJ11], another algebraic relationship is established between persistent homology and persistent relative cohomology, based on the observation that the filtration for relative cohomology reverses the ordering of cells in the total (co)boundary matrix. The same reversal of ordering holds for the dual filtered cell complexes defined here, so we obtain a similar relationship between the persistence diagrams. Our proof of the correspondence between persistence pairs in dual filtrations uses the matrix rank function and pairing

uniqueness lemma in a similar way to the combinatorial Helmholtz-Hodge decomposition of [EÖ20]. Nonetheless, Theorem 3.1.4 interprets the underlying linear algebra in the setting of dual filtered complexes and only uses the concept of persistent homology without using the connection to persistent relative cohomology, which makes it more accessible.

For this section suppose (K, f) and (K^*, g) are dual filtered cell complexes with $n+1$ cells. Suppose that a linear ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$ of the cells in K is compatible with the filtration (K, f), and that $\sigma_n^*, \sigma_{n-1}^*, \ldots, \sigma_0^*$ is the dual linear ordering compatible with g. Let D be the total boundary matrix of K and D^* be the total boundary matrix of K^* with their respective orderings.

Remark 3.1.1. *A useful indexing observation is that σ_i^* is the $(n-i)$-th cell of the dual filtration.*

We denote by $D^\perp$ the anti-transpose of the matrix D, that is the reflection across the minor diagonal: $D^\perp_{i,j} = D_{n-j,n-i}$. Anti-transposition is also the composition of standard matrix transposition with a reversal of the order of the columns and of the rows.

Lemma 3.1.2. *The matrix D^* is the anti-transpose $D^\perp$ of D, that is,*

$$D^*_{i,j} = D_{n-j,n-i} = D^\perp_{i,j}.$$

Proof. The equivalences below follow from the definition of D, of dual cell complexes, and the above remark.

$$D_{n-j,n-i} = 1 \Leftrightarrow \sigma_{n-j} \lhd \sigma_{n-i} \Leftrightarrow \sigma^*_{n-i} \lhd \sigma^*_{n-j} \Leftrightarrow D^*_{i,j} = 1.$$

□

Lemma 3.1.3. *The sub-matrices defined in Section 2.3.2 satisfy*

$$(D_i^j)^\perp = (D^\perp)^{n-i}_{n-j}$$

and thus

$$\text{rank } D_i^j = \text{rank } (D^\perp)^{n-i}_{n-j}$$

and

$$r_D(i,j) = r_{D^\perp}(n-j, n-i).$$

Proof. The first statement follows from

$$(D_i^j)^\perp = (D[i:n, 0:j])^\perp = D^\perp[(n-j):n, 0:(n-i)] = (D^\perp)^{n-i}_{n-j}.$$

The second statement follows because anti-transposition is attained by composing the rank preserving operations of transposition and row and column permutations. The third statement follows from the second through:

$$\begin{aligned} r_D(i,j) &= \text{rank } D_i^j - \text{rank } D_i^{j-1} - \text{rank } D_{i+1}^j + \text{rank } D_{i+1}^{j-1} \\ &= \text{rank}(D^\perp)^{n-i}_{n-j} - \text{rank}(D^\perp)^{n-i}_{n-j+1} - \text{rank}(D^\perp)^{n-i-1}_{n-j} + \text{rank}(D^\perp)^{n-i-1}_{n-j+1} \\ &= r_{D^\perp}(n-j, n-i). \end{aligned}$$

□

Theorem 3.1.4 (Persistence of Dual Filtrations)**.** *Let (K, f) and (K^*, g) be dual filtered complexes with compatible ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$. Then*

1. *(σ_i, σ_j) is a persistence pair in the filtered complex (K, f) if and only if (σ_j^*, σ_i^*) is a persistence pair in (K^*, g).*
2. *σ_i is essential in (K, f) if and only if σ_i^* is essential in (K^*, g).*

Proof. Lemma 3.1.3 implies that $r_D(i, j) = r_{D^*}(n - j, n - i)$. Therefore,

$$r_D(i, j) = 1 \Leftrightarrow r_{D^*}(n - j, n - i) = 1.$$

By the Pairing Uniqueness Lemma 2.3.1, the above implies that (σ_i, σ_j) is a persistence pair whenever the $(n - j)$-th cell of the dual filtration (K^*, g) is paired with the $(n - i)$-th, thus proving Part (1). For Part (2), Lemma 3.1.3 also tells us that the following two statements are equivalent:

- Both $r_D(i, j) \neq 1$ and $r_D(j, i) \neq 1$ for all j.
- Both $r_{D^*}(n - j, n - i) \neq 1$ and $r_{D^*}(n - i, n - j) \neq 1$ for all $n - j$.

By Corollary 2.3.2, this means that σ_i is an essential cell in (K, f) if and only if the $(n - i)$-th cell σ_i^* is essential in the dual filtration (K^*, g). □

Corollary 3.1.5. *Let (K, f) and (K^*, g) be dual filtered complexes. Then*

1. $$[f(\sigma_i), f(\sigma_j)) \in \mathrm{Dgm}_{\mathbf{F}}^k(f) \Leftrightarrow [g(\sigma_j^*), g(\sigma_i^*)) \in \mathrm{Dgm}_{\mathbf{F}}^{d-k-1}(g).$$
2. $$[f(\sigma_i), \infty) \in \mathrm{Dgm}_{\infty}^k(f) \Leftrightarrow [g(\sigma_i^*), \infty) \in \mathrm{Dgm}_{\infty}^{d-k}(g).$$

Proof. Note that for a persistence pair (σ_i, σ_j), found for an ordering compatible with the function f, the birth value is $f(\sigma_i)$ and the death value is $f(\sigma_j)$. The result then follows directly from Theorem 3.1.4. □

Remark 3.1.6. *It is worth noting that there is a dimension shift between essential and non-essential pairs coming from the fact that the birth cell defines the dimension of a homological feature. For finite persistence pairs, the birth cell changes from σ_i (of dimension k) to σ_j^* (of dimension $d - (k + 1)$) in the dual, while for an essential cycle, the birth cell in the dual is σ_i^*. This dimension shift also appears in our results on images later on.*

3.2 Filtered Cell Complexes from Digital Images

As described in the introduction, the motivating application for the duality results of this chapter is grayscale digital image analysis. This section begins with the definition of grayscale digital images and describes the two standard ways to model such images by cubical complexes as well as the modifications required to make these dual filtered complexes.

Definition 3.2.1. A *d-dimensional grayscale digital image* of size $(n_1, n_2, \ldots, n_d)$ is an $\mathbb{R}$-valued array $\mathcal{I} \in M_{n_1 \times n_2 \times \ldots \times n_d}(\mathbb{R})$. Equivalently, it is a real-valued function on a d-dimensional rectangular grid

$$\mathcal{I} : I = [\![1, n_1]\!] \times [\![1, n_2]\!] \times \ldots \times [\![1, n_d]\!] \to \mathbb{R}$$

where $[\![1, n_i]\!]$ is the set $\{ k \in \mathbb{N} \mid 1 \leq k \leq n_i \}$. The index set, I, of $\mathcal{I}$ is also called the *image domain*.

Recall that elements $p \in I$ are called *pixels* when d=2, *voxels* if $d \geq 3$, and the value $\mathcal{I}(p) \in \mathbb{R}$ is the *grayscale value* of p.

One would like to use persistent homology to analyse such images via their sublevel sets. However, the canonical topology on $I \subseteq \mathbb{Z}^d \subset \mathbb{R}^d$ makes it a totally disconnected discrete space. To induce a meaningful topology on the image that better represents the perceived connectivity of the voxels, grayscale digital images are modelled by regular cubical complexes [Kov89].

Definition 3.2.2. An *elementary k-cube* $\sigma \subset \mathbb{R}^d$ is the product of d elementary intervals,

$$\sigma = e_1 \times e_2 \times \ldots \times e_d$$

such that k of the intervals have the form $e_i = [l_i, l_i + 1]$ and $d - k$ are degenerate, $e_i = [l_i, l_i]$.

A *cubical complex* $K \subset \mathbb{R}^d$ is a cell complex consisting of a set of elementary k-cubes, such that all faces of $\sigma \in K$ are also in K, and such that all vertices of K are related by integer offsets.

3.2.1 Top-cell and Vertex Constructions

There are two common ways to build a filtered cubical complex from an image $\mathcal{I} : I \longrightarrow \mathbb{R}$. One method is to represent the voxels as vertices of the cubical complex as in [RWS11]. We call this cubical complex the vertex construction, or V-construction for short. The second method takes voxels as top-dimensional cells: we call it the top-cell construction, or T-construction. It is shown in [Kov89], that the vertex construction corresponds to the graph-theoretical direct adjacency used in traditional digital image processing and the top-cell construction to the indirect adjacency model. These adjacency models are also respectively referred to as the open and closed digital topologies.

An example of how each construction is built from an image is given in Figure 3.1. The explicit definitions of such constructions are given below.

Definition 3.2.3. Given a d-dimensional grayscale digital image $\mathcal{I} : I \to \mathbb{R}$, of size $(n_1, n_2, \ldots, n_d)$, the *V-construction* is a filtered cell complex $(V(I), V(\mathcal{I}))$ defined as follows.

1. $V(I)$ is a cubical complex built from an array of $(n_1 - 1) \times \ldots \times (n_d - 1)$ elementary d-cubes and all their faces.
2. The vertices $v^{(0)} \in V(I)$ are indexed exactly by the elements $p \in I$, and we define the function $V(\mathcal{I})$ firstly on these vertices as,

$$V(\mathcal{I})(v^{(0)}) = \mathcal{I}(p).$$

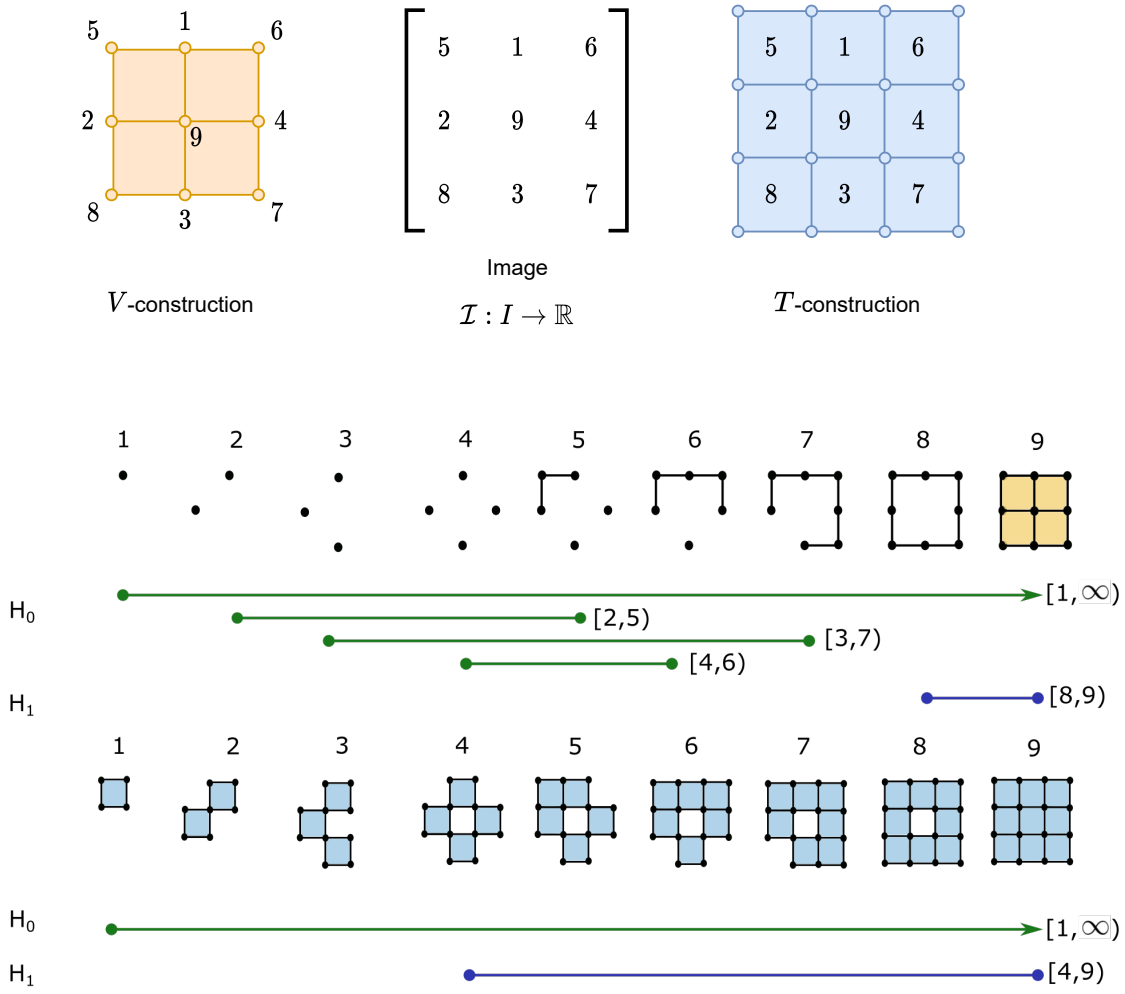

Figure 3.1: Top: The V- and T-constructions generated by an image $\mathcal{I} : I \to \mathbb{R}$ with the values of $\mathcal{I}$ indicated on the vertices and the top-dimensional cells, respectively.
Middle: the filtration $V(\mathcal{I}) : V(I) \to \mathbb{R}$ and the corresponding persistence pairs.
Bottom: the filtration $T(\mathcal{I}) : T(I) \to \mathbb{R}$ and the corresponding persistence pairs.

Then for an elementary k-cube σ, the function takes the maximal value of its vertices

$$V(\mathcal{I})(\sigma) = \max_{v^{(0)} \preceq \sigma} V(\mathcal{I})(v^{(0)}).$$

This ensures that $V(\mathcal{I})$ is monotonic with respect to the face relation on $V(I)$.

Definition 3.2.4. Given a d-dimensional grayscale digital image $\mathcal{I} : I \to \mathbb{R}$, of size $(n_1, n_2, \ldots, n_d)$, the *T-construction* is a filtered cell complex $(T(I), T(\mathcal{I}))$ defined as follows.

1. $T(I)$ is a cubical complex built from the array of $n_1 \times \ldots \times n_d$ elementary d-cubes and all their faces.

2. The d-cells $\tau^{(d)} \in T(I)$ are indexed exactly by the elements $p \in I$, and we define the function $T(\mathcal{I})$ firstly on these top-dimensional cells as,

$$T(\mathcal{I})(\tau^{(d)}) = \mathcal{I}(p).$$

Then for an elementary k-cube σ, the function takes the smallest value of any adjacent d-cubes,

$$T(\mathcal{I})(\sigma) = \min_{\sigma \preceq \tau^{(d)}} T(\mathcal{I})(\tau^{(d)}).$$

This ensures that $T(\mathcal{I})$ is monotonic with respect to the face relation on $T(I)$.

The next section describes how to modify the original image and take quotients to obtain dual complexes and filtrations.

3.2.2 Modifications for Duality

The cubical complexes defined using the top-cell and vertex constructions are not strictly dual to each other in the standard context of a rectangular digital image domain due to the presence of a boundary. If the rectangular image domain happens to be the unit cell for a periodic structure, then we can identify opposite faces and form the d-torus. On this d-torus, $V(I)$ and $T(I)$ are dual cubical complexes with vertices and d-cubes indexed by I. Taking $V(\mathcal{I})$ as the function on $V(I)$ and $T(-\mathcal{I})$ as the function on $T(I)$, we obtain dual filtered complexes and can immediately apply Corollary 3.1.5 to deduce the persistence pairs of one filtration from the other.

Otherwise, the more commonly encountered situation is that the image is a simple convex domain in $\mathbb{R}^d$. A standard approach to handling the boundary is to form a quotient identifying the boundary to a point so that the convex domain becomes a subset of the d-sphere. To obtain dual cell complexes on the d-sphere, we increase the size of the image domain before taking the quotient modulo the boundary. The image function is assigned a large arbitrary value on these extra voxels and dual filtered complexes are obtained by considering $\mathcal{I}$ in one construction and $-\mathcal{I}$ in the other as detailed in the following definitions and results.

Let $\mathcal{I} : I \to \mathbb{R}$ be a grayscale digital image with index set $I = [\![1, n_1]\!] \times \ldots \times [\![1, n_d]\!]$, and set

$$N > \max_{p \in I} \mathcal{I}(p).$$

Definition 3.2.5. The *padded image* $\mathcal{I}^{\mathbf{P}} : I^{\mathbf{P}} \to \mathbb{R}$ has image domain $I^{\mathbf{P}} = [\![0, n_1 + 1]\!] \times \ldots \times [\![0, n_d + 1]\!]$ and image function

$$\mathcal{I}^{\mathbf{P}}(p) = \begin{cases} \mathcal{I}(p), & \text{for } p \in I \\ N, & \text{for } p \in I^{\mathbf{P}} \setminus I. \end{cases}$$

As shown in Figure 3.2, the padded image is simply obtained by adding a shell of N-valued voxels to $\mathcal{I}$. We apply the V- and T-constructions to the padded image to obtain the functions $V(\mathcal{I}^{\mathbf{P}}) : V(I^{\mathbf{P}}) \to \mathbb{R}$ and $T(\mathcal{I}^{\mathbf{P}}) : T(I^{\mathbf{P}}) \to \mathbb{R}$, respectively.

Let $T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$ denote the cell complex obtained from $T(I)$ by attaching a d-cell $\kappa^{(d)}$ along the boundary $\partial T(I)$ (see [Hat02, p.5] for details). Furthermore, let $V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$ be the quotient cell complex obtained by identifying all points of the boundary (see [Hat02, p.8] for details). Note that both these modifications create cells that are not elementary cubes.

Lemma 3.2.6. *Given a rectangular digital image domain, I, the quotient, $V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$, of the padded V-construction modulo its boundary is the combinatorial dual of $T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$.*

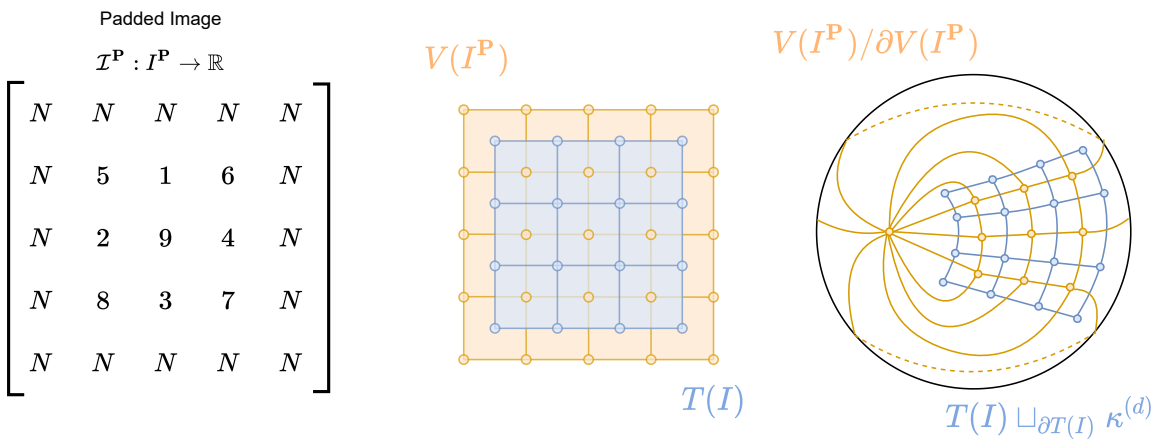

Figure 3.2: The transformation of the V- and T-construction into dual cell complexes (right) using the padded image (left) and a mapping from $V(I^{\mathbf{P}})$ to $T(I)$ (center).

Proof. Each elementary k-cube, $\sigma \in V(I^{\mathbf{P}})$ takes the form

$$\sigma = e_1 \times \ldots \times e_d, \quad e_i = [l_i, l_i + 1] \text{ or } e_i = [p_i, p_i]$$

where k of the elementary intervals are non-degenerate with $l_i \in \{0, \ldots, n_i\}$ and $(d-k)$ are degenerate with $p_i \in \{0, \ldots, n_i + 1\}$. Note that $\sigma \in \partial V(I^{\mathbf{P}})$ if at least one degenerate interval has $p_i = 0$ or $(n_i + 1)$. Now consider the following cell constructed from σ:

$$\sigma^* = e_1^* \times \ldots \times e_d^*, \quad e_i^* = [l_i + \tfrac{1}{2}, l_i + \tfrac{1}{2}] \text{ or } [p_i - \tfrac{1}{2}, p_i + \tfrac{1}{2}]$$

with l_i and p_i as defined above. This cell has k degenerate intervals and $(d-k)$ non-degenerate ones so σ^* is an elementary $(d-k)$-cube. If we insist that $\sigma \notin \partial V(I^{\mathbf{P}})$, then we see that $p_i \in \{1, \ldots, n_i\}$, and the degenerate coordinate values $(l_i + \frac{1}{2}) \in \{\frac{1}{2}, \frac{3}{2} \ldots, (n_i + \frac{1}{2})\}$. Thus we obtain a bijection between k-cells in $V(I^{\mathbf{P}}) \setminus \partial V(I^{\mathbf{P}})$ and $(d-k)$-cells in $T(I)$. Mapping the 0-cell $[\partial V(I^{\mathbf{P}})] \in V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$ to the d-cell attached to $\partial T(I)$ yields a dimension reversing bijection between all cells of $V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$ and those of $T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$.

The next step is to confirm that the face relations between cells in $V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$ are mapped to coface relations in $T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$. By the construction above, all interior face relations for $V(I^{\mathbf{P}})$ map to coface relations for $T(I)$. Given that only cells in the boundary belong to $[\partial V(I^{\mathbf{P}})]$, this correspondence is inherited by the quotient.

Hence, the last detail we need to check is that the vertex $[\partial V(I^{\mathbf{P}})]$ in $V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$ has dual face relations to the d-cell $\kappa^{(d)}$ attached to the boundary of $T(I)$ in $T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$. This is equivalent to the statement that

$$[\partial V(I^{\mathbf{P}})] \preceq \sigma \text{ in } V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}}) \Leftrightarrow \sigma^* \preceq \kappa^{(d)} \text{ in } T \sqcup_{\partial T} \kappa^{(d)}.$$

Now, σ^* is a face of $\kappa^{(d)}$ if and only if $\sigma^* \in \partial T(I)$, which means at least one of the degenerate elementary intervals of σ^* has $l_i + \frac{1}{2} = \frac{1}{2}$ or $(n_i + \frac{1}{2})$. This makes $l_i = 0$ or n_i, so the corresponding elementary interval in the dual cell σ is $e_i = [l_i, l_i + 1] = [0, 1]$ or $[n_i, n_i + 1]$. This forces $\sigma \cap \partial V(I^{\mathbf{P}}) \neq \emptyset$, so that $[\partial V(I^{\mathbf{P}})] \preceq \sigma$. The converse implication follows in the same manner, and we are done. □

Lemma 3.2.7. *Given a rectangular digital image domain, I, the quotient, $T(I^{\mathbf{P}})/\partial T(I^{\mathbf{P}})$, of the padded T-construction modulo its boundary is the combinatorial dual of $V(I^{\mathbf{P}}) \sqcup_{\partial V(I^{\mathbf{P}})} \kappa^{(d)}$.*

Proof. This follows from the same arguments as the previous lemma with the roles of T and V reversed. Note that we pad the V-construction before attaching the cell $\kappa^{(d)}$ to account for the fact that $T(I)$ naturally has more cells than $V(I)$. □

We have described how the two cubical complex models can be augmented to form dual cell complexes of the d-sphere. We now show how to obtain dual filtered cell complexes by comparing the image function on one construction with its negative on the other. The details are made precise in the lemmata below.

First note that the function $V(-\mathcal{I}^{\mathbf{P}})$ is constant on $\partial V(I^{\mathbf{P}})$ so it induces a function on the quotient space, $\widetilde{V}(-\mathcal{I}^{\mathbf{P}}) : V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}}) \to \mathbb{R}$ with $\widetilde{V}(-\mathcal{I}^{\mathbf{P}})([\partial V(I^{\mathbf{P}})]) = -N$ and agreeing with $V(-\mathcal{I}^{\mathbf{P}})$ on all other cells. Similarly, the function $T(\mathcal{I})$ extends to a function $\widehat{T}(\mathcal{I})$ on $T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$ with $\widehat{T}(\mathcal{I})(\kappa^{(d)}) = N$.

Lemma 3.2.8. *For each $\sigma \in T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$ and dual cell $\sigma^* \in V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$ we have*

$$-\widehat{T}(\mathcal{I})(\sigma) = \widetilde{V}(-\mathcal{I}^{\mathbf{P}})(\sigma^*).$$

Proof. Firstly, suppose $\dim \sigma = d$ and $\sigma \neq \kappa^{(d)}$, the d-cell attached to the boundary. Suppose $p \in I$ is the corresponding element of the image domain, so that $\mathcal{I}(p) = \widehat{T}(\mathcal{I})(\sigma)$. The dual cell $\sigma^* \in V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}})$ corresponds to the same voxel but is given the negative value

$$\widetilde{V}(-\mathcal{I}^{\mathbf{P}})(\sigma^*) = -\mathcal{I}(p) = -\widehat{T}(\mathcal{I})(\sigma).$$

For the remaining d-cell $\kappa^{(d)}$, with dual $[\partial V(I^{\mathbf{P}})]^*$, the function values satisfy

$$-\widehat{T}(\mathcal{I})(\kappa^{(d)}) = -N = \widetilde{V}(-\mathcal{I}^{\mathbf{P}})([\partial V(I^{\mathbf{P}})]).$$

Lastly, suppose $\sigma \in T(I) \sqcup_{\partial T(I)} \kappa^{(d)}$ and $\dim \sigma < d$. By construction, it follows that

$$\begin{aligned} -\widehat{T}(\mathcal{I})(\sigma) &= -\min_{\tau^{(d)} \succeq \sigma} \widehat{T}(\mathcal{I})(\tau^{(d)}) = \max_{\tau^{(d)} \succeq \sigma} -\widehat{T}(\mathcal{I})(\tau^{(d)}) = \max_{\upsilon^{(0)} \preceq \sigma^*} \widetilde{V}(-\mathcal{I}^{\mathbf{P}})(\upsilon^{(0)}) \\ &= \widetilde{V}(-\mathcal{I}^{\mathbf{P}})(\sigma^*) \end{aligned}$$

as required. □

Now define the functions $\widetilde{T}(-\mathcal{I}^{\mathbf{P}})$ on $T(I^{\mathbf{P}})/\partial T(I^{\mathbf{P}})$ and $\widehat{V}(\mathcal{I}^{\mathbf{P}})$ on $V(I^{\mathbf{P}}) \sqcup_{\partial V(I^{\mathbf{P}})} \kappa^{(d)}$ similarly to those above.

Lemma 3.2.9. *For each $\sigma \in V(I^{\mathbf{P}}) \sqcup_{\partial V(I^{\mathbf{P}})} \kappa^{(d)}$ and dual cell $\sigma^* \in T(I^{\mathbf{P}})/\partial T(I^{\mathbf{P}})$ we have*

$$-\widehat{V}(\mathcal{I}^{\mathbf{P}})(\sigma) = \widetilde{T}(-\mathcal{I}^{\mathbf{P}})(\sigma^*).$$

Proof. Similar to Lemma 3.2.8 with the roles of V and T interchanged. □

Corollary 3.2.10. *For a grayscale digital image $\mathcal{I} : I \to \mathbb{R}$*

1. *The filtered complexes $(T(I) \sqcup_{\partial T(I)} \kappa^{(d)}, \widehat{T}(\mathcal{I}))$ and $(V(I^{\mathbf{P}})/\partial V(I^{\mathbf{P}}), \widetilde{V}(-\mathcal{I}^{\mathbf{P}}))$ are dual.*
2. *The filtered complexes $(V(I^{\mathbf{P}}) \sqcup_{\partial V(I^{\mathbf{P}})} \kappa^{(d)}, \widehat{V}(\mathcal{I}^{\mathbf{P}}))$ and $(T(I^{\mathbf{P}})/\partial T(I^{\mathbf{P}}), \widetilde{T}(-\mathcal{I}^{\mathbf{P}}))$ are dual.*

Proof. This follows directly from applying Lemma 3.2.8 for part (1) and Lemma 3.2.9 for part (2), then Proposition 2.1.4. □

3.3 Persistence Diagrams of the Modified Filtrations

In the previous section, we showed that the T- and V-constructions built from an image can be modified via padding, cell attachment, and taking quotients to become dual cell complexes of the d-sphere. Our next step is to examine the effect of such operations on the persistence module and diagram.

As in Section 3.2, suppose $\mathcal{I} : I \to \mathbb{R}$ is a grayscale digital image and $N > \max \mathcal{I}$. The specific operations we study are

1. Padding an image $\mathcal{I} : I \to \mathbb{R}$ with a outer shell of N-valued pixels, then forming the V- and T-constructions.
2. Attaching a d-cell to the boundary of $V(I^{\mathbf{P}})$ or to $T(I)$ with value N.
3. Taking the quotient modulo the boundary in the negative padded filtration, *i.e.* changing from $V(-\mathcal{I}^{\mathbf{P}})$ to $\widetilde{V}(-\mathcal{I}^{\mathbf{P}})$ and from $T(-\mathcal{I}^{\mathbf{P}})$ to $\widetilde{T}(-\mathcal{I}^{\mathbf{P}})$.

Of these, the first two have relatively transparent effects on the persistent homology of the filtered spaces. Padding the image as in (1) does not change the persistence diagrams; attaching a d-cell as in (2) simply creates an essential d-cycle with birth at N. We summarise these formally as follows.

Proposition 3.3.1. *For a grayscale digital image $\mathcal{I} : I \to \mathbb{R}$*

1. $\mathrm{Dgm}(V(\mathcal{I}^{\mathbf{P}})) = \mathrm{Dgm}(V(\mathcal{I}))$ *and* $\mathrm{Dgm}(T(\mathcal{I}^{\mathbf{P}})) = \mathrm{Dgm}(T(\mathcal{I}))$

2.

$$\mathrm{Dgm}(\widehat{V}(\mathcal{I})) = \mathrm{Dgm}(V(\mathcal{I})) \cup \{\, [N, \infty)_d \,\}$$

and

$$\mathrm{Dgm}(\widehat{T}(\mathcal{I})) = \mathrm{Dgm}(T(\mathcal{I})) \cup \{\, [N, \infty)_d \,\}.$$

The remaining operation to investigate is the third, namely the effect of taking the quotient modulo the boundary. For this we need some machinery we will now introduce.

3.3.1 Long Exact Sequence of a Filtered Pair

To examine the effect of taking quotients on persistence diagrams, we use the description of persistence modules as functors together with a long exact sequence (LES) for these. Given a pair of cell complexes (K, A) with $A \subseteq K$, we obtain a short exact sequence (SES) of cellular chain complexes inducing the LES

$$\ldots \to H_k(A) \xrightarrow{i} H_k(K) \xrightarrow{p} H_k(K, A) \xrightarrow{\delta} H_{k-1}(A) \to \ldots$$

which is a standard tool for analysing the homology of the pair, where $H_k(K, A)$ denotes the relative homology of the pair.

Similarly, suppose we have a filtered cell complex (K, f) and a sub-complex $A \subseteq K$. Then the restriction $f|_A : A \to \mathbb{R}$ induces a filtered sub-complex $(A, f|_A)$ and, at each index

$r \in \mathbb{R}$, we obtain a pair (K_r, A_r), where $K_r = f^{-1}(-\infty, r]$ and $A_r = f|_A^{-1}(-\infty, r]$. Theorem 4.5 in [VYP18] states that the $\mathbb{R}$-indexed collection of SESs of cellular chain complexes

$$0 \to C_*(A_r) \to C_*(K_r) \to C_*(K_r, A_r) \to 0$$

yields a LES of persistence modules:

Theorem 3.3.2 (Long Exact Sequence for Relative Persistence Modules)**.** *Given a monotonic function $f : K \to \mathbb{R}$ and a sub-complex $A \subseteq K$, there is a long exact sequence of persistence modules*

$$\ldots \to H_k(f|_A) \to H_k(f) \to H_k(f, f|_A) \to H_{k-1}(f|_A) \to H_{k-1}(f) \to \ldots$$

where $H_k(f, f|_A)$ denotes the persistence module given by $r \mapsto H_k(K_r, A_r)$.

Here the persistence modules are functors from the poset category $(\mathbb{R}, \leq)$ to $\mathrm{Vec}_{\mathbb{Z}/2\mathbb{Z}}$ and the morphisms are natural transformations obtained from the standard connecting homomorphisms and the linear transformations induced by inclusions and projections at each filtration index. The kernels and cokernels of the morphisms are themselves persistence modules defined by taking the kernel or cokernel at each filtration index, with maps between corresponding vector spaces at different filtration indices induced by inclusions. This result is implicit in the recent work in [BM21, Mil20], where it follows as corollary of the fact that persistence modules form an abelian category whereby the snake lemma holds.

3.3.2 Persistence of the Image with Boundary Identified

The remaining operation to investigate is the effect of taking the quotient of a padded image modulo the boundary filtered with the negative of the image function. We state the result in terms of a space K homeomorphic to the d-dimensional closed disc D^d, filtered by a function $f : K \to \mathbb{R}$ taking a constant minimal value, $\min f = -N$, on the boundary of K so that Lemma 3.3.3 applies to both T- and V-constructions. Using the long exact sequence of a pair, we show that the $(d-1)$-cycle with interval $[-N, \max f)_{d-1}$ representing the boundary is removed while a d-cycle with interval $[\max f, \infty)_d$ is added.

Lemma 3.3.3. *Take a monotonic function $f : K \cong D^d \to \mathbb{R}$ with*

$$\sigma \in \partial K \Rightarrow f(\sigma) = -N = \min f$$

and induced quotient map $\widetilde{f} : K/\partial K \to \mathbb{R}$. Then

$$\mathrm{Dgm}(\widetilde{f}) = \big(\mathrm{Dgm}(f) \setminus \{\, [-N, \max f)_{d-1} \,\}\big) \cup \{\, [\max f, \infty)_d \,\}.$$

Proof. For pairs of cell complexes the relative homology groups are naturally isomorphic to the reduced homology groups $\tilde{H}_k(\tilde{f})$ of the quotient [Hat02, p.124]. Naturality implies that the result extends to persistence modules and that the reduced persistence modules differ only by the essential interval $\mathbb{I}_{[-N,\infty)}$ in degree 0. To compute the reduced persistence modules $\tilde{H}_k(\tilde{f})$ of the quotient, we therefore consider the LES of the filtered pair $(f, f|_{\partial K})$

$$\ldots \to H_k(f|_{\partial K}) \xrightarrow{\alpha_k} H_k(f) \to H_k(f, f|_{\partial K}) \to H_{k-1}(f|_{\partial K}) \xrightarrow{\alpha_{k-1}} H_{k-1}(f) \to \ldots$$

where α_k is the map induced by the inclusion $\partial K \subseteq K$. Taking the cokernel of α_k and the kernel of α_{k-1} the LES yields the SES

$$0 \to \mathsf{Coker}(\alpha_k) \to H_k(f, f|_{\partial K}) \to \mathsf{Ker}(\alpha_{k-1}) \to 0.$$

First assume $d > 1$ and note that, in this case,

$$H_k(f|_{\partial K}) \cong \begin{cases} \mathbb{I}_{[-N,\infty)} & \text{for } k = d-1, 0 \\ 0 & \text{otherwise.} \end{cases}$$

Thus $\mathsf{Im}(\alpha_k) \cong \mathsf{Ker}(\alpha_k) = 0$ for $k \neq d-1, 0$. For α_{d-1}, the image of the essential $(d-1)$-cycle of the boundary dies once all cells in (K, f) have been filtered at function value $\max f$. Hence

$$\mathsf{Im}(\alpha_{d-1}) \cong \mathbb{I}_{[-N,\max f)} \quad \text{and} \quad \mathsf{Ker}(\alpha_{d-1}) \cong \mathbb{I}_{[\max f,\infty)}.$$

As $-N = \min f$ we conclude $\alpha_0(\mathbb{I}_{[-N,\infty)}) = \mathbb{I}_{[-N,\infty)}$, so that

$$\mathsf{Im}(\alpha_0) \cong \mathbb{I}_{[-N,\infty)} \quad \text{and} \quad \mathsf{Ker}(\alpha_0) = 0.$$

Since K is homeomorphic to a d-dimensional disc, $H_d(f) = 0$. Hence $\mathsf{Coker}(\alpha_d) = 0$ and, for $k = d$, the SES implies

$$\tilde{H}_d(\tilde{f}) \cong H_d(f, f|_{\partial K}) \cong \mathsf{Ker}(\alpha_{d-1}) \cong \mathbb{I}_{[\max f,\infty)}.$$

For $0 \leq k < d$ the persistence module on the right of the SES is trivial. Thus

$$\tilde{H}_k(\tilde{f}) \cong H_k(f, f|_{\partial K}) \cong \mathsf{Coker}(\alpha_k) \cong \begin{cases} H_{d-1}(f)/\mathbb{I}_{[-N,\max f)} & \text{for } k = d-1 \\ H_k(f) & \text{for } 0 < k < d-1 \\ H_0(f)/\mathbb{I}_{[-N,\infty)} & \text{for } k = 0 \end{cases}$$

and the result follows for $d > 1$.

For $d = 1$ the only non-trivial persistence module of the boundary is $H_0(f|_{\partial K}) \cong \mathbb{I}_{[-N,\infty)} \oplus \mathbb{I}_{[-N,\infty)}$ and we obtain

$$\mathsf{Im}(\alpha_0) \cong \mathbb{I}_{[-N,\infty)} \oplus \mathbb{I}_{[-N,\max f)} \quad \text{and} \quad \mathsf{Ker}(\alpha_0) \cong \mathbb{I}_{[\max f,\infty)}.$$

For $k = 1$ we proceed as above and, for $k = 0$, the SES yields

$$\tilde{H}_0(\tilde{f}) \cong H_0(f, f|_{\partial K}) \cong \mathsf{Coker}(\alpha_0) \cong H_0(f)/\big(\mathbb{I}_{[-N,\infty)} \oplus \mathbb{I}_{[-N,\max f)}\big).$$

As above we conclude $H_0(\tilde{f}) \cong H_0(f)/\mathbb{I}_{[-N,\max f)}$. □

Corollary 3.3.4. *For a d-dimensional image $\mathcal{I} : I \to \mathbb{R}$*

$$\mathsf{Dgm}(\widetilde{V}(-\mathcal{I}^{\mathbf{P}})) = \mathsf{Dgm}(V(-\mathcal{I}^{\mathbf{P}})) \setminus \{\, [-N, -\min \mathcal{I})_{d-1} \,\} \cup \{\, [-\min \mathcal{I}, \infty)_d \,\}$$

and

$$\mathsf{Dgm}(\widetilde{T}(-\mathcal{I}^{\mathbf{P}})) = \mathsf{Dgm}(T(-\mathcal{I}^{\mathbf{P}})) \setminus \{\, [-N, -\min \mathcal{I})_{d-1} \,\} \cup \{\, [-\min \mathcal{I}, \infty)_d \,\}.$$

Proof. This follows from Lemma 3.3.3 applied to $f = V(-\mathcal{I}^{\mathbf{P}})$ and $f = T(-\mathcal{I}^{\mathbf{P}})$ respectively, using $\max V(-\mathcal{I}^{\mathbf{P}}) = -\min \mathcal{I}$ and $\max T(-\mathcal{I}^{\mathbf{P}}) = -\min \mathcal{I}$. □

3.4 Duality Results for Images

In this section, we explicitly describe the relationship between the diagrams of both the T- and V-constructions. Software to compute persistent homology of an image $\mathcal{I} : I \to \mathbb{R}$ typically builds one of the two constructions implicitly so the results in this section provide a solution to the problem of how to use software based on the V-construction to compute a persistence diagram with respect to the T-construction, and vice versa.

For the algorithms that we define in this section we assume the following sub-routines given a grayscale digital image $\mathcal{I}$.

1. PAD$(\mathcal{I}, N)$: returns the image padded with an outer shell of N-valued voxels.
2. NEG$(\mathcal{I})$: multiplies each voxel value by -1.
3. $\max(\mathcal{I})$, $\min(\mathcal{I})$: returns the maximum and minimum voxel values of $\mathcal{I}$ respectively.
4. VCON$(\mathcal{I})$, TCON$(\mathcal{I})$: returns the persistence diagrams $\mathsf{Dgm}(V(\mathcal{I}))$ and $\mathsf{Dgm}(T(\mathcal{I}))$ of the V- and T-construction of the image respectively.

3.4.1 From the V-construction to the T-construction

Suppose we have software that computes the persistent homology of a d-dimensional grayscale digital image $\mathcal{I}$ using the V-construction. The following theorem states that the persistence diagram of the T-construction for $\mathcal{I}$ can be calculated directly from the pairs in that of the V-construction of the negative padded image.

Theorem 3.4.1 (T from V)**.** *For a grayscale digital image $\mathcal{I} : I \to \mathbb{R}$ the diagrams of the V- and T-constructions satisfy*

$$\mathsf{Dgm}_{\mathbf{F}}(T(\mathcal{I})) = \{\, [-q,-p)_{d-k-1} \mid [p,q)_k \in \mathsf{Dgm}_{\mathbf{F}}(V(-\mathcal{I}^{\mathbf{P}})) \,\} \setminus \{\, [\min \mathcal{I}, N)_0 \,\}$$

and

$$\mathsf{Dgm}_{\infty}(T(\mathcal{I})) = \{\, [\min \mathcal{I}, \infty)_0 \,\}.$$

Proof. That $\mathsf{Dgm}_{\infty}(T(\mathcal{I})) = \{\, [\min \mathcal{I}, \infty)_0 \,\}$ follows from the fact that $T(I) \cong D^{(d)}$ and the first cell in the filtration occurs at time $\min T(\mathcal{I}) = \min \mathcal{I}$. For the finite case:

$$\begin{aligned}
\mathsf{Dgm}_{\mathbf{F}}(T(\mathcal{I})) &= \mathsf{Dgm}_{\mathbf{F}}(\widehat{T}(\mathcal{I})) \\
&= \{\, [-q,-p)_{d-k-1} \mid [p,q)_k \in \mathsf{Dgm}_{\mathbf{F}}(\widetilde{V}(-\mathcal{I}^{\mathbf{P}})) \,\} \\
&= \{\, [-q,-p)_{d-k-1} \mid [p,q)_k \in \mathsf{Dgm}_{\mathbf{F}}(V(-\mathcal{I}^{\mathbf{P}})) \setminus \{\, [-N, -\min \mathcal{I})_{d-1} \,\} \,\} \\
&= \{\, [-q,-p)_{d-k-1} \mid [p,q)_k \in \mathsf{Dgm}_{\mathbf{F}}(V(-\mathcal{I}^{\mathbf{P}})) \,\} \setminus \{\, [\min \mathcal{I}, N)_0 \,\}
\end{aligned}$$

where the equalities follow from Proposition 3.3.1, Corollary 3.1.5 and Corollary 3.3.4 respectively. □

The structure of the algorithm follows immediately from the theorem and is summarised below.

Algorithm 1 Computing the T-construction persistence diagram with V-construction software.

Require: An image $\mathcal{I}$ and the sub-routine VCON.

1: $\mathrm{Dgm}(T(\mathcal{I})) \leftarrow \{\, [\min(\mathcal{I}), \infty)_0 \,\}$
2: $N \leftarrow \max(\mathcal{I}) + C$ ▷ choose C to ensure $N \gg \max(\mathcal{I})$
3: $-\mathcal{I}^{\mathbf{P}} \leftarrow \mathrm{NEG}(\mathrm{PAD}(\mathcal{I}, N))$
4: $\mathrm{Dgm}(V(-\mathcal{I}^{\mathrm{P}})) \leftarrow \mathrm{VCON}(-\mathcal{I}^{\mathbf{P}})$ ▷ Apply V-construction software.
5: **for** $[p, q)_k$ in $\mathrm{Dgm}(V(-\mathcal{I}^{\mathbf{P}}))$ with $p \neq -N$ **do**
6: $\quad \mathrm{Dgm}(T(\mathcal{I})) \leftarrow \mathrm{Dgm}(T(\mathcal{I})) \cup \{\, [-q, -p)_{d-k-1} \,\}$
7: **return** $\mathrm{Dgm}(T(\mathcal{I}))$ ▷ Output T-construction persistence diagram.

3.4.2 From the T-construction to the V-construction

In the other direction, suppose we have software that computes the persistent homology of a d-dimensional grayscale digital image $\mathcal{I} : I \to \mathbb{R}$ using the T-construction. The following theorem states that a persistence diagram for the V-construction of $\mathcal{I}$ can be calculated directly from the pairs computed for the negative padded image using the T-construction.

Theorem 3.4.2 (V from T)**.** *For a grayscale digital image $\mathcal{I} : I \to \mathbb{R}$ the diagrams of the V- and T-constructions satisfy*

$$\mathrm{Dgm}_{\mathbf{F}}(V(\mathcal{I})) = \{\, [-q, -p)_{d-k-1} \mid [p, q)_k \in \mathrm{Dgm}_{\mathbf{F}}(T(-\mathcal{I}^{\mathbf{P}})) \,\} \setminus \{\, [\min \mathcal{I}, N)_0 \,\}$$

and

$$\mathrm{Dgm}_{\infty}(V(\mathcal{I})) = \{\, [\min \mathcal{I}, \infty)_0 \,\}.$$

Proof. That $\mathrm{Dgm}(V(\mathcal{I})) = \{\, [\min \mathcal{I}, \infty)_0 \,\}$ follows from the fact that $V(I) \cong D^{(d)}$ and the first cell in the filtration occurs at time $\min \mathcal{I}$. For the finite case, we have that

$$\begin{aligned}
\mathrm{Dgm}_{\mathbf{F}}(V(\mathcal{I})) &= \mathrm{Dgm}_{\mathbf{F}}(V(\mathcal{I}^{\mathbf{P}})) \\
&= \mathrm{Dgm}_{\mathbf{F}}(\widehat{V}(\mathcal{I}^{\mathbf{P}})) \\
&= \{\, [-q, -p)_{d-k-1} \mid [p, q)_k \in \mathrm{Dgm}_{\mathbf{F}}(\widetilde{T}(-\mathcal{I}^{\mathbf{P}})) \,\} \\
&= \{\, [-q, -p)_{d-k-1} \mid [p, q)_k \in \mathrm{Dgm}_{\mathbf{F}}(T(-\mathcal{I}^{\mathbf{P}})) \setminus \{\, [-N, -\min \mathcal{I})_{d-1} \,\} \,\} \\
&= \{\, [-q, -p)_{d-k-1} \mid [p, q)_k \in \mathrm{Dgm}_{\mathbf{F}}(T(-\mathcal{I}^{\mathbf{P}})) \,\} \setminus \{\, [\min \mathcal{I}, N)_0 \,\}
\end{aligned}$$

where the equalities follow from Proposition 3.3.1, Corollary 3.1.5 and Corollary 3.3.4 respectively. □

The structure of the algorithm follows immediately from the theorem and is summarised below.

Algorithm 2 Computing the V-construction persistence diagram with T-construction software.

Require: An image $\mathcal{I}$ and the sub-routine TCON.

1: $\mathrm{Dgm}(V(\mathcal{I})) \leftarrow \{\, [\min(\mathcal{I}), \infty)_0 \,\}$
2: $N \leftarrow \max(\mathcal{I}) + C$ ▷ choose C to ensure $N \gg \max(\mathcal{I})$
3: $-\mathcal{I}^{\mathbf{P}} \leftarrow \textsc{Neg}(\textsc{Pad}(\mathcal{I}, N))$
4: $\mathrm{Dgm}(T(-\mathcal{I}^{\mathrm{P}})) \leftarrow \textsc{Tcon}(-\mathcal{I}^{\mathbf{P}})$ ▷ Apply T-construction software.
5: **for** $[p, q)_k$ in $\mathrm{Dgm}(V(-\mathcal{I}^{\mathbf{P}}))$ with $p \neq -N$ **do**
6: $\quad \mathrm{Dgm}(V(\mathcal{I})) \leftarrow \mathrm{Dgm}(V(\mathcal{I})) \cup \{\, [-q, -p)_{d-k-1} \,\}$
7: **return** $\mathrm{Dgm}(V(\mathcal{I}))$ ▷ Output V-construction persistence diagram.

Example 3.4.3. *Suppose we are working with the two dimensional digital grayscale image given in Figure 3.1 and have only the software to compute the T-construction. We depict the filtration of $T(-\mathcal{I}^{\mathbf{P}})$ in Figure 3.3, and the corresponding intervals in the persistence module. Similarly, we show the filtered V-construction $V(\mathcal{I})$ in Figure 3.4. The reader may confirm that the correspondence between the intervals is accurately described by Theorem 3.4.2.*

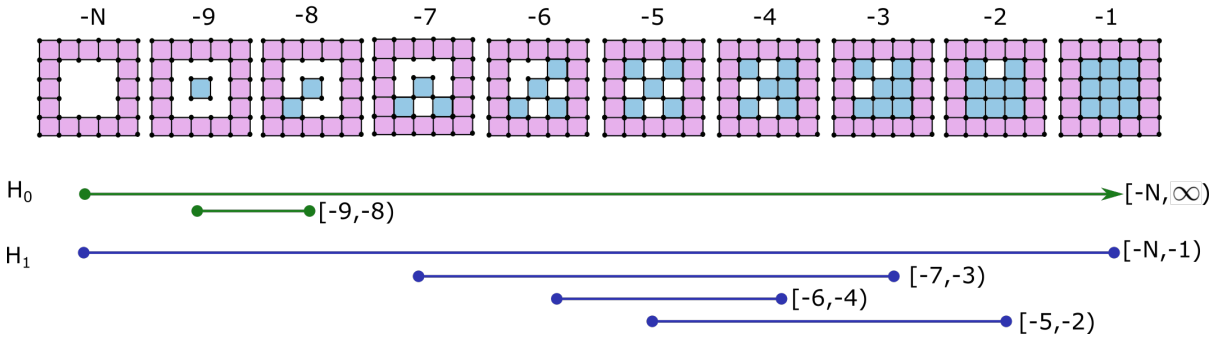

Figure 3.3: The filtration $T(-\mathcal{I}^{\mathbf{P}})$ and intervals of the persistence diagram $\mathrm{Dgm}(T(-\mathcal{I}^{\mathbf{P}}))$ for the image $\mathcal{I} : I \to \mathbb{R}$ of Figure 3.1.

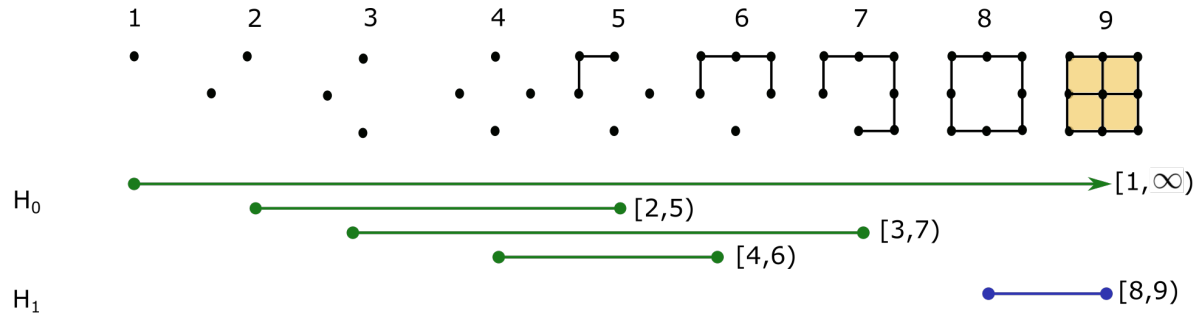

Figure 3.4: The filtration $V(\mathcal{I})$ and intervals of the persistence diagram $\mathrm{Dgm}(V(\mathcal{I}))$ for the image $\mathcal{I} : I \to \mathbb{R}$ of Figure 3.1.

3.5 Discussion

Our results clarify the relationship between the two cubical complex constructions commonly used in digital image analysis software and provide a simple method to use software that implements one construction to compute a persistence diagram for the other. This permits a user's choice of adjacency type for their images to depend on that appropriate to the

application rather than on the type of construction used in available efficient persistence software. In addition to facilitating this application, the results of Sections 3.1 and 3.3 may be of independent interest for the following reasons.

Theorem 3.1.4 is a new interpretation of a duality relationship that manifests in many contexts such as the correspondence between persistent homology and persistent relative cohomology [DSMVJ11], the duality theorem of extended persistence [CSEH09], and a discrete Helmholtz-Hodge decomposition [EÖ20].

The results of Section 3.3 are formulated specifically for the case of an image with its domain homeomorphic to a closed ball, but could be extended to spaces with more interesting topology. We anticipate that the long exact sequence of a pair can be used to derive a relationship between filtered cell complexes that satisfy conditions for duality if their boundaries can be capped or quotiented as in Section 3.2 to obtain a manifold.

As discussed at the beginning of this chapter, the duality and symmetry results of extended persistence do not apply in the cubical setting because the Partition Lemma fails. However, using the T-construction for homology and the V-construction for relative homology (or vice versa), we can derive an extension of extended persistence to cubical filtrations of images satisfying duality and symmetry similar to the results in [CSEH09]. Analyzing properties of this definition and comparing this approach to that in [ES12] could be interesting future work.

There are also interesting questions about algorithm performance to explore. The results of Section 3.4 suggest that persistence diagram computation from grayscale images should have the same average run time independent of the choice of T- and V-construction. If the T-construction executes faster on a particular image, then the V-construction should execute faster on the negative of the image. To answer this question fully requires a careful analysis of the effects of taking the anti-transpose of the boundary matrix on the run time of the matrix reduction algorithm and the extra cells added when padding the image.

CHAPTER 4

The Impact of Changes in Resolution on the Persistent Homology of Images

This chapter is based on [HTS+21b][1]. Therefore, the CC-BY license of this thesis does not apply to this chapter, instead the license of the original IEEE article applies. The extended version of this chapter on arXiv [HTS+21a] contains additional appendices.

Digital images enable quantitative analysis of material properties at micro and macro length scales, but choosing an appropriate resolution when acquiring the image is challenging. A high resolution means longer image acquisition and larger data requirements for a given sample, but if the resolution is too low, significant information may be lost. This chapter studies the impact of changes in resolution on persistent homology, a tool from topological data analysis that provides a signature of structure in an image across all length scales. Given prior information about a function, the geometry of an object, or its density distribution at a given resolution, we provide methods to select the coarsest resolution yielding results within an acceptable tolerance. We present numerical case studies for an illustrative synthetic example and samples from porous materials where the theoretical bounds are unknown.

Prior work. There are several results on point-cloud approximations of manifolds in the context of geometric triangulations built from randomly sampled points near a manifold embedded in $\mathbb{R}^d$, [ACDL00, NSW08, CCSL09, ALS13, KSC+20]. Most of these results start with the assumption that the point-cloud approximation is close to the original object X in the Hausdorff distance. In contrast, the main challenge in Section 4.3 is to establish that the digital approximation is close to the object X in the Hausdorff distance. Once this is achieved, we apply Lemma 4.3.6 — an analogous result to those for point-clouds, but in the setting of digital images and extended to the signed distance transforms.

An earlier paper working with the persistent homology of digital images [BEK10] uses an adaptive grid derived from an oct-tree data structure to reduce data size. Those authors bound the distances between persistence diagrams of the original high-resolution image and their oct-tree approximation, similar in spirit to the results in our Section 4.2 and the numerical case study in Section 4.4.1, but using the adaptive grid in the approximation rather than a single coarser voxel size.

[1] Copyright © 2021 IEEE.

Related work in [DW18] starts with a known continuous function, f, builds an adaptive rectangular subdivision of the domain and uses rigorous computer arithmetic to guarantee their piecewise constant approximation is within ε of f thus guaranteeing the same bound on the bottleneck distances between persistence diagrams.

Papers that have studied how estimates of material properties from micro-CT images change with image resolution include [BS16, HHS21]. Another paper [HLCK21] studies the pore structure of firn (old glacial ice-pack) using two-parameter filtrations derived from the grayscale threshold and distance transform.

Outline. After describing the mathematical model in Section 4.1, we state results in Section 4.2 comparing real-valued functions (not necessarily continuous) with digital approximations for different resolutions. Since we define the digital approximation of f by taking average values within each voxel, these results involve fairly direct bounds on the variation of f and an application of the stability theorem for persistence diagrams.

In Section 4.3 we focus on a particular application where the ground-truth is the continuous signed Euclidean distance transform (CSEDT) derived from an object $X \subset \mathbb{R}^d$. We compare this function with the discrete signed Euclidean distance transform (DSEDT) of a digital approximation to X. This is more complex than simply applying the results of Section 4.2 to the CSEDT of X, because the digital approximation to X may result in small components being lost, and this in turn may lead to a large difference between the CSEDT and the DSEDT. The results and bounds obtained in this section are illustrated with simple examples to show their relevance.

In Section 4.4.1 we present a numerical case study of a synthetic image with structure on different length scales to illustrate circumstances where the theoretical bounds are unknown. Finally, we study some porous materials in Section 4.4.2. These experiments show that the actual data behaves even better than the bounds derived in Section 4.3 suggest.

4.1 Mathematical Models

4.1.1 Digital Images as Discretized Functions

There is a variety of different methods to define digital approximations to algebraic functions whose domain is $\mathbb{R}^d$. Here we consider a tame μ-integrable function $\mathrm{f}\colon \mathbb{R}^d \to \mathbb{R}$ (where μ denotes Lebesgue measure) and take local averages over each voxel.

Let r be the desired spacing for the digital grid and define a voxel σ to be an open d-cube of side length r, and volume $\mu(\sigma) = r^d$. Voxels are indexed by the integer grid $\mathbb{Z}^d$ so that $\sigma(i) = \sigma(i_1, \ldots, i_d) \subset \mathbb{R}^d$ is the product of d open intervals $(i_k r, (i_k + 1)r)$. The *digital approximation* to f is the function

$$\mathrm{f_r} : \mathbb{R}^d \to \mathbb{R},$$

which is a piecewise constant function defined on voxels $\sigma(i)$ with

$$\mathrm{f_r}(x) = r^{-d} \int_{\sigma(i)} \mathrm{f}\, d\mu, \quad \text{if } x \in \sigma(i).$$

On the voxel faces, we define $\mathrm{f_r}(x)$ to be the minimum value taken on voxels σ with $x \in \partial\sigma$.

If we restrict $\mathrm{f_r}$ to the voxel centers indexed by integers $i \in \mathbb{Z}^d$, this discrete function is usually referred to as a *grayscale digital image*. The above definition of $\mathrm{f_r}$ then agrees with

the T-construction for digital images defined in Chapter 3, Definition 3.2.4. The cubical complex and its sublevel sets use the indirect adjacency of digital grids.

In Section 4.2 we use a bound on the differences $|\mathrm{f}(x) - \mathrm{f_r}(x)|$ to obtain a corresponding bound on the bottleneck distance of their persistence diagrams.

4.1.2 Digital Images of Objects and their Distance Transforms

Let $X \subset \mathbb{R}^d$ be a compact subset representing the solid object we approximate with a digital image. By a solid object, we mean $\mathrm{cl}(\mathrm{int}(X)) = X$. Typically $d = 2, 3$, and we can shift X so that it is a subset of a rectangular prism, R.

A *digital approximation to* X is defined using a discrete function $\rho_r : \mathbb{Z}^d \to [0, 1]$ that quantifies the proportion of space occupied by X in each voxel $\sigma(i)$ with

$$\rho_r(i) = \frac{\mu(X \cap \sigma(i))}{\mu(\sigma(i))}.$$

Equivalently, ρ_r is the grayscale digital image obtained from the digital approximation to the indicator function for X.

The digital approximation $X(r, t)$ is then the union of closed voxels of size r, with ρ_r-values at least threshold t with $0 < t \leq 1$,

$$X(r, t) = \bigcup_{\rho_r(i) \geq t} \mathrm{cl}(\sigma(i)).$$

This definition of digital approximation is a simplified model of segmenting an x-ray CT image. If the material consists of just two components with a high degree of x-ray contrast, (e.g., silica and air), then the CT image measures the average x-ray density of each voxel-sized patch of the sample. In this case, x-ray density encodes the proportion of space occupied by the higher-density material. Segmenting converts the grayscale image to a binary image that approximates the distribution in space of the higher-density material. The simplest method of segmentation is to use a single threshold value, as we do here. In the presence of noise and imaging artefacts this method is unsatisfactory, and there is a considerable literature describing practical methods for achieving good segmentations [WS13].

Note that changing the choice of threshold t can cause significant changes to the homology of $X(r, t)$. Fig. 4.1 shows an example where there is no choice of t for which $X(r, t)$ and X have the same homology. The depicted problem of losing thin but possibly large features of an object when the resolution is too low also causes problems in persistent homology. Overcoming this is the main challenge in Section 4.3.

To quantify the size of the holes and other topological and geometric features of X, a suitable function to use is the signed Euclidean distance. The *continuous signed Euclidean distance transform* (CSEDT) of X is defined as

$$d_X^{\mp} \colon \mathbb{R}^d \to \mathbb{R};\ y \mapsto \begin{cases} -d(y, \partial X) & \text{if } y \in X \\ d(y, \partial X) & \text{if } y \in X^{\mathrm{c}}, \end{cases}$$

where ∂X denotes the boundary of X and X^{c} denotes the complement of X in $\mathbb{R}^d$.

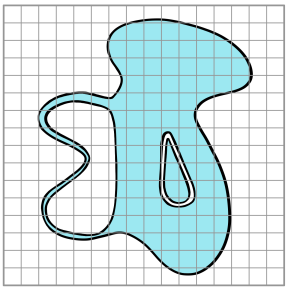

Figure 4.1: An object X in blue, with a voxel grid of side r overlayed in gray. There is no t that makes $X(r,t)$ have the same homology as X. If t is close to 1, the handle-shaped part of X is lost. If t is small, then the narrow annulus in the middle will be filled in.

In practice, since we want to compare $d_X^{\mp}$ and its discretized version, we restrict the distance transforms to a rectangular prism R that contains X. The discrete analog of the CSEDT is the discrete signed Euclidean distance transform (DSEDT)

$$D[X(r,t)] : R \to \mathbb{R}$$

This is a piecewise constant function defined on voxels $\sigma(i) \subseteq X(r,t)$ and $\sigma(j) \subseteq X(r,t)^{\mathrm{c}}$ by

$$x \mapsto \begin{cases} -\min_{\sigma(j)\subseteq X(r,t)^{\mathrm{c}}} r\, d(i,j) & \text{if } x \in \sigma(i) \subseteq X(r,t) \\ \min_{\sigma(i)\subseteq X(r,t)} r\, d(i,j) & \text{if } x \in \sigma(j) \subseteq X(r,t)^{\mathrm{c}}. \end{cases}$$

On voxel faces, $D[X(r,t)]$ takes the minimum value over all voxels adjacent to the given face. Again, this corresponds to the T-construction mentioned earlier.

In Section 4.3, we investigate conditions on the geometry of X and the voxel size r necessary to control the distance between the persistence diagrams of the CSEDT of X and the DSEDT of $X(r,t)$. Even though the DSEDT depends on the threshold t, the bounds we obtain do not depend on t. In other words, even though a wise choice of t might help to achieve a better approximation at a larger voxel size, we can guarantee a good approximation for any value of $t > 0$ once r is chosen small enough compared to geometric characteristics of X, such as the *reach*.

Intuitively, the reach of a closed subset $A \subseteq \mathbb{R}^d$ encodes the minimum distance at which two or more fire fronts meet after A is set on fire [Blu67]. Mathematically, the reach of a closed set $A \subseteq \mathbb{R}^d$ is the largest ε (possibly ∞) such that for every $p \in \mathbb{R}^d$ with distance $d(p,A) < \varepsilon$, A contains a unique point, $\xi_A(p)$, nearest to p, i.e. $d(p,A) = d(p,\xi_A(p))$ [Fed59]. In this thesis we use only one property of the reach, proven by Federer in [Fed59, Theorem 4.8 (12)]; see Lemma 4.1.3. The lemma states that when walking from a point $a \in \partial A$ orthogonally a distance $r < \operatorname{reach}(A)$ away from the closed set A, then the closest point of A is still the starting point a, and thus the distance to A is r. Before we can state this rigorously in Lemma 4.1.3, we need formal definitions [Fed59, Definitions 4.3 and 4.4] of tangent vectors and normal vectors:

Definition 4.1.1 (Tangent vector)**.** Let $A \subseteq \mathbb{R}^d$ be closed and $a \in \partial A$. Then $u \in \mathbb{R}^d$ is a *tangent vector of A at a* if either $u = 0$ or for every $\varepsilon > 0$ there exists a point $b \in A$ with

$$0 < \|b-a\| < \varepsilon \text{ and } \left\| \frac{b-a}{\|b-a\|} - \frac{u}{\|u\|} \right\| < \varepsilon.$$

Definition 4.1.2 (Normal vector)**.** Let $A \subseteq \mathbb{R}^d$ be closed and $a \in \partial A$. Then $v \in \mathbb{R}^d$ is a *normal vector of A at a* if for every tangent vector u of A at a, the scalar product $v \cdot u$ is non-positive.

Lemma 4.1.3 (Federer 1959)**.** *Let $A \subseteq \mathbb{R}^d$ be closed and $a \in \partial A$. Let $reach(A) > r > 0$. Let v be a normal vector of A at a. Then,*

$$d(a + r\frac{v}{\|v\|}, A) = d(a + r\frac{v}{\|v\|}, a) = r.$$

Note that the $\mathrm{reach}(A)$ characterizes the geometry of the complement of A. Since we work with the signed Euclidean distance transform of a solid object X in this chapter, we will use

$$\mathrm{reach}(\partial X) = \min\{\mathrm{reach}(X), \mathrm{reach}(\mathrm{cl}(X^{\mathrm{c}}))\}$$

to characterize the geometry of both X and its complement. The geometric attributes that determine the reach of ∂X are its radii of curvature and distances to the generalised critical points of the signed Euclidean distance transform. It is known [Thä08], that if ∂X is a closed C^2 submanifold of $\mathbb{R}^d$, then $\mathrm{reach}(\partial X) > 0$.

In Section 4.3 we establish a bound on the difference between the CSEDT of X and the DSEDT of $X(r,t)$ in terms of a geometric quantity we call the leash. We denote by A_δ, the set of all points within distance δ of a set A. Recall now, that the *erosion* of X by balls of radius δ is given by

$$\mathrm{E}_\delta(X) = ((X^{\mathrm{c}})_\delta)^{\mathrm{c}} = \{x \in X \mid B_\delta(x) \subseteq X\}.$$

The *leash* then measures the Hausdorff distance between X and $E_\delta(X)$:

$$\mathrm{leash}_X(\delta) = d_{\mathcal{H}}(X, \mathrm{E}_\delta(X)) = \sup_{x \in X} d(x, \mathrm{E}_\delta(X)).$$

The name leash can be best understood by imagining an infinitesimally small dog connected by a leash to the center of a ball of radius s, see Fig. 4.2. The ball must stay fully inside the set A, while the dog is free to visit the whole of $\mathbb{R}^d$. The minimal length of the leash such that the dog can access every point of A is then $\mathrm{leash}_A(s)$.

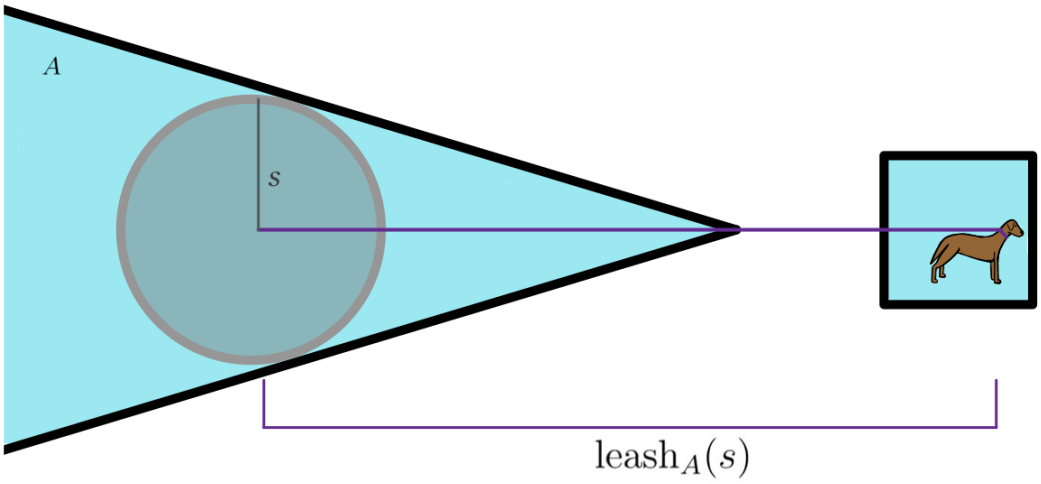

Figure 4.2: Illustration of the $\mathrm{leash}_A(s)$: The ball of radius s gets stuck inside part of A but with a long enough leash the dog can reach every part of A. © 2021 IEEE.

In Lemma 4.3.8, we show that if the reach is strictly positive and $0 < s < \operatorname{reach}(\operatorname{cl}(X)^{\mathrm{c}})$, then $\operatorname{leash}_X(s) = s$. This is equivalent to saying that the closed s-neighborhood of $E_s(X)$ exactly recovers X.

As for the reach, we need to characterize both the geometry of X and its complement X^c, so we will use the two-sided leash, defined as

$$l_X(\delta) = \max\{\operatorname{leash}_X(\delta), \operatorname{leash}_{\operatorname{cl}(X^{\mathrm{c}})}(\delta)\}.$$

4.2 Persistent Homology of Grayscale Images

Using the definitions and tools outlined in Section 4.1, we compare the persistence diagrams of a real-valued function to the persistence diagrams of its digital approximation at a given voxel size. This is achieved by bounding the difference between the function and its approximation and invoking the stability theorem for persistence diagrams in Proposition 4.2.1. This result has a number of simple corollaries giving us bounds on the differences between persistence diagrams of digital approximations for different voxel sizes, and for the case that the ground-truth function is Lipschitz continuous.

Proposition 4.2.1. *Let* f *be a tame* μ*-integrable function, and* $\mathrm{f_r}$ *its digital approximation on a grid with spacing* r*. Suppose there is a positive number* $M_r > 0$ *such that for every closed voxel* $\operatorname{cl}(\sigma(i))$ *in the domain of* f*, the difference*

$$\sup_{x \in \operatorname{cl}(\sigma(i))} \mathrm{f}(x) - \inf_{x \in \operatorname{cl}(\sigma(i))} \mathrm{f}(x) \leq M_r.$$

Then, $\|\mathrm{f} - \mathrm{f_r}\|_\infty \leq M_r$ *and thus* $\mathrm{d_B}(\mathrm{Dgm}(\mathrm{f}), \mathrm{Dgm}(\mathrm{f_r})) \leq M_r$*.*

Proof. For any x in the domain of f, there is at least one voxel $\sigma(i)$ with side-length r such that $x \in \operatorname{cl}(\sigma(i))$. If x belongs to more than one closed voxel, choose the one for which $\mathrm{f_r}(x) = \mathrm{f_r}(y)$ for $y \in \sigma(i)$. Let

$$M_x := \sup_{y \in \operatorname{cl}(\sigma(i))} \mathrm{f}(y), \quad \text{and} \quad m_x := \inf_{y \in \operatorname{cl}(\sigma(i))} \mathrm{f}(y).$$

We have $M_r \geq M_x - m_x$. The definition of $\mathrm{f_r}$ and choice of $\sigma(i)$ ensures that $m_x \leq \mathrm{f_r}(x) \leq M_x$. It follows that

$$\begin{aligned} |\mathrm{f}(x) - \mathrm{f_r}(x)| &\leq \max\{\mathrm{f}(x) - m_x, M_x - \mathrm{f}(x)\} \\ &\leq M_x - m_x \\ &\leq M_r. \end{aligned}$$

We conclude that $\|\mathrm{f} - \mathrm{f_r}\|_\infty \leq M_r$. Thus, by the stability theorem [CSEH07] the bottleneck distance between the persistence diagrams is also bounded by M_r. □

We now compare the persistence diagrams of the digital approximations $\mathrm{f_r}$ at different resolutions. Given two grid spacings $r_1 > 0$ and $r_2 > 0$, suppose r_2 is divisible by r_1, i.e., $a := r_2/r_1 \in \mathbb{N}$, then any voxel of size r_2 contains a^d voxels of size r_1. Since the measure of the voxel faces is zero, f_{r_2} is the r_2 digital approximation of f_{r_1}. Therefore, Proposition 4.2.1, applied to f_{r_1}, implies an upper bound for the distance between the persistence diagrams of the two digital approximations f_{r_1} and f_{r_2} as follows.

Corollary 4.2.2. *Suppose $r_2 > r_1 > 0$, $a = r_2/r_1 \in \mathbb{N}$, and choose $M \geq 0$ such that for all voxels $\sigma(i)$ of size r_2, the difference*

$$\max_{\sigma' \subseteq \sigma(i)} \{\mathrm{f}_{r_1}(x) : x \in \mathrm{cl}(\sigma')\} - \min_{\sigma' \subseteq \sigma(i)} \{\mathrm{f}_{r_1}(x) : x \in \mathrm{cl}(\sigma')\} \leq M,$$

where σ' refers to voxels of size r_1.
Then, $\|\mathrm{f}_{r_1} - \mathrm{f}_{r_2}\|_\infty \leq M$ and thus $\mathrm{d_B}(\mathrm{Dgm}(\mathrm{f}_{r_1}), \mathrm{Dgm}(\mathrm{f}_{r_2})) \leq M$.

For the special case when f is Lipschitz continuous, we can estimate the constant M in Proposition 4.2.1 by the Lipschitz constant of f. Recall that a function $\mathrm{f}\colon \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous with constant $L > 0$ if $|\mathrm{f}(x) - \mathrm{f}(y)| \leq L\|x - y\|$. For brevity, we say f is L-Lipschitz continuous. Note that the CSEDT $d_X^{\mp}$ is Lipschitz continuous with constant 1.

Corollary 4.2.3. *Suppose $\mathrm{f}\colon \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz continuous. Then, $\|f - f_r\|_\infty \leq Lr\sqrt{d}$ and thus $\mathrm{d_B}(\mathrm{Dgm}(\mathrm{f}), \mathrm{Dgm}(\mathrm{f_r})) \leq Lr\sqrt{d}$.*

Proof. For any voxel $\sigma(i)$, since the function f is continuous and $\mathrm{cl}(\sigma(i))$ is compact, there are $x_1, x_2 \in \mathrm{cl}(\sigma(i))$ such that

$$\mathrm{f}(x_1) = \max_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y), \quad \text{and} \quad \mathrm{f}(x_2) = \min_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y).$$

It follows that

$$\begin{aligned} \max_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y) - \min_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y) &= |\mathrm{f}(x_1) - \mathrm{f}(x_2)| \\ &\leq L\|x_1 - x_2\| \leq Lr\sqrt{d}. \end{aligned}$$

From Proposition 4.2.1, we see that $\|f - f_r\|_\infty \leq Lr\sqrt{d}$ and thus the bottleneck distance between the persistence diagrams is bounded by $Lr\sqrt{d}$. □

Under the Lipschitz continuity assumption, the bottleneck distance between persistence diagrams of two digital approximations is bounded as follows.

Corollary 4.2.4. *Suppose $\mathrm{f}\colon \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz continuous. Then, $\|\mathrm{f}_{r_1} - \mathrm{f}_{r_2}\|_\infty \leq Lr_2\sqrt{d}$ and thus $\mathrm{d_B}(\mathrm{Dgm}(\mathrm{f}_{r_1}), \mathrm{Dgm}(\mathrm{f}_{r_2})) \leq Lr_2\sqrt{d}$, when $r_2 > r_1 > 0$ and r_2 is divisible by r_1.*

Proof. For any voxel $\sigma(i)$ of size r_2, since the function f is continuous and $\mathrm{cl}(\sigma(i))$ is compact, there are x_1 and x_2 in $\mathrm{cl}(\sigma(i))$ such that

$$\mathrm{f}(x_1) = \max_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y), \quad \text{and} \quad \mathrm{f}(x_2) = \min_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y).$$

Note that

$$\begin{aligned} \max_{\sigma' \subset \sigma(i)} \left\{\mathrm{f}_{r_1}(x) : x \in \mathrm{cl}(\sigma')\right\} &\leq \max_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y), \\ \min_{\sigma' \subset \sigma(i)} \left\{\mathrm{f}_{r_1}(x) : x \in \mathrm{cl}(\sigma')\right\} &\geq \min_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y). \end{aligned}$$

Therefore, the constant M defined in Proposition 4.2.2 is bounded as

$$\begin{aligned} M &\leq \max_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y) - \min_{y \in \mathrm{cl}(\sigma(i))} \mathrm{f}(y) \\ &= \mathrm{f}(x_1) - \mathrm{f}(x_2) \leq L||x_1 - x_2|| \leq Lr_2\sqrt{d}. \end{aligned}$$

Hence, by Proposition 4.2.2, $\|\mathrm{f}_{r_1} - \mathrm{f}_{r_2}\|_\infty \leq Lr_2\sqrt{d}$ and thus $\mathrm{d_B}(PD(\mathrm{f}_{r_1}), PD(\mathrm{f}_{r_2})) \leq Lr_2\sqrt{d}$. □

When r_2 is not an integer multiple of r_1, it is still possible to bound the bottleneck distance between the persistent diagrams of f_{r_1} and f_{r_2} using Proposition 4.2.1 and triangle inequality. However, in this case, the bound is not as tight as the bound in Corollary 4.2.4.

Corollary 4.2.5. *Suppose* $\mathrm{f}\colon \mathbb{R}^d \to \mathbb{R}$ *is* L*-Lipschitz continuous. Then,* $\|\mathrm{f}_{r_1} - \mathrm{f}_{r_2}\|_\infty \leq L(r_1 + r_2)\sqrt{d}$ *and thus* $\mathrm{d_B}(\mathrm{Dgm}(\mathrm{f}_{r_1}), \mathrm{Dgm}(\mathrm{f}_{r_2})) \leq L(r_1 + r_2)\sqrt{d}$*, for all* $r_1, r_2 > 0$*.*

Proof. From triangle inequality and Corollary 4.2.3, we have

$$\begin{aligned} ||f_{r_1} - f_{r_2}||_\infty &\leq ||f_{r_1} - f||_\infty + ||f_{r_2} - f||_\infty \\ &\leq Lr_1\sqrt{d} + Lr_2\sqrt{d} = L(r_1 + r_2)\sqrt{d}. \end{aligned}$$

By the stability theorem, the bottleneck distance between the persistence diagrams is also bounded by $L(r_1 + r_2)\sqrt{d}$. □

4.3 Persistent Homology of the Distance Transform of Binary Images

In this section, we consider $X \subseteq \mathbb{R}^d$ as a solid object that is imaged with voxel spacing r, yielding its digital approximation $X(r,t)$ for some choice of $0 < t \leq 1$. We derive bounds on the bottleneck distance between the persistence diagram of the DSEDT of $X(r,t)$, and the CSEDT of X, by first comparing both of them to the CSEDT of $X(r,t)$, as outlined in Fig. 4.3.

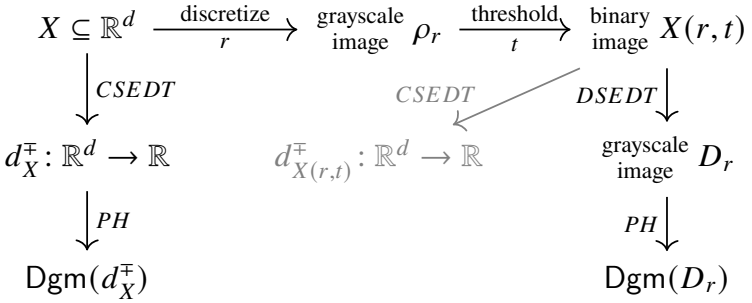

Figure 4.3: Diagram of our model for the digital approximation of a solid object X, and its geometric characterisation using signed Euclidean distance transforms and persistent homology. The top row is a simplified version of CT-imaging and segmentation. In gray: In the proof of Theorem 4.3.1, instead of comparing the continuous distance transform $d_X^{\mp}$ to the discrete distance transform D_r directly, we compare both to the continuous distance transform $d_{X(r,t)}^{\mp}$ of the discrete object. © 2021 IEEE.

4.3.1 Bounds using the reach or leash of X

The most relevant result of this section is Corollary 4.3.9, which states that when discretizing with voxel length $r < \frac{1}{\sqrt{d}}\mathrm{reach}(\partial X)$, the bottleneck distance is bounded by a constant times r. This follows directly from Theorem 4.3.1, which bounds the bottleneck distance in terms

of the two-sided leash $l_X(\sqrt{d}r)$ when discretizing with any voxel length r, not necessarily smaller than the reach. Note that the bounds we give here do not depend on the threshold t. The results could be slightly improved by incorporating t but we do not do so for the sake of brevity.

Theorem 4.3.1. *Given $X \subseteq \mathbb{R}^d$ with $X = \mathrm{cl}(\mathrm{int}(X))$. Let $d_X^{\mp}$ be the CSEDT of X, $r > 0$, $t \in (0, 1]$, and $X(r,t)$ be its digital approximation. Let $D_r = D[X(r,t)] \colon \mathbb{R}^d \to \mathbb{R}$ denote the discrete signed Euclidean distance transform. Then,*

$$\mathrm{d_B}(\mathrm{Dgm}(d_X^{\mp}), \mathrm{Dgm}(D_r)) \leq l_X(\sqrt{d}r) + 2\sqrt{d}r$$

The proof requires the following lemmata. The first lemma shows that the DSEDT of $X(r,t)$ is a good approximation to the CSEDT of $X(r,t)$.

Lemma 4.3.2. *With the notation of Theorem 4.3.1,*

$$\left\| d_{X(r,t)}^{\mp} - D_r \right\|_\infty \leq \sqrt{d}r.$$

Proof. Let $p \in \mathbb{R}^d$ and assume $p \in X(r,t)$; the proof for $p \in X(r,t)^{\mathrm{c}}$ works analogously. Let y be a closest point of $\partial X(r,t)$ to p, i.e. $d_{X(r,t)}^{\mp}(p) = -d(p, \partial X(r,t)) = -d(p,y)$, see Fig. 4.4. As $y \in \partial X(r,t)$, there exists a voxel $\sigma(i) \subseteq X(r,t)^{\mathrm{c}}$ with voxel center y' of distance

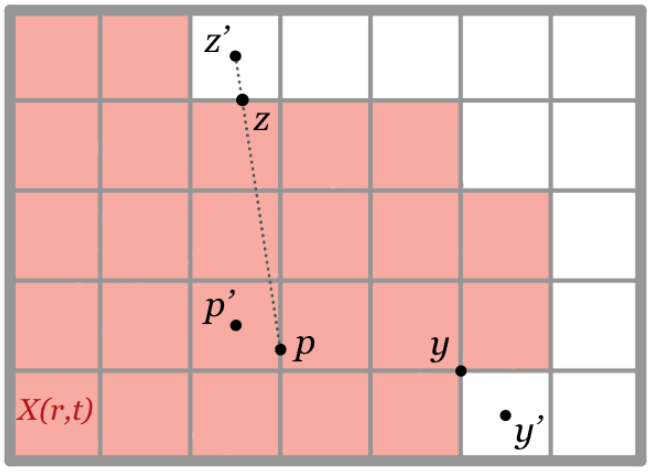

Figure 4.4: Illustration of the variables appearing in the proof of Lemma 4.3.2. © 2021 IEEE.

$d(y,y') \leq \frac{1}{2}\sqrt{d}r$. Let $\sigma(j)$ be the voxel with p in its closure with minimal filtration value (in case there are several), i.e. the voxel giving filtration value $D_r(p) = D_r(\sigma(j))$ to p, and let p' be its center. Thus, $d(p,p') \leq \frac{1}{2}\sqrt{d}r$. Hence,

$$\begin{aligned} -D_r(p) &= -D_r(\sigma(j)) \\ &= \min\{d(p',c) \mid c \text{ voxel center of } \sigma(k) \subseteq X(r,t)^{\mathrm{c}}\} \\ &\leq d(p',y') \leq d(p',p) + d(p,y) + d(y,y') \\ &\leq \frac{1}{2}\sqrt{d}r - d_{X(r,t)}^{\mp}(p) + \frac{1}{2}\sqrt{d}r, \end{aligned}$$

yielding $d_{X(r,t)}^{\mp}(p) - D_r(p) \leq \sqrt{d}r$.

Let z' be the voxel center minimizing

$$\min\{d(p', c) \mid c \text{ voxel center of } \sigma(k) \subseteq X(r,t)^{\mathrm{c}}\} = -D_r(p).$$

As $z' \in X(r,t)^{\mathrm{c}}$ and $p \in X(r,t)$, there exists a $z \in \partial X(r,t)$ on the straight line segment between z' and p. Hence,

$$\begin{aligned} -d^{\mp}_{X(r,t)}(p) &= d(p, \partial X(r,t)) \leq d(p,z) \leq d(p,z') \\ &\leq d(p,p') + d(p',z') \leq \frac{1}{2}\sqrt{d}r - D_r(p), \end{aligned}$$

yielding $-(d^{\mp}_{X(r,t)}(p) - D_r(p)) \leq \frac{1}{2}\sqrt{d}r$. □

The authors of [CSEH07] mention that, by definition, the Hausdorff distance between two sets equals the L_∞-distance between the (unsigned) Euclidean distance transforms of these two sets. For the proof of Theorem 4.3.1 an extension of this to the *signed* Euclidean distance transform is needed, see Lemma 4.3.3. Note that the bound is between the maximum and the sum of the two terms $d_{\mathcal{H}}(A,B)$ and $d_{\mathcal{H}}(A^{\mathrm{c}}, B^{\mathrm{c}})$.

Lemma 4.3.3. *Let $A, B \subseteq \mathbb{R}^d$, then*

$$\begin{aligned} \left\| d^{\mp}_A - d^{\mp}_B \right\|_\infty \leq \max\{ &\sup_{b \in B^c} d(A^c, b) + \sup_{a \in A} d(a, B), \\ &\sup_{b \in B} d(A, b) + \sup_{a \in A^c} d(a, B^c)\} \end{aligned}$$

Proof. Let d_A denote the unsigned Euclidean distance transform defined as $d_A(p) = d(p, A)$, for any $p \in \mathbb{R}^d$. As mentioned above, we have by definition that $\|d_A - d_B\|_\infty = d_{\mathcal{H}}(A,B)$. To bound $\left\|d^{\mp}_A - d^{\mp}_B\right\|_\infty$, we consider four cases for the point $p \in \mathbb{R}^d$.

Case 1: $p \in A,\ p \in B$. Then, $|d^{\mp}_A(p) - d^{\mp}_B(p)| = |-d_{A^{\mathrm{c}}}(p) + d_{B^{\mathrm{c}}}(p)| \leq \|d_{B^{\mathrm{c}}} - d_{A^{\mathrm{c}}}\|_\infty = d_{\mathcal{H}}(A^{\mathrm{c}}, B^{\mathrm{c}}) = \max\{\sup_{b \in B^{\mathrm{c}}} d(A^{\mathrm{c}}, b),\ \sup_{a \in A^{\mathrm{c}}} d(a, B^{\mathrm{c}})\}$.

Case 2: $p \notin A,\ p \notin B$. Analogously, $|d^{\mp}_A(p) - d^{\mp}_B(p)| \leq \max\{\sup_{a \in A} d(a,B), \sup_{b \in B} d(A,b)\}$.

Case 3: $p \in A,\ p \notin B$. Then, $|d^{\mp}_A(p) - d^{\mp}_B(p)| = |-d(A^{\mathrm{c}}, p) - d(p,B)| = d(A^{\mathrm{c}}, p) + d(p,B) \leq \sup_{b \in B^{\mathrm{c}}} d(A^{\mathrm{c}}, b) + \sup_{a \in A} d(a,B)$.

Case 4: $p \notin A,\ p \in B$. Analogously, $|d^{\mp}_A(p) - d^{\mp}_B(p)| \leq \sup_{b \in B} d(A,b) + \sup_{a \in A^{\mathrm{c}}} d(a, B^{\mathrm{c}})$. □

Remark 4.3.4. *In the special case that $\max\{d_{\mathcal{H}}(A,B), d_{\mathcal{H}}(A^c, B^c)\} < reach(\partial A)$, Lemma 4.3.3 can be tightened to $\left\|d^{\mp}_A - d^{\mp}_B\right\|_\infty \leq \max\{d_{\mathcal{H}}(A,B), d_{\mathcal{H}}(A^c, B^c)\}$, see Appendix B in the extended version of this chapter on arXiv [HTS+21a].*

However, when the sets differ by more than their reach, the bound in Lemma 4.3.3 can be tight, as Example 4.3.5 shows.

Example 4.3.5. *Let $A = [-1.5, 1.5] \times \mathbb{R}$ and $B = [-1.5, 1.5] \times ((-\infty, -1] \cup [1, \infty))$. Then, the bound given in Lemma 4.3.3 is tight: $\left\|d^{\mp}_A - d^{\mp}_B\right\|_\infty = |d^{\mp}_A(0) - d^{\mp}_B(0)| = |-1.5 - 1| = 1.5 + 1 = \sup_{b \in B^c} d(A^c, b) + \sup_{a \in A} d(a,B)$.*

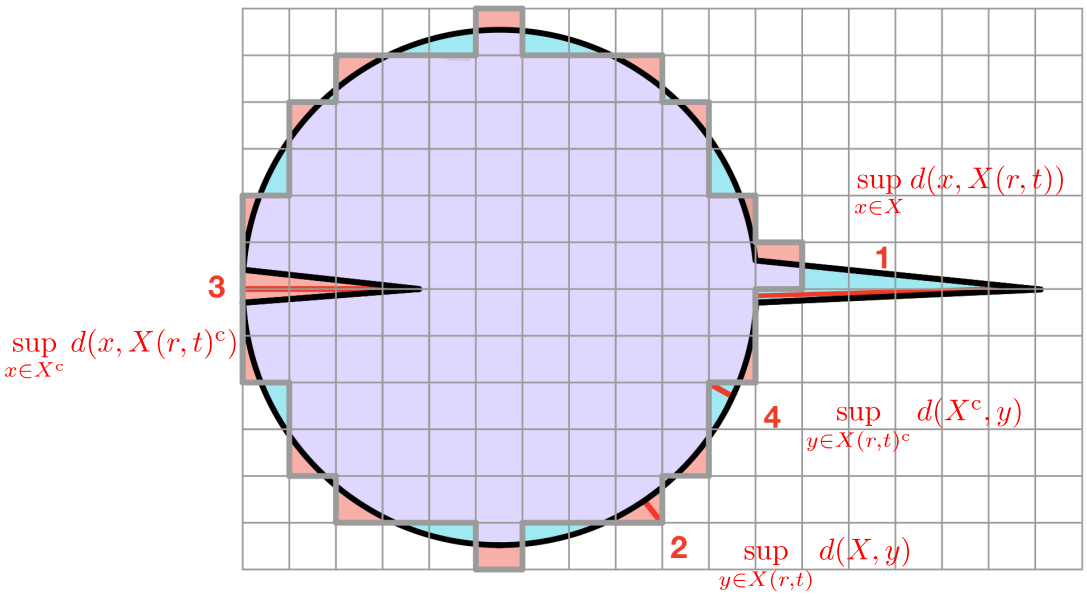

Figure 4.5: The four different suprema in the bound of Lemma 4.3.3 can all have different values. In the proof of Theorem 4.3.1 we show that suprema 1 and 3 are bounded by the two-sided leash $l_X(\sqrt{d}r)$, while suprema 2 and 4 are bounded by the voxel diameter $\sqrt{d}r$. © 2021 IEEE.

Lemma 4.3.6. *With the notation of Theorem 4.3.1,*

$$\begin{aligned}
&\mathrm{d_B}(\mathrm{Dgm}(d_X^{\mp}), \mathrm{Dgm}(D_r)) \\
&\leq \max\{ \sup_{y\in X(r,t)^c} d(X^c, y) + \sup_{x\in X} d(x, X(r,t)), \\
&\qquad \sup_{y\in X(r,t)} d(X, y) + \sup_{x\in X^c} d(x, X(r,t)^c)\} + \sqrt{d}r
\end{aligned}$$

Proof. The proof is a combination of the stability theorem of persistent homology [CSEH07], the triangle inequality, and Lemmata 4.3.3, 4.3.2:

$$\begin{aligned}
&\mathrm{d_B}(\mathrm{Dgm}(d_X^{\mp}), \mathrm{Dgm}(D_r)) \\
&\leq \left\| d_X^{\mp} - D_r \right\|_\infty \\
&\leq \left\| d_X^{\mp} - d_{X(r,t)}^{\mp} \right\|_\infty + \left\| d_{X(r,t)}^{\mp} - D_r \right\|_\infty \\
&\leq \max\{ \sup_{y\in X(r,t)^c} d(X^c, y) + \sup_{x\in X} d(x, X(r,t)), \\
&\qquad \sup_{y\in X(r,t)} d(X, y) + \sup_{x\in X^c} d(x, X(r,t)^c)\} + \sqrt{d}r.
\end{aligned}$$

□

Lemma 4.3.7. *With the notation of Theorem 4.3.1,*

1. $\sup_{x\in X} d(x, X(r,t)) \leq leash_X(\sqrt{d}r)$
2. $\sup_{x\in X^c} d(x, X(r,t)^c) \leq leash_{\mathrm{cl}(X^c)}(\sqrt{d}r)$
3. $\sup_{y\in X(r,t)} d(X, y) \leq \sqrt{d}r$

4. $\sup_{y\in X(r,t)^c} d(X^c, y) \leq \sqrt{d}r$

Proof. 1) To prove $\sup_{x\in X} d(x, X(r,t)) \leq \mathrm{leash}_X(\sqrt{d}r)$, let $x \in X$ be arbitrary. By the definition of $\mathrm{leash}_X(\sqrt{d}r)$, there exists a point $a \in ((X^c)_{\sqrt{d}r})^c$ with $d(x,a) \leq \mathrm{leash}_X(\sqrt{d}r)$. As a is at least $\sqrt{d}r$ far from X^c, the open ball $B_{\sqrt{d}r}(a)$ is fully inside X. Let σ be a voxel containing a in its closure. As the diameter of the voxel is $\mathrm{diam}(\sigma) = \sqrt{d}r$, the voxel $\sigma \subseteq B_{\sqrt{d}r}(a) \subseteq X$ has density $\rho_r(\sigma) = 1 \geq t$, for all $t \in (0,1]$. Hence, $a \in \mathrm{cl}(\sigma) \subseteq X(r,t)$ and $d(x, X(r,t)) \leq d(x,a) \leq \mathrm{leash}_X(\sqrt{d}r)$.

2) The proof for $\sup_{x\in X^c} d(x, X(r,t)^c) \leq \mathrm{leash}_{\mathrm{cl}(X^c)}(\sqrt{d}r)$ is analogous.

3) To prove $\sup_{y\in X(r,t)} d(X, y) \leq \sqrt{d}r$, let $y \in X(r,t)$ be arbitrary. Let $\sigma \subseteq X(r,t)$ be such that $y \in \mathrm{cl}(\sigma)$. As $\rho_r(\sigma) > t > 0$, there exists a point $x \in X \cap \sigma$. Hence, $d(X,y) \leq d(x,y) \leq \mathrm{diam}(\sigma) = \sqrt{d}r$.

4) The proof for $\sup_{y\in X(r,t)^c} d(X^c, y) \leq \sqrt{d}r$ is analogous. □

Proof of Theorem 4.3.1. Lemma 4.3.7 shows that both the terms $\sup_{y\in X(r,t)^c} d(X^c, y) + \sup_{x\in X} d(x, X(r,t))$ and $\sup_{y\in X(r,t)} d(X,y) + \sup_{x\in X^c} d(x, X(r,t)^c)$ are bounded by $\sqrt{d}r + l_X(\sqrt{d}r)$. In combination with Lemma 4.3.6 this finishes the proof. □

In the special case when the voxel diameter $\sqrt{d}r$ is smaller than $\mathrm{reach}(\partial X)$, we can calculate the term $l_X(\sqrt{d}r)$ explicitly using the following Lemma.

Lemma 4.3.8. *Let A be a subset of $\mathbb{R}^d$ with $A = \mathrm{cl}(\mathrm{int}(A))$ and $reach(\mathrm{cl}(A^c)) > 0$. Let $s < reach(\mathrm{cl}(A^c))$, then $leash_A(s) = s$.*

Proof. To show that $\mathrm{leash}_A(s) = \sup_{a\in A} d(a, \mathrm{E}_s(A))$ is at most s, we need to find for every $a \in A$ an element $b \in \mathrm{E}_s(A)$ with distance $d(a,b)$ at most s. Let $a \in A$ be arbitrary. If $a \in \mathrm{E}_s(A)$, then $b = a$ fulfills $d(a,b) = 0 \leq s$. Therefore $a \notin \mathrm{E}_s(A)$, i.e. $a \in (A^c)_s$. Let $x \in \partial\,\mathrm{cl}(A^c)$ be s.t. $d(a,x) = d(a, \mathrm{cl}(A^c))$. If $x \neq a$, then $n_x = \frac{a-x}{\|a-x\|}$ is a unit normal vector of $\mathrm{cl}(A^c)$ at x (for a rigorous definition of normal vector, see Definition 4.1.2). If $x = a$, let n_x be any unit normal vector of $\mathrm{cl}(A^c)$ at x. Define $b = x + sn_x$. Since $s < \mathrm{reach}(\mathrm{cl}(A^c))$, Federer's Lemma 4.1.3 yields, $d(A^c, b) = d(\mathrm{cl}(A^c), b) = d(x,b) = s$. Hence, $b \notin (A^c)_s$ and thus $b \in ((A^c)_s)^c = \mathrm{E}_s(A)$. The points x, a, and b lie on a straight line by construction. The point a has distance $d(a,x) = d(a, \mathrm{cl}(A^c))) = d(a, A^c)$ to x, which is less than s as $a \in (A^c)_s$. Hence, a lies on the line segment between x and b, and thus $d(a,b) \leq d(x,b) = s$, concluding the proof of $\mathrm{leash}_A(s) \leq s$.

For any $a \in \partial A$, we get $d(a, \mathrm{E}_s(A)) \geq s$, yielding $\mathrm{leash}_A(s) = s$. □

Corollary 4.3.9. *Using the notation of Theorem 4.3.1. If $r < \frac{1}{\sqrt{d}} reach(\partial X)$, then*

$$\mathrm{d_B}(\mathrm{Dgm}(d_X^{\mp}), \mathrm{Dgm}(D_r)) \leq 3\sqrt{d}r$$

Proof. Apply Theorem 4.3.1 and Lemma 4.3.8. □

Remark 4.3.10. *Using Remark 4.3.4, the constant 3 in Corollary 4.3.9 can be tightened to 2, see Appendix B in the extended version of this chapter on arXiv [HTS+21a].*

4.3.2 Bounds using the density ρ_r

Suppose we start with ρ_r, the function that tells us the proportion of X in each voxel defined in Section 4.1.2. As mentioned there, this is a reasonable model of the information contained in an x-ray CT image of a two-phase material with high x-ray contrast. Theorem 4.3.12 bounds the bottleneck distance between the persistence diagrams of $d_X^{\mp}$ and D_r in terms of ρ_r. The proof reuses Lemma 4.3.6, but we need to bound the suprema by quantities other than the leash this time.

Lemma 4.3.11. *Using the notation of Theorem 4.3.1.*

1. $\sup_{x\in X} d(x, X(r,t)) \leq \max\left(0, \max_{\rho_r(i)>0} D_r(\sigma(i))\right)$
2. $\sup_{x\in X^c} d(x, X(r,t)^c) \leq \max\left(0, \max_{\rho_r(i)<1} -D_r(\sigma(i))\right)$.

Proof. We prove the first statement here; the second can be shown similarly. Let $x \in X$, and assume $x \notin X(r,t)$ (otherwise the statement follows trivially). As $x \in X = \mathrm{cl}(\mathrm{int}(X))$, any voxel $\sigma(j)$ containing x in its closure has $\rho_r(j) > 0$. Let x' be the center of $\sigma(j)$. As $x \notin X(r,t)$, its voxel $\sigma(j)$ is in $X(r,t)^{\mathrm{c}}$. Let y' be the voxel center minimizing

$$\min\{d(x', c) \mid c \text{ voxel center } \in X(r,t)\} = D_r(\sigma(j)).$$

As $x - x'$ is a vector pointing from a voxel center to a point in the closure of the same voxel, adding this vector to the voxel center y' yields a point in the closure of the same voxel, and thus a point in $X(r,t)$. Hence,

$$\begin{aligned} d(x, X(r,t)) &\leq d(x, y' + (x - x')) \\ &= d(x - (x - x'), y') \\ &= d(x', y') \\ &= \min\{d(x', c) \mid c \text{ voxel center } \in X(r,t)\} \\ &= D_r(\sigma(j)) \leq \max_{\rho_r(i)>0} D_r(\sigma(i)). \end{aligned}$$

□

Define m_{ρ_r} as the maximum of 0, $\max_{\rho_r(i)>0} D_r(\sigma(i))$, and $\max_{\rho_r(i)<1} -D_r(\sigma(i))$. This measures how far $X(r,t)$ is from the two most extreme ways of thresholding.

Theorem 4.3.12. *With the notation of Theorem 4.3.1,*

$$\mathrm{d_B}(\mathrm{Dgm}(d_X^{\mp}), \mathrm{Dgm}(D_r)) \leq m_{\rho_r} + 2\sqrt{d}r$$

Proof. Combining Lemma 4.3.6, Lemma 4.3.11, and items 3 and 4 of Lemma 4.3.7. □

4.3.3 Examples

We describe an example that illustrates many of the bounds derived above. The ground-truth object X consists of an array of small circular spots lying inside a large disk together with the outside of this disk; see the image at top left of Fig. 4.6. In total this image is 2048×2048 pixels and the large white disk is a circle of radius $R_2 = 510$ pixels. The small black disks have radius $R_1 = 5$ pixels and centers $w = 85$ pixels apart.

The reach of ∂X is the radius R_1 of the small spots, so the image with 2048^2 pixels has $1 = r < \text{reach}(\partial X)/\sqrt{2} = 3.54$. For any resolution with $r < 3.54$, (no. pixels $> 580^2$) Remark 4.3.10 guarantees that the bottleneck distance between the persistence diagrams for $d_X^{\mp}$ and $\mathrm{D}[X(r,t)]$ is no larger than $2\sqrt{2}r$.

At coarser resolutions, Theorem 4.3.1 still applies with $l_X(\sqrt{d}r) = R_2 - \frac{w}{\sqrt{2}} + R_1 + \sqrt{2}r = 454.9 + \sqrt{2}r$ when $\text{reach}(\partial X) \leq \sqrt{2}r \leq \frac{w}{\sqrt{2}} - R_1$. So we see that

$$\mathrm{d_B}(\mathrm{Dgm}(d_X^{\mp}), \mathrm{Dgm}(\mathrm{D}[X(r,t)])) \leq 454.9 + 3\sqrt{2}r,$$

for $3.55 \leq r \leq 39.0$.

For this synthetic example, we can also compute ρ_r for all choices of r. Let $\epsilon > 0$ be a small tolerance to accommodate noise. The pixels where $\epsilon < \rho_r < 1 - \epsilon$ are those that have non-empty intersection with ∂X. At high resolutions, i.e., pixel sizes $r < \text{reach}(\partial X)/\sqrt{2}$, this means the Hausdorff distances between $X(r,t)$ and $X(r, 1-\epsilon)$ or $X(r,\epsilon)$ are small.

At coarser resolutions, for example the image with 64×64 pixels depicted at lower left of Fig. 4.6, each spot is smaller than a pixel and all pixels intersecting the large disk have $\epsilon < \rho_r < 1 - \epsilon$. In this situation, setting $t = 1 - \epsilon$ means the set $X(r, 1-\epsilon)$ is contained in the outside of the large disk, while $X(r,\epsilon)$ is the entire square.

Fig. 4.6 shows that the red bound from Section 4.3.2 has the advantage of being tighter than the green bound from Section 4.3.1 at lower resolutions (i.e., pixel sizes larger than the reach). However, at high resolutions the bound given by Remark 4.3.10 is better.

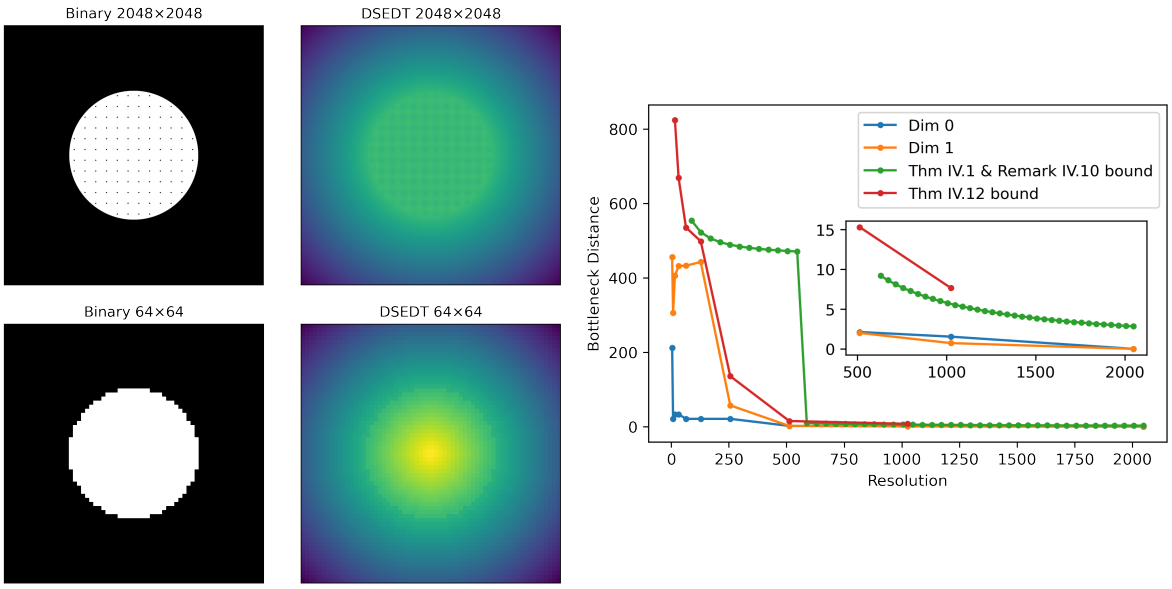

Figure 4.6: Left two columns: The top row shows a high resolution binary image with corresponding DSEDT and the bottom row shows a lower resolution version obtained by downsampling with the averaging method and thresholding with $t = 0.5$ and its corresponding DSEDT. Right column: Bottleneck distances comparing persistence diagrams for each resolution with that for the highest resolution. The distances for dimensions 0 and 1 are shown along with the bounds provided by Theorem 4.3.1 with Remark 4.3.10 (green) and Theorem 4.3.12 (red). © 2021 IEEE.

4.4 Applications

In any real-world situation, the reach and the leash for the object being imaged are likely to be unknown, and/or computationally expensive to estimate. Similarly, the density ρ_r at a given pixel length r might be known, but noisy. So in this section, we explore a different approach. We treat the high-resolution image of the object as a ground truth, and describe how much information is lost in the persistence diagrams of DSEDTs computed for a succession of larger voxel sizes, i.e., decreasing image dimensions.

In our case studies, we begin with the highest resolution image and downsample to lower resolution images as follows. For a d-dimensional binary image with grid spacing r, and dimensions $n_1 \times \cdots \times n_d$, we compute the averages in blocks of a^d voxels, where a is called the kernel size. Based on these average values, we create the downsampled binary image with grid spacing $r' = ar$ and dimensions $n_1/a \times \cdots \times n_d/a$ by thresholding at value $t = 0.5$, similar to the process described in Section 4.1.2. From the new image, we compute the DSEDT, with the new grid spacing r' used as a scaling factor.

All examples in this section are square, so the image size is n^d and we use n to quantify resolution. The voxel size is specified with arbitrary units so that $r = 1$ for the highest-resolution image in each case. To avoid issues caused by boundary effects, we only choose kernel sizes that fit evenly within the original high-resolution image dimensions. Thus kernel sizes are always integer divisors of the image size. This gives us a collection of images of different resolutions that are approximations of the highest resolution image. Then we compute the bottleneck distance between the persistence diagram for each resolution with the persistence diagram from the high resolution image. This allows us to quantify how much information is lost when we downsample.

We compute the persistence diagrams and bottleneck distance, using the Giotto-TDA python package[TLT+20]. Code for this project, including code to generate the synthetic examples, can be found on GitHub[Tym].

4.4.1 Structure at Different Length Scales

Our goal in this section is to explore how the persistent homology of a DSEDT changes at different image resolutions when approximating a particular object $X \subset \mathbb{R}^d$, with structure at different length scales, illustrated in Fig. 4.7. Although the overall trend is that bottleneck distance decreases with resolution, this decrease is not monotonic. The distinct length scales mean the distances show a succession of plateaus as each feature is resolved and the persistence diagrams remain relatively stable over an interval of resolutions.

We now define a Δ-ε plateau to capture the change in bottleneck distance between persistence diagrams as the image resolution changes. Recall that the (linear) image size n is used to quantify resolution, and, for the remainder of this section, denote the DSEDT $\mathrm{D}[X(r,t)]$ by D^n. Let N denote the highest resolution available and take an interval $\Delta = [\ell, m]$ of image resolutions and $\varepsilon > 0$. We then say there is a Δ-ε *plateau* if for all $j, k \in [\ell, m]$,

$$|\mathrm{d_B}\left(\mathrm{Dgm}(D^j), \mathrm{Dgm}(D^N)\right) - \mathrm{d_B}\left(\mathrm{Dgm}(D^k), \mathrm{Dgm}(D^N)\right)| < \varepsilon.$$

By the triangle inequality $\mathrm{d_B}\left(\mathrm{Dgm}(D^j), \mathrm{Dgm}(D^k)\right) < \varepsilon$ is a sufficient condition for a Δ-ε plateau.

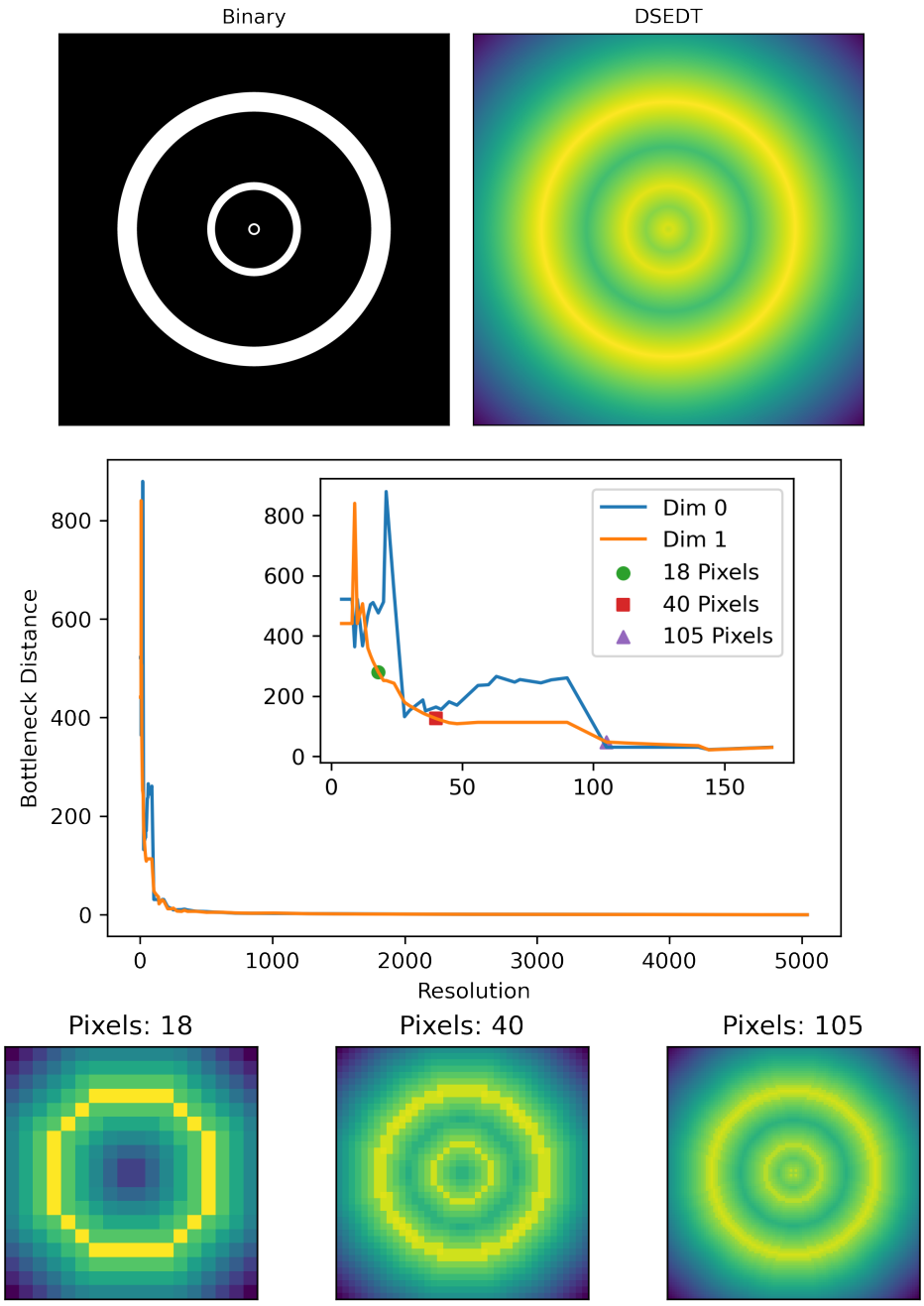

Figure 4.7: Top, the digital image we use as the ground truth with the corresponding DSEDT. Middle, the plot of the bottleneck distances between 1-dimensional persistence diagrams from each resolution and the highest resolution. Below, the DSEDT of the downsampled image at three different resolutions. © 2021 IEEE.

As an example, consider the image in Fig. 4.7 with three nested rings of different thicknesses. The original high-resolution image, X_N, has $r = 1$ and $N^2 = 5040^2$ pixels. We downsample the original image to create 57 different images, where each new image has a kernel of size a where a is a proper divisor of 5040, so that the resolution in each case is $n = 5040/a$, and the voxel size at this resolution is $r = a$. The DSEDT and persistence diagrams are computed for each image, and then we find the bottleneck distance $\mathrm{d_B}(\mathrm{Dgm}(D^n), \mathrm{Dgm}(D^{5040}))$ comparing each lower-resolution image with the original one. The plot of these results in Fig. 4.7 shows three specific behaviors we would like to emphasize; spikes, plateaus, and final plateaus.

As we increase the resolution, some fluctuations in the bottleneck distance are to be expected, even within a plateau. These fluctuations are on the order of magnitude of the pixel diameter. For the example in Fig. 4.7 it makes a difference whether the center of the image is the center of a pixel (for n odd) or the center is in the closure of 4 pixels (for n even). These effects can cause the bottleneck distance to spike, i.e. to increase and then immediately decrease. The lower the resolution, the more pronounced these spikes usually appear. The blue zero-dimensional bottleneck distance curve in Fig. 4.7 has a spike at $n = 21$, the only odd number in this range of resolutions, with an increase by 366 from $n = 20$ to $n = 21$ and a decrease by 333 from $n = 21$ to $n = 24$. The magnitude of this spike should be compared with the pixel diameter for $n = 21$, which is $r\sqrt{2} = 240\sqrt{2} = 339.4$.

Next, we observe the plateau behavior outlined in the definition of a Δ-ε plateau. This is caused by the introduction of a new topological feature, such as a grain of sand or a small pore. In the nested ring example, the plot of one-dimensional bottleneck distances shows three plateaus. The zero-dimensional bottleneck distances mimics this behavior, but the first one or two plateaus are overshadowed by noise, like spikes, explained above. The three plateaus are due to the three different ring widths. After each of these is resolved, an increase in resolution does little to change the persistence diagram. As shown in Fig. 4.7, we see just one ring at resolution 18, two rings at $n = 40$, the third ring starts being resolved at $n = 105$ (where it only consist of 4 pixels) and gets fully resolved at $n = 210$. Specifically, the first Δ-ε plateau is at $\Delta = [18, 24]$ with $\varepsilon = 36.96$ in dimension 1 and $\varepsilon = 402.51$ in dimension 0. The second Δ-ε plateau is at $\Delta = [40, 90]$ with $\varepsilon = 17.31$ in dimension 1 and $\varepsilon = 104.49$ in dimension 0. The third Δ-ε plateau is at $\Delta = [210, 5040]$ with $\varepsilon = 13.43$ in dimension 1 and $\varepsilon = 15.80$ in dimension 0.

The existence of structure at just a few distinct, well separated length scales in the synthetic example means there is a sequence of plateaus as each structure is resolved. This does not happen for the porous materials discussed in Section 4.4.2. However, we note that Corollary 4.3.9, and Remark 4.3.10 guarantee the existence of a final plateau for any object with positive reach. Specifically, let M be the lowest resolution required for the corresponding voxel size $r_M < \frac{1}{\sqrt{d}}\mathrm{reach}(\partial X)$, and assume that the highest resolution image has $N > M$. Then the bottleneck distances will have a Δ-ε plateau with $\Delta = [M, N]$ and $\varepsilon = 4\sqrt{d}\, r_M \leq 4\mathrm{reach}(\partial X)$. In practice, we often see a final plateau even when the reach is zero, such as when the leash bounding the bottleneck distance converges to zero. The glass bead packing in the following section is such an example.

Although a final plateau suggests the image resolution is sufficient to capture the actual underlying structure of the imaged object, we can never definitively know whether a plateau is final or not, because we do not know what structure exists at length-scales finer than the voxel size. Hence, the plateau serves only as a guide for resolution choice.

4.4.2 Application to Porous Materials

It is common for materials science examples to have zero reach. For example, packings of spherical or elliptical grains have $\mathrm{reach}(X) = 0$, because two grains touch at a single point, while other porous materials, such as a metal foam, have sharp corners. Here we present three examples of micro-CT images of porous materials segmented into the two phases of solid and void [SP15]. The images are subregions from a packing of spherical glass beads, a sandstone (from Castlegate) and an unconsolidated sandpack, each with 512^3 voxels.

In Fig. 4.8 we depict slices through each 3D binary image and their signed Euclidean distance transforms, followed by plots of the bottleneck distances between persistence diagrams computed at different resolutions. In each case, we downsample the binary image to a lower resolution using the averaging technique described at the start of Section 4.4 with a threshold $t = 0.5$. We only use kernel sizes that fit evenly within the image; with $N = 512$ we must use powers of two, $a = 2^k$. As the persistence diagrams for these examples have so many points, we use an approximation algorithm for the bottleneck distances, as implemented in Giotto-TDA. Specifically, we use an approximation value of $\delta = 0.1$ for the glass bead packing, and $\delta = 0.5$ for the other two to make these computations feasible.

The green curve shown on the plots of bottleneck distances in Fig. 4.8 is the function $2\sqrt{3}r = 2\sqrt{3}(512/n)$, where $r = 1$ is the voxel spacing for the highest-resolution image, and n is the resolution as measured by the number of voxels along each side of the cube. This is the bound on bottleneck distance we derived in Section 4.3 for the case that the voxel size $r < \mathrm{reach}(\partial X)/\sqrt{d}$. As already argued, the reach is zero for the glass bead pack, and likely to be zero for the sand pack, so the fact that this bound holds suggests that our estimation results are too generous, and/or that $l_X(s)$ is approximately s despite the reach being zero for these examples.

We note that the glass bead example has significantly larger bottleneck distances between its 1-dimensional persistence diagrams compared to the 0-dimensional distances for the resolutions $n = 32, 64, 128, 256$. Inspection of the dimension one $\mathrm{Dgm}^1(D^n)$ diagrams and the original image shows that the larger values of $d_B(\mathrm{Dgm}^1(D^n), \mathrm{Dgm}^1(D^{512}))$ are due to the presence of just a couple of high-persistence cycles near the boundary of the image, each cycle due to a bead that intersects two faces of the boundary. The sensitivity of the bottleneck distance to outliers is well known and is the reason Wasserstein distances between diagrams are often preferred.

An important physical parameter associated with porous materials is the percolation threshold, l_c. This is the radius of the largest sphere that can pass through the pore space from one side of the image to the opposite. The distribution of points in $\mathrm{Dgm}^0(D^n)$ shows a clear signature of this critical length scale; see [RSDS16] for details. As can be seen in the persistence diagrams provided in Appendix D in the extended version of this chapter on arXiv [HTS+21a], this signature yields the same estimate of l_c for image resolutions $n = 512, 256, 128$ in the three example materials. This supports the designation of a Δ-ε plateau in the 0-dimensional bottleneck distances with $\Delta = [64, 512]$, and $\varepsilon = 10$ for all three samples.

4.5 Discussion

This chapter has presented two sets of results about the digital approximation of functions, and of solid objects. Section 4.2 makes explicit how far points in a persistence diagram can

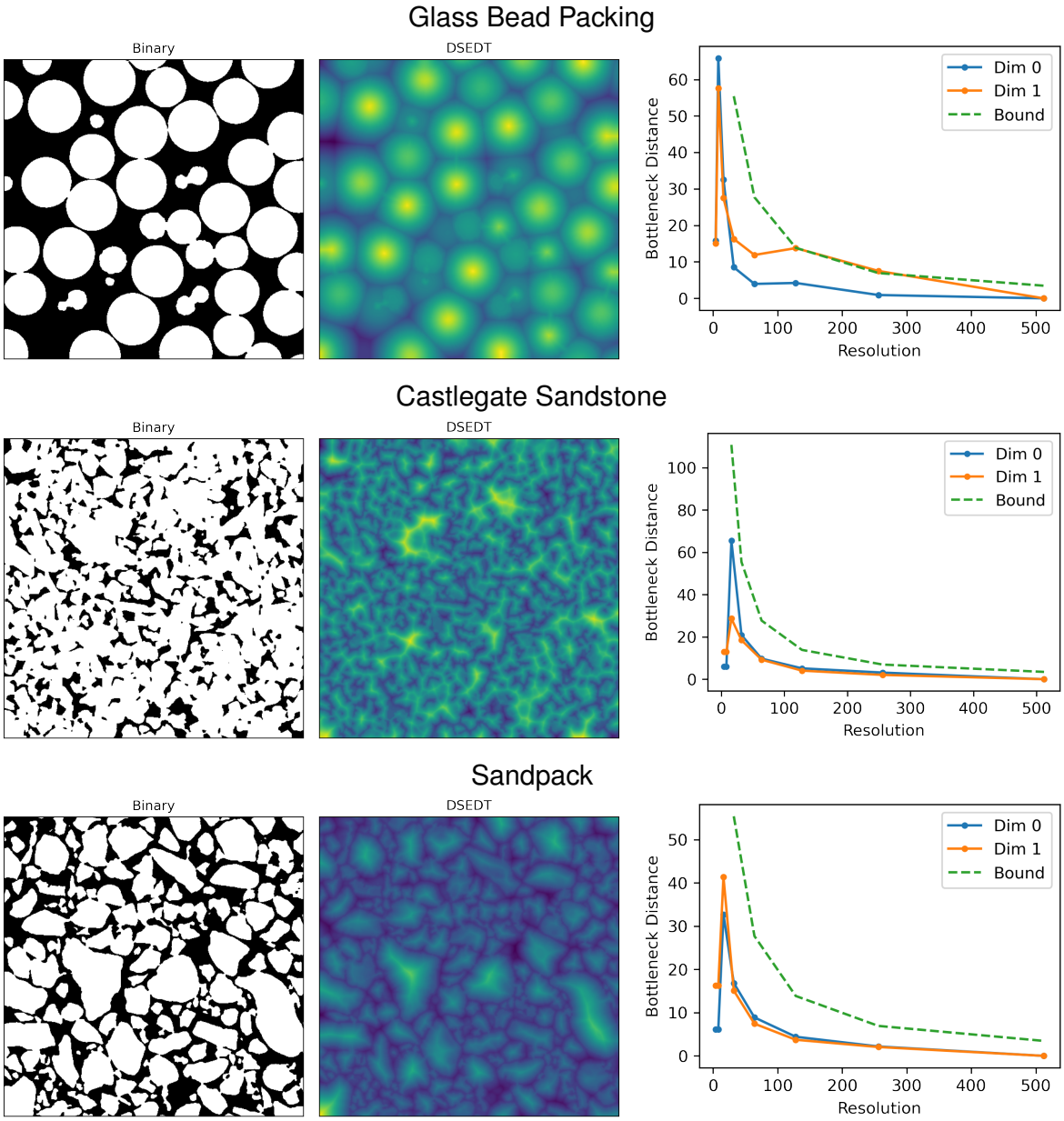

Figure 4.8: 2D slices of the binary and DSEDT images for the glass bead packing, Castlegate sandstone, and sand packing samples. The colormap for the DSEDT is scaled to the max and min values in each case. The plots in the right column show the bottleneck distance between persistence diagrams for each downsampled resolution and the highest resolution. The green dashed curve is the function $2\sqrt{3}(512/n)$, where n is the image resolution. © 2021 IEEE.

move when working with locally averaged digital approximations to a function at different resolutions.

Section 4.3 has considered a more subtle question of how close are the persistence diagrams of a solid object X and a digital approximation to it, $X(r,t)$, when they are filtered by the continuous and discrete signed Euclidean distance transforms respectively. These results are the analogues of seminal work for point-cloud approximations of manifolds, which do not translate easily to the digital image setting.

In general, the difference between the continuous and discrete distance transforms is given in terms of the voxel diameter plus the leash (Theorem 4.3.1). The leash may be large, for example, if X is a material that has regions of micro-porosity or extended structures with geometric detail below the voxel size r. When the voxel diameter $\sqrt{d}r < \mathrm{reach}(\partial X)$, we show that the leash, $l_X(\sqrt{d}r) = \sqrt{d}r$, so that the bottleneck distance between persistence diagrams is bounded by $2\sqrt{d}r$ (Corollary 4.3.9 and Remark 4.3.10).

The practical consequences of these results are that we expect the persistent homology to converge with increasing image resolution, but the error may not be monotonic, especially when considering images at low resolutions. When there is no prior information about the critical length scales of the object X, the x-ray density function given by the CT-scan can be interpreted as a (noisy) approximation to the density function ρ_r, and the Hausdorff distance between the two threshold choices $X(r, 1-\epsilon)$ and $X(r,\epsilon)$ provides an estimate of the possible error in the persistence diagrams (Theorem 4.3.12).

Ultimately, our results provide guidance to practitioners on how to balance the time, cost, and processing power required for image acquisition and persistent homology computations against the desired level of accuracy in their results.

CHAPTER 5

The Density Fingerprint of a Periodic Point Set

This chapter is based on [EHK+21].

Modeling a crystal as a periodic point set, we present a fingerprint consisting of density functions that facilitates the efficient search for new materials and material properties. We prove invariance under isometries, continuity, and completeness in the generic case, which are necessary features for the reliable comparison of crystals. The proof of continuity integrates methods from discrete geometry and lattice theory, while the proof of generic completeness combines techniques from geometry with analysis. The fingerprint has a fast algorithm based on Brillouin zones and related inclusion-exclusion formulae. We have implemented the algorithm and describe its application to crystal structure prediction.

Prior work. The prior work in this area is best summarized by listing the software systems currently used in practice: COMPACK [CM05], MERCURY [MEM+06], and COMPSTRU [FOT+16]. These systems are of great help in comparing crystals, but they employ heuristics like cut-offs and tolerances, which come with the usual drawbacks. It is our ambition to develop the mathematical and computational foundations needed to overcome the current deficiencies.

Outline. Section 5.1 introduces the density functions for a periodic point set and the corresponding density fingerprint map. Section 5.2 proves that the density fingerprint map is continuous with respect to perturbations of the periodic point set. Section 5.3 proves that the density fingerprint map is complete for generic periodic point sets. Section 5.4 explains how the density fingerprint is computed using the Brillouin zones of the points. Section 5.5 describes a preliminary application of the density fingerprint map to Crystal Structure Prediction. Section 5.6 concludes the chapter.

5.1 The Density Fingerprint and its Invariance

In this entire chapter, we assume all lattices to be full lattices, i.e. of dimension three.

A continuous invariant for comparing crystals is the density, defined as the total volume of balls centered at points in the motif divided by the volume of the unit cell. To avoid the choice of radii, we grow the balls continuously and simultaneously from their centers and

get a 1-dimensional function rather than a single number. There are still many periodic point sets this function cannot distinguish, for example any hexagonal close packing from the face-centered cubic lattice. We therefore add information by distinguishing points covered by a different number of balls.

Definition 5.1.1 (Density Functions and Fingerprint)**.** Let $A = M + \Lambda \subset \mathbb{R}^3$ be a periodic point set and write $A(t)$ for the collection of closed balls, $B(a;t)$, of radius $t \geq 0$ centered at the points $a \in A$. The *k-fold cover* of $A(t)$, denoted $\bigcup^k A(t)$, consists of all points $x \in \mathbb{R}^3$ contained in k or more of these balls. The fractional volume of the k-fold cover, $\varphi_k^A(t) = \operatorname{vol}\left(\operatorname{Unit} U \cap \bigcup^k A(t)\right)/\operatorname{vol}(\operatorname{Unit} U)$, is also the probability that a point chosen uniformly at random within a unit cell, $\operatorname{Unit} U$, belongs to at least k balls, and subtracting the fractional volume of the $(k+1)$-fold cover, we get the probability that the random point belongs to exactly k balls:

$$\varphi_k^A(t) = \operatorname{Prob}[x \in B(a;t) \text{ for } k \text{ or more points } a \in A]; \tag{5.1}$$

$$\psi_k^A(t) = \varphi_k^A(t) - \varphi_{k+1}^A(t) = \operatorname{Prob}[x \in B(a;t) \text{ for exactly } k \text{ points } a \in A]. \tag{5.2}$$

We call $\psi_k^A \colon [0,\infty) \to [0,1]$ the *k-th density function* of A. The *density fingerprint* of A is the vector of density functions: $\Psi(A) = (\psi_0^A, \psi_1^A, \ldots, \psi_k^A, \ldots)$, and $A \mapsto \Psi(A)$ is the *density fingerprint map*.

See Figure 5.1, which illustrates the density functions for the hexagonal and the square lattices in $\mathbb{R}^2$. Note that the density fingerprint is an isometry invariant and that it neither depends on the lattice used to write A as a periodic point set, nor on its basis.

Lemma 5.1.2 (Invariance under Isometries)**.** *Let $A \subseteq \mathbb{R}^3$ be a periodic point set, and let $Q \subseteq \mathbb{R}^3$ be isometric to A. Then $\Psi(A) = \Psi(Q)$.*

Proof. Let now $\operatorname{iso} \colon \mathbb{R}^3 \to \mathbb{R}^3$ be the isometry for which $Q = \operatorname{iso}(A)$, and note that it also maps $A(t)$ to $Q(t)$ and $\bigcup^k A(t)$ to $\bigcup^k Q(t)$ for every $k \geq 0$. It follows that $\psi_k^A(t) = \psi_k^Q(t)$, for every $k \geq 0$, and therefore $\Psi(A) = \Psi(Q)$, as claimed. □

While the fingerprint map is not invariant under similarities, we can write $\Psi(sA) = (\psi_0^A \circ s, \psi_1^A \circ s, \ldots, \psi_k^A \circ s, \ldots)$, in which $s(t) = st$ scales the radius. It would therefore be easy to construct a fingerprint map that is invariant under similarities, namely by normalizing the radius, e.g. by letting the radius be tr, in which r is the packing radius of A.

5.2 Continuity

We prove that the density fingerprint map is Lipschitz continuous with respect to small perturbations of the points. To formalize this result, we introduce distances between periodic point sets and between density fingerprints. For sets $A, Q \subseteq \mathbb{R}^3$ of equal cardinality, the *(Euclidean) bottleneck distance* is the infimum, over all bijections, $\gamma \colon A \to Q$, of the supremum Euclidean distance between matched points, and for density fingerprints, $\Psi(A), \Psi(Q)$, we use the supremum of the weighted infinity norms of the differences between corresponding density functions:

$$d_B(A,Q) = \inf_{\gamma \colon A \to Q} \sup_{a \in A} \|a - \gamma(a)\|_2, \tag{5.3}$$

$$d_\infty(\Psi(A), \Psi(Q)) = \sup_{k \geq 0} \tfrac{1}{\sqrt[3]{k+1}^2} \|\psi_k^A - \psi_k^Q\|_\infty. \tag{5.4}$$

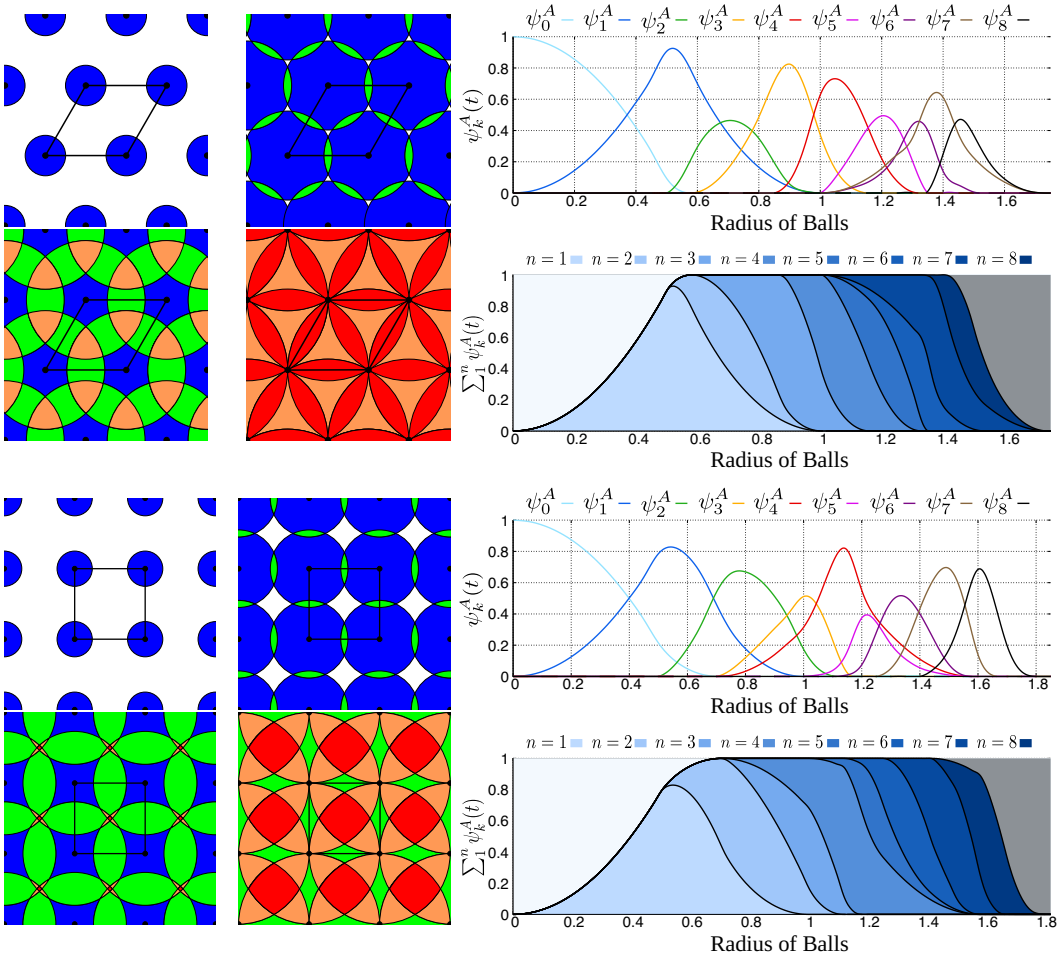

Figure 5.1: The density fingerprint map of the hexagonal lattice on the *top* and, for comparison, of the square lattice on the *bottom*. *Left*: the k-fold covers of the two sets for four different radii each: $t = 0.25, 0.55, 0.75, 1.00$. *Right*: the graphs of the respective first nine density functions above the corresponding *densigram*, in which the zeroth function can be seen upside-down and the remaining density functions are accumulated from left to right.

Note the damping of the difference between corresponding density functions. The reason for it is technical and related to the fact that density functions with higher k tend to vanish at later values of t. As a consequence, the sensitivity of the density function to any perturbation increases with growing k, and the damping compensates for this tendency. Before proving Lipschitz continuity, we show that two periodic point sets with small bottleneck distance between them necessarily have a common lattice.

Lemma 5.2.1 (Common Lattice)**.** *Let A, Q be periodic point sets in $\mathbb{R}^3$, and let $r_Q > 0$ be the packing radius of Q. If $d_B(A, Q) < r_Q$, then there is a lattice Λ with unit cell $\operatorname{Unit} U$ in $\mathbb{R}^3$ such that $\#(A \cap \operatorname{Unit} U) = \#(Q \cap \operatorname{Unit} U)$ and $A = (A \cap \operatorname{Unit} U) + \Lambda$ and $Q = (Q \cap \operatorname{Unit} U) + \Lambda$.*

Proof. Since $A, Q \subseteq \mathbb{R}^3$ are periodic, there are lattices with unit cells such that $A = (A \cap \operatorname{Unit} U_A) + \Lambda_A$ and $Q = (Q \cap \operatorname{Unit} U_Q) + \Lambda_Q$. To get a contradiction, we assume that there is however no common lattice for A and Q. Equivalently, $\Lambda_A \cap \Lambda_Q$ is a lattice of dimension at most 2. Therefore there exists a basis vector, v, of Λ_A such that $nv \in \Lambda_Q$ implies $n = 0$. Picking a point $a \in A$, we consider the infinitely many points $a(n) = a + nv$,

with $n \in \mathbb{Z}$. For each $a(n)$, let $q(n)$ be the point in Λ_Q such that $a(n) \in q(n) + \operatorname{Unit} U_Q$, and define $b(n) = a(n) - q(n)$, which we note belongs to $\operatorname{Unit} U_Q$.

There are infinitely many pairwise different points $b(n)$ in the unit cell, and it suffices to prove that at least one is at distance larger than $\delta = d_B(A, Q)$ from all points in Q. To see this, let $b(i)$ and $b(j)$ be at distance less than $\varepsilon = r_Q - \delta$ from each other, and note that $b(i + n[j - i]) = b(i) + n[b(j) - b(i)]$, for $n \in \mathbb{Z}$, provided the point on the right-hand side of the equation belongs to $\operatorname{Unit} U_Q$. In other words, we have an entire line of points with distance less than ε between contiguous points. The gap between balls of radius δ centered at the points in Q is at least 2ε, which implies that at least one of the points on the line is outside all such balls. This contradicts the assumption that the bottleneck distance between A and Q is $\delta = r_Q - \varepsilon$. The existence of a common lattice of A and Q follows. □

The proof of Lipschitz continuity makes use of the common lattice of the sets before and after the perturbation. We therefore formulate the claim assuming that the bottleneck distance between the two sets is less than the packing radii.

Theorem 5.2.2 (Fingerprint Continuity)**.** *Let A, Q be periodic point sets in $\mathbb{R}^3$, both with packing radius at least $r > 0$ and with covering radius at most $R < \infty$. If $\delta = d_B(A, Q) < r$, then there exists a constant $C = C(r, R)$ such that $d_\infty(\Psi(A), \Psi(Q)) \leq C \cdot d_B(A, Q)$.*

Proof. By Lemma 5.2.1, there is a lattice, $\Lambda \subseteq \mathbb{R}^3$, that is common to both sets, A and Q, and we write $\operatorname{Unit} U$ for the corresponding unit cell. Let $\gamma\colon A \to Q$ be a bijection such that $d_B(A, Q)$ is the supremum Euclidean distance between corresponding points, let k be a non-negative integer, and let t be a positive real number. We need an upper bound for

$$\left|\psi_k^A(t) - \psi_k^Q(t)\right| = \frac{\left|\operatorname{vol}\left(A_t^k \cap \operatorname{Unit} U\right) - \operatorname{vol}\left(Q_t^k \cap \operatorname{Unit} U\right)\right|}{\operatorname{vol}\left(\operatorname{Unit} U\right)}, \tag{5.5}$$

in which $A_t^k = \bigcup^k A(t) \setminus \bigcup^{k+1} A(t)$ consists of all points $x \in \mathbb{R}^3$ contained in exactly k balls of $A(t)$, and similarly for Q_t^k. As a first step, we find an upper bound on the numerator, Δ, for the case in which γ is the identity except for one point, $a \in M$, which it maps to $q = \gamma(a) \in B(a; \delta)$; that is: $Q = A \setminus (a + \Lambda) \cup (q + \Lambda)$. A point $x \in \mathbb{R}^3$ is possibly covered by a different number of balls before and after the perturbation only if $x \in [B(a; t) \ominus B(q; t)] + \Lambda$, with $\ominus$ denoting the symmetric difference. Observe that this set is contained in $[B(\frac{a+q}{2}; t + \frac{\delta}{2}) \setminus B(\frac{a+q}{2}; t - \frac{\delta}{2})] + \Lambda$. Hence,

$$\Delta \leq \operatorname{vol}\left(B(a; t) \ominus B(q; t)\right) \leq \frac{4\pi}{3}\left[\left(t + \tfrac{\delta}{2}\right)^3 - \left(t - \tfrac{\delta}{2}\right)^3\right] = \frac{4\pi}{3}\left[3\delta t^2 + \tfrac{1}{4}\delta^3\right]. \tag{5.6}$$

Perturbing one point of M after the other, we can bound the error by (5.6) each time. Using the intensity, $\rho = \#M / \operatorname{vol}\left(\operatorname{Unit} U\right)$, this implies

$$\left|\psi_k^A(t) - \psi_k^Q(t)\right| \leq \rho\Delta \leq \rho\frac{4\pi}{3}\left[3\delta t^2 + \tfrac{1}{4}\delta^3\right]. \tag{5.7}$$

We can eliminate the dependence on t by observing that for each k there is a value of t beyond which the k-th density functions of A and Q vanish. To determine this value, consider a point $y \in \mathbb{R}^3$ and the sets $A \cap B(y; t)$ and $Q \cap B(y; t)$. By the definition of R, the balls of radius R centered at the points of A cover $B(y; t - R)$, and similarly for Q. It follows that the two sets contain at least $(t/R - 1)^3$ points each. Setting $k + 1 \leq (t/R - 1)^3$, we

see that for $t \geq R\sqrt[3]{k+1} + R$, both sets have at least $k+1$ points each. Equivalently, y is covered by at least $k+1$ balls of radius t. Since this holds for every point $y \in \mathbb{R}^3$, we have $\psi_k^A(t) = \psi_k^Q(t) = 0$ for all $t \geq R\sqrt[3]{k+1} + R$. Note that $R\sqrt[3]{k+1} + R \leq 2R\sqrt[3]{k+1}$ for all $k \geq 0$. Replacing t in (5.7) by the latter bound, we get

$$\frac{1}{\sqrt[3]{k+1}^2} \left\| \psi_k^A - \psi_k^Q \right\|_\infty \leq 16\pi\rho R^2\delta + \frac{\pi}{3}\rho\delta^3 \leq \frac{12R^2}{r^3}\delta + \frac{1}{4r^3}\delta^3, \tag{5.8}$$

in which we use $\rho\frac{4\pi}{3}r^3 \leq 1$ to get the final inequality. Using $\delta^2 < r^2 < R^2$, this gives $C = 13R^2/r^3$ as an upper bound for the Lipschitz constant. □

Figure 5.2 illustrates Theorem 5.2.2 for a periodic point set, A, and its perturbation, Q, in $\mathbb{R}^2$ by showing the first eight (undamped) density functions for both sets in different colors.

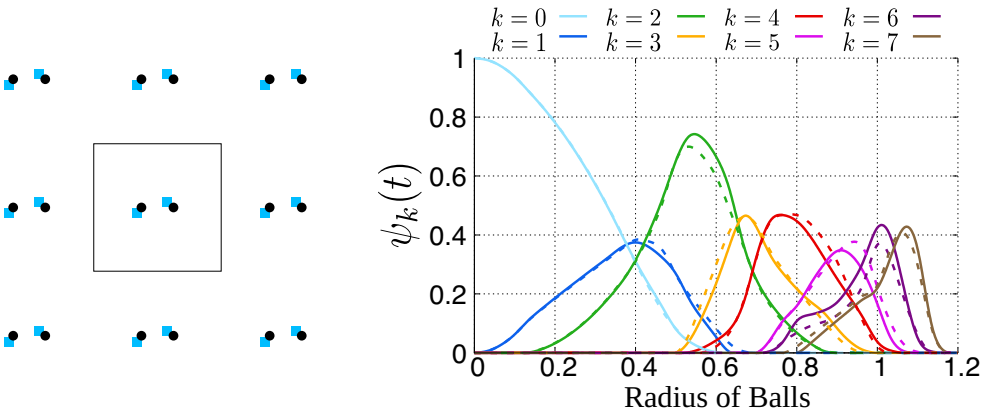

Figure 5.2: *Left*: a periodic point set with two *black* points in its square unit cell, and the perturbed periodic point set with two *blue* points in the same unit cell. *Right*: the graphs of the density functions are *solid* for the original set and *dashed* for the perturbed set. As predicted by Theorem 5.2.2, the small perturbation of the periodic point set causes a small change in the fingerprint.

5.3 Completeness

The fingerprint map is *complete* if it is injective up to isometries; that is: non-isometric periodic point sets are mapped to different fingerprints. We prove completeness generically, i.e. on a dense open subset; compare to [BK04, Sen08, LSS03]. The density fingerprint also distinguishes non-generic sets for which other means fail, as will be illustrated by an example in Section 5.3.2. The completeness of the fingerprint for all periodic point sets, however, remains an open question. Indeed, at the time of writing this thesis, the author is not aware of a 3-dimensional counterexample to completeness, but there is a 1-dimensional counterexample due to Morteza Saghafian: letting $V = \{0, 4, 9\}$ and $W = \{0, 1, 3\}$, it can be checked that the finite sets $V + W$ and $V - W$, and the periodic point sets $15\mathbb{Z} + (V + W)$ and $15\mathbb{Z} + (V - W)$ cannot be distinguished by the 1-dimensional density fingerprint map.

5.3.1 Generic Completeness

We prove the completeness of the density fingerprint map for generic periodic point sets in $\mathbb{R}^3$. The notion of genericity is defined by conditions that are satisfied by an open and

dense subset of the space of periodic point sets. We formulate such conditions in terms of the *circumradius* of edges, triangles, and tetrahedra, which is the radius of the smallest sphere that passes through the vertices of the simplex. To avoid infinitely many constraints, we introduce an upper bound on the circumradii to consider. Denoting by $L = L(A, \vartheta)$ the list of all edges (pairs), triangles (triples) and tetrahedra (quadruples) spanned by points of a periodic point set A whose circumradius is at most ϑ, we call A *generic for a constant threshold,* ϑ, if—apart from necessary violations due to periodicity—it satisfies the following three conditions:

I. the circumradii of different simplices in L are different;

II. the circumradii of different edges in L are not related to each other by a factor of 2.

III. for every $t \leq \vartheta$, there is at most one set of six circumradii of simplices in L such that the edges with twice their lengths assemble to a tetrahedron whose circumradius is t.

We call an edge a *lattice edge* if its length is the distance between two points in the lattice. Lattice edges violate Condition II and can thus be identified as such. A *lattice triangle* has three lattice edges, and a *lattice tetrahedron* has six lattice edges. The important difference between lattice and non-lattice simplices is that only the latter are unique up to lattice translations.

Since Conditions I, II, III can be phrased via finitely many algebraic equations in the vectors $x \in M, v_1, v_2, v_3$, the set of generic periodic point sets with threshold ϑ is open and dense (with respect to perturbations of these vectors) in the space of all periodic point sets with at most m motif points. We write $\mathrm{Rad}(A)$ for the largest finite circumradius of $p \leq 4$ points in A with pairwise distance at most four times the diameter of the unit cell. Since the diameter is the distance between two lattice points, this implies that $\mathrm{Rad}(A)$ is at least double the diameter.

Theorem 5.3.1 (Generic Completeness)**.** *Let $A, Q \subseteq \mathbb{R}^3$ be non-isometric periodic point sets that are generic for the threshold $\vartheta = \max\{\mathrm{Rad}(A), \mathrm{Rad}(Q)\}$. Then $\Psi(A) \neq \Psi(Q)$.*

Proof. Let $[A]$ denote the isometry class of A. We prove the unique reconstruction of $[A]$ from $\Psi(A)$ in two steps:

$$\Psi(A) \xrightarrow{\text{STEP 1}} \text{tetrahedra in } L(A, \vartheta), \text{ up to isometries} \xrightarrow{\text{STEP 2}} [A]$$

STEP 1: Each density function is a weighted sum of the volumes of intersections of 2, 3, or 4 balls around points of A; see [EIH18, Equation (5)]. The volume formulas of such intersections are given in [EF94]. It is cumbersome but not difficult to prove that they are piecewise analytic, and that the circumradii of edges, triangles and tetrahedra spanned by points of A are the positions where the functions are not analytic. Therefore, the set of all positions up to ϑ where at least one density function is not analytic yields the set of circumradii of simplices of L. We avoid the technicalities of using the differences between the left- and right-derivatives to distinguish which of these are caused by 2, 3, or 4 balls meeting, with the following trick. We treat all circumradii as if they were circumradii of edges, multiply them by two (to get the edge length), and try to assemble six of these edge lengths to form a tetrahedron. Whenever this gives a circumradius of a simplex of L, we have found a tetrahedron of A by Condition III. This way we can uniquely construct all tetrahedra of L up to isometries.

STEP 2: To start the process, we choose a non-lattice tetrahedron from the list. If there is no such tetrahedron, then A is a lattice and can be reconstructed from the lexicographically shortest lattice tetrahedron from the list—i.e. the tetrahedron consisting of the shortest lattice edge, the second-shortest lattice edge (linearly independent from the first), and so on—defining a (Minkowski-)reduced [NS04] and therefore primitive unit cell of A. On the other hand, if there exist non-lattice tetrahedra, we choose the lexicographically shortest one, $abcd$, with non-lattice edge ab.

Placing $abcd$ in space—as we are only interested in the isometry class of A, we can place it arbitrarily—we identify all tetrahedra $abce$ from the list that have abc as a face and try to glue them onto $abcd$. There are two possibilities (related by a reflection) of how to glue $abce$; we denote the two different tip positions by e_1 and e_2. We prove that at most one of the two options gives a positive result when checking if the tetrahedron $abde_i$ is in the list of tetrahedra from Step 1: The triangles abd and abe are non-lattice, and therefore unique in A up to lattice translations by Condition I. Thus, when glued along ab, they span a uniquely defined tetrahedron $abde$ with a certain edge length de that is the distance between d and e_i for at most one of its two possible positions.

This gluing procedure yields (among others) all points of distance at most four times the diameter of the unit cell to a, b, c, d (by definition of $\mathrm{Rad}(A)$), except the ones that lie on a plane spanned by the triangles abc or abd. This neighborhood is large enough such that it contains every motif point at least once and such that it contains a lattice basis, which can be identified by computing the pairwise differences between the reconstructed points and checking whether they satisfy Condition II. Repeating the reconstructed points with respect to the lattice yields the isometry class of A. As the construction was unique given the genericity conditions, we get $\Psi(A) \neq \Psi(Q)$. □

5.3.2 Distinguishing Non-Generic periodic point sets

There are indications that the density fingerprint map distinguishes all periodic point sets and not just the generic ones. We now give the reason for our optimism. Example 5.3.2 describes two periodic point sets that violate the above genericity conditions and can nevertheless be distinguished by the density fingerprint map. On the other hand, the two sets can neither be distinguished by their density nor by their X-ray diffraction patterns; two means commonly used in crystallography to determine the structure of a crystal. X-ray diffraction patterns give all pairwise distance vectors of the periodic point set, but they do not determine the isometry class of a periodic point set [PS30]: there exist *homometric structures*, which are non-isometric periodic point sets with the same 2-point autocorrelation functions; that is: identical multisets of pairs, up to translation. There even exist periodic point sets with the same 2- and 3-point autocorrelation functions, as we now explain.

Example 5.3.2. *Let $A^{(1)}$ and $Q^{(1)}$ be sets with periodicity 32 in $\mathbb{R}$, each with 16 points in the corresponding motif:*

$$0, 7, 8, 9, 12, 15, 17, 18, 19, 20, 21, 22, 26, 27, 29, 30; \tag{5.9}$$

$$0, 1, 8, 9, 10, 12, 13, 15, 18, 19, 20, 21, 22, 23, 27, 30; \tag{5.10}$$

see Figure 5.3. The authors of [GM95, Section 5.3] show that $A^{(1)}$ and $Q^{(1)}$ have the same 2- and 3-point autocorrelation functions. Taking the Cartesian product with $\mathbb{Z}^2$ preserves the equality between the autocorrelation functions, which yields periodic point sets, $A, Q \subseteq \mathbb{R}^3$,

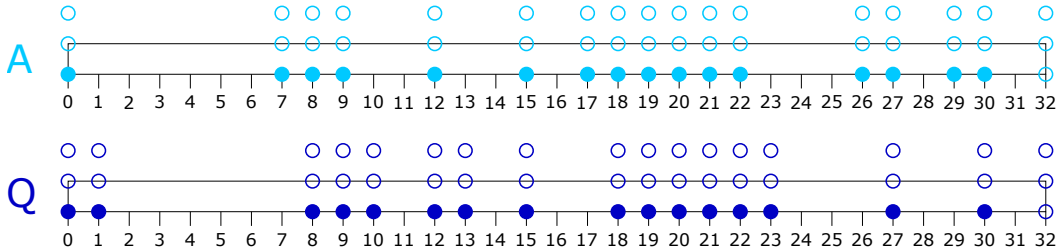

Figure 5.3: periodic point sets A and Q from Example 5.3.2, pictured with rectangular unit cells in two dimensions, for simplicity. Filled dots belong to the motifs while unfilled dots show the periodicity.

with matching 2- and 3-point autocorrelation functions. Nevertheless, our density fingerprint map distinguishes them, as shown in Table 5.1: the L_∞-distances between the first four corresponding density functions vanish but the next five L_∞-distances are strictly positive.

k	0	1	2	3	4	5	6	7	8
$\|\psi_k^A - \psi_k^Q\|_\infty$	0.000	0.000	0.000	0.000	0.005	0.007	0.013	0.022	0.007

Table 5.1: L_∞-distances between the corresponding density functions of the sets A and Q in Example 5.3.2.

5.3.3 How Many Density Functions are Needed?

The density fingerprint consists of infinitely many density functions, of which we can evaluate only a finite number in practice; for example the first K, for some constant K. Assuming completeness, it will be important to have estimates of the smallest K for which the vector of the first K density functions, denoted by Ψ_K, distinguishes two given non-isometric periodic point sets.

We prove $K = \Omega(\omega^3)$, in which $\omega = \mathrm{Diam}(\mathrm{Unit}\, U)/r$ is the ratio of the unit cell diameter over the packing radius. We lack an upper bound in the general case, as this would solve the open completeness question. However, a variant of the proof of Theorem 5.3.1 implies an asymptotically matching upper bound in the generic case (argument omitted). The top half of Figure 5.4 illustrates the construction used to prove the lower bound in one dimension. For every positive integer, ℓ, we let $[0, 2\ell + 4)$ be the unit cell in $\mathbb{R}$, and we let A_ℓ contain all integer points in the unit cell other than 0 and $\ell + 1$. The next missing point along the line is $2\ell + 4$, so there are gaps of size $\ell + 1$ and $\ell + 3$ between them. Similarly, Q_ℓ contains all integer points in the unit cell other than 0 and $\ell + 2$, with two gaps of size $\ell + 2$ each. Hence, $\mathrm{Diam}(\mathrm{Unit}\, U) = 2\ell + 4$, $r = 0.5$, and $\omega = 4\ell + 8$. It is not difficult to show that the first $\ell - 1$ density functions of A_ℓ and Q_ℓ are the same; see Figure 5.4 and the proof of the 3-dimensional case in Lemma 5.3.3. This implies that $K \geq \ell = \frac{\omega}{4} - 2$ is necessary to distinguish A_ℓ and Q_ℓ with Ψ_K. This bound generalizes to $K = \Omega(\omega^3)$ in three dimensions.

Lemma 5.3.3 (Number of Necessary Density Functions)**.** *Unless $K = \Omega(\omega^3)$, there are non-isometric periodic point sets $A, Q \subseteq \mathbb{R}^3$, both with unit cell diameter over packing radius at most ω, such that $\Psi_K(A) = \Psi_K(Q)$.*

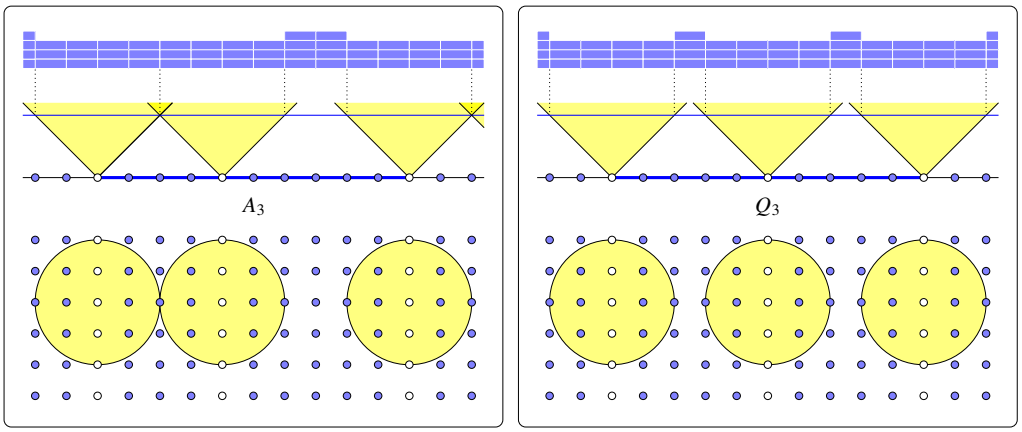

Figure 5.4: *Upper half:* the periodic point sets A_ℓ and Q_ℓ with $\ell = 3$ in one dimension. The blue points are present and the white points are missing. The cones illustrate the range where the absence of the points is felt as the radius increases from bottom to top. The bricks above the cones show how the intervals for radius $t = 2.0$ cover the lines. *Lower half:* the 2-dimensional extensions of the 1-dimensional data sets. The (growing) balls centered at missing points in different slices start touching at radius $t = 2.0$ in A_3 and at radius $t = 2.5$ in Q_3.

Proof. We prove $K = \Omega(\omega^3)$ by generalizing the 1-dimensional example to $\mathbb{R}^3$, taking $\operatorname{Unit} U = [0, 2\ell{+}4){\times}[-0.5, 0.5)^2$ as unit cell, and A_ℓ, Q_ℓ containing the integer points with the same omissions as in one dimension. The diameter of the unit cell is $\operatorname{Diam}(\operatorname{Unit} U) < 2\ell + 5$, the packing radius is $r = 0.5$, and the ratio is $\omega < 4\ell + 10$. As indicated in Figure 5.4, A_ℓ and Q_ℓ have parallel slices of missing points, with gaps of size $\ell + 1$ and $\ell + 3$ between the slices in A_ℓ, and gaps of size $\ell + 2$ between the slices in Q_ℓ. If follows that no two balls of radius $t_0 = \ell/2$ centered at missing points in different slices of A_ℓ (or of Q_ℓ) have a non-empty intersection. Hence, the k-fold covers of A_ℓ and Q_ℓ with balls of radius $t \le t_0$ have the same volume fractions for all k.

Note that no ball of radius $t_0 = \ell/2$ contains missing points of more than one slice. We can therefore use a volume argument to show that there are positive constants $c_0 < C_0$ such that the minimum number of points of A_ℓ (or of Q_ℓ) in any ball of radius t_0 satisfies $c_0 t_0^3 \le k_0 \le C_0 t_0^3$. Hence, the k-fold cover with balls of radius $t \ge t_0$ centered at the points in A_ℓ (or in Q_ℓ) exhausts the entire $\mathbb{R}^3$ whenever $k \le k_0$. It follows that for $k < k_0$, the k-th density functions do not distinguish between A_ℓ and Q_ℓ for any radius. This implies that $K \ge k_0 = \Omega(\omega^3)$ is necessary to distinguish A_ℓ and Q_ℓ with Ψ_K. □

Lemma 5.3.3 studies the dependence of K on ω. A slightly modified argument relates K to the motif size. Replacing A_ℓ by ℓA_1 before taking the Cartesian product with $\mathbb{Z}^2$, we get constant motif size equal to 4 and K increasing with ℓ. This implies that a bounded motif size does not imply a bounded K.

5.4 Computation

The algorithm for the density fingerprint map is based on two related geometric concepts: the k-th Dirichlet–Voronoi domain and the k-Brillouin zone of a point. After introducing both, we explain how they are used, and how much time it takes to construct them.

5.4.1 Dirichlet–Voronoi Domains and Brillouin Zones

Let $A \subseteq \mathbb{R}^3$ be a locally finite set of points. For every positive integer k, the *k-th Dirichlet–Voronoi domain* of a point $a \in A$ is the set of points in $\mathbb{R}^3$ for which a is among the k closest points in A, and the *k-th Brillouin zone* is the difference between the k-th and the $(k-1)$-st Dirichlet–Voronoi domains:

$$\mathrm{dom}_k(a, A) = \{x \in \mathbb{R}^3 \mid \|x - b\| < \|x - a\| \text{ for at most } k-1 \text{ points } b \in A\}, \tag{5.11}$$

$$\mathrm{zone}_k(a, A) = \mathrm{dom}_k(a, A) \setminus \mathrm{dom}_{k-1}(a, A); \tag{5.12}$$

see Figure 5.5. Here we set $\mathrm{dom}_0(a, A) = \emptyset$ so that the first Brillouin zone is well defined. Note that $\mathrm{zone}_k(a, A)$ is the set of points $x \in \mathbb{R}^3$ for which there are exactly $k-1$ points $b \in A$ that are closer to x than a is. Observe also that $\mathrm{dom}_k(a, A)$ is closed and star-convex, and if A is Delone, then it is also compact. If A is a lattice, $A = \Lambda$, then all k-th Dirichlet–Voronoi domains are translates of each other, and similarly for the Brillouin zones: $\mathrm{dom}_k(a, \Lambda) = \mathrm{dom}_k(0, \Lambda) + a$ and $\mathrm{zone}_k(a, \Lambda) = \mathrm{zone}_k(0, \Lambda) + a$. Except for a measure zero subset of $\mathbb{R}^3$, every point x has a unique k-closest point in Λ. This implies that the k-th Brillouin zones *tile* $\mathbb{R}^3$, by which we mean that their closures cover $\mathbb{R}^3$ while their interiors are pairwise disjoint. These properties generalize to a periodic point set, $A = M + \Lambda$: the k-th Brillouin zones of the points in $a + \Lambda$ are translates of each other, and the k-th Brillouin zones of all $a \in A$ tile $\mathbb{R}^3$.

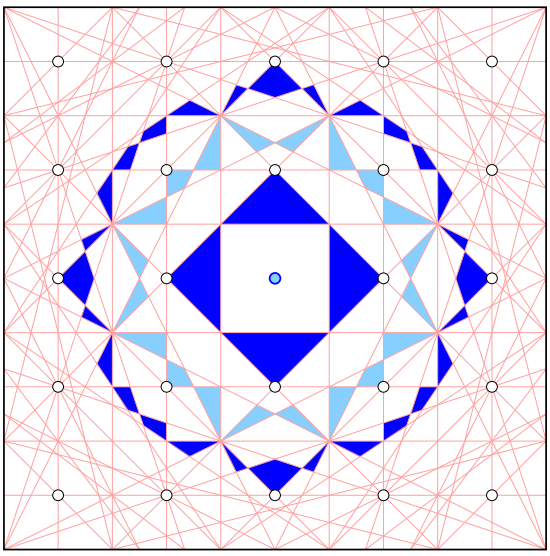

Figure 5.5: The Brillouin zones of the point in the center of the integer lattice $\mathbb{Z}^2$. The second and tenth Brillouin zones are colored *dark blue* and the sixth Brillouin zone is colored *light blue*. The bisector used to construct the first ten Brillouin zones are colored in red. This figure is a modification of a figure from [EGG+24].

5.4.2 Decomposed Multiple Cover

Assume from here on that $A = M + \Lambda$ is a periodic point set. To compute φ_k^A, we may use any fundamental domain of the lattice. Particularly convenient is the union of the k-th Brillouin zones of the points in M as it lends itself to finding the subset covered by at least k of the balls.

Theorem 5.4.1 (Density for periodic point set)**.** *Let $A = M + \Lambda$ be periodic with lattice $\Lambda \subseteq \mathbb{R}^3$ and motif $M \subseteq \operatorname{Unit} U$ in the unit cell of Λ, and let $k \geq 1$ be an integer. Then the probability that a random point $x \in \operatorname{Unit} U$ belongs to k or more balls of radius $t \geq 0$ centered at the points of A is*

$$\varphi_k^A(t) = \tfrac{1}{\operatorname{vol}(\operatorname{Unit} U)} \sum_{a \in M} \operatorname{vol}\left(\operatorname{zone}_k(a, A) \cap B(a; t)\right). \tag{5.13}$$

Proof. Let M_k be the union of the k-th Brillouin zones of the points $a \in M$ and note that $M_k + \Lambda$ tiles $\mathbb{R}^3$. It follows that $\operatorname{vol}(M_k) = \operatorname{vol}(\operatorname{Unit} U)$. Let $x \in M_k$ be in the interior of $\operatorname{zone}_k(a, A)$. By construction, a is the unique k-closest point to x, so x lies in k or more balls if and only if $x \in B(a; t)$. Summing over all points $a \in M$ gives (5.13). □

Clearly, $\varphi_0^A(t) = 1$ for all radii t. Given $k \geq 0$ and $t \geq 0$, we use (5.13) to compute $\varphi_k^A(t)$ and $\varphi_{k+1}^A(t)$, and we get $\psi_k^A(t) = \varphi_k^A(t) - \varphi_{k+1}^A(t)$. To implement (5.13), we need to compute the volume of the intersection of a ball with a convex polyhedron. We could, for example, decompose the polyhedron into tetrahedra and use explicit expressions for the volume of intersections between balls and simplices; see for example [ABI88]. A C++ implementation evaluating the density functions for a given periodic point set using this strategy can be found at [Smi20]. Alternatively, we could use inclusion-exclusion, which allows for further consolidation of the formula, writing $\varphi_k^A(t)$ as an alternating sum of common intersections of up to four balls each. This does not lead to any asymptotic improvements of the running time, so we omit further details and refer to [EIH18] instead.

5.4.3 Algorithm and Running Time

To evaluate the density functions $\psi_0^A, \psi_1^A, \ldots, \psi_k^A$ at a value t, we compute a plane arrangement for each point $a \in M$ that consists of enough planes so that the first $k + 1$ Brillouin zones of a occur. Specifically, for a large enough radius, s, we consider for each $b \neq a$ in $A \cap B(a; s)$ the *bisector* of b and a, which is the plane defined by $\|x - a\| = \|x - b\|$. These bisectors decompose $\mathbb{R}^3$ into convex cells. We refer to this decomposition as the *arrangement* of the planes. The 3-dimensional cells that are separated from a by exactly $j - 1$ planes form the *j-th belt* of the arrangement.

We now address the question how small we can choose s such that the first $k + 1$ belts are the first $k + 1$ Brillouin zones of a. To begin, we recall that $t \geq 2R\sqrt[3]{k + 1}$ implies that $\psi_i(t) = 0$ for $0 \leq i \leq k$; see the proof of Theorem 5.2.2. To express this insight geometrically, let $R_{k+1}(a)$ be the maximum distance of a point in the $(k + 1)$-st Brillouin zone of a from a. That the density functions ψ_0^A to ψ_k^A are zero for $t \geq 2R\sqrt[3]{k + 1}$ implies $\varphi_{k+1}^A(t) = 1$, for these values of t, and therefore $R_{k+1}(a) \leq 2R\sqrt[3]{k + 1}$. To capture all the relevant planes, it thus suffices to consider all points $b \in A \setminus \{a\}$ at distance at most $s = 2R_{k+1}(a)$ from a. Using a straightforward volume argument, we see that $B(a; 2R_{k+1}(a))$ contains at most $(4R\sqrt[3]{k + 1} + r)^3/r^3 = O(k)$ points, in which we treat r and R as constants.

Constructing the arrangement of $O(k)$ planes incrementally, as described in [Ede87, chapter 7], takes time $O(k^3)$. Doing this for each point in the motif takes time $O(\#M \cdot k^3)$, and within the same time bound we can evaluate the first $k + 1$ density functions.

5.5 An Application to Crystal Structure Prediction

Crystal Structure Prediction (CSP) aims to predict whether a selected molecule can exist as a functional material, i.e. a crystal with useful functions or properties. In other words, CSP seeks to answer the question of whether copies of a molecule can be arranged in such a way that the resulting crystal is stable (will not deform and lose its properties over time) as well as useful. Crucially, CSP tries to answer this question without setting foot in a laboratory, with the hope of dramatically reducing the need to perform the time-consuming process of physically synthesizing crystals.

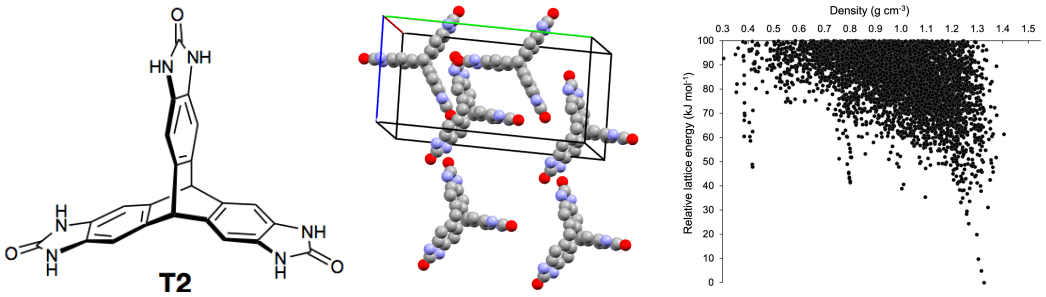

Figure 5.6: *Left*: a T2 molecule. *Middle*: the T2-δ crystal with highlighted unit cell. *Right*: the output of CSP for the T2 molecule. It is a plot of 5679 simulated T2 crystal structures [Pea17, Fig. 2d], each represented by two coordinates: the physical density (atomic mass within a unit cell divided by the unit cell volume) and energy (determining the crystal's thermodynamic stability). Structures at the bottom of the 'downward spikes' are likely to be stable.

Our collaborators at Liverpool's Materials Innovation Factory [Pea17] used CSP to predict that the T2 molecule (Figure 5.6) can be crystallized into a new structure that has half the physical density of the only previously known structure for T2, a desirable property for applications such as gas storage. As part of this process, they also identified four other structures of interest. Following the CSP predictions, they synthesized 5 families of T2-crystals in the laboratory by varying parameters like temperature and pressure, calling them T2-α, T2-β, ..., T2-ϵ. One of them, T2-γ, indeed had the desired property of having only half the physical density of the previously known structure T2-α. They scanned the synthesized crystals using X-ray powder diffraction yielding Crystallographic Information Files, each containing the unit cell and the motif points representing the atoms. These files were then compared with the results of the simulations, either by using their physical density alongside the COMPACK algorithm—which compares only a finite portion of the structure—or by looking at visualizations of the crystal structures. This comparison showed that the synthesized crystals matched the prediction well. Our collaborators deposited these structures into the globally used Cambridge Structural Database.

At a later time, we used our newly developed fingerprints to verify our collaborators' matchings between the synthesized crystals T2-α to T2-ϵ and the simulated crystals entry 99, 28, 62, 09, 01. We did so by computing, for each of the five matches, the distance between the density functions of the synthesized and the simulated crystal. As one is the prediction of the other, we expected to see small distances. And for four of the five structures this was true: T2-γ, for example, always has an L_∞-distance of less than 0.04 over the first eight pairs of corresponding density functions; see Table 5.2. However, when we came to check the distances between density functions of T2-δ with its predicted structure, we were

surprised to see large distances (the final row of Table 5.2). It turned out that a mix-up of files had happened, and what was uploaded to the Cambridge Structural Database as T2-δ was in fact T2-β' (a crystal from the T2-β family). The density fingerprint revealed this error, which was verified by chemists upon a visual inspection, and it is because of this that T2-δ was subsequently correctly deposited.

$\|\psi_k^A - \psi_k^Q\|_\infty$	$k=0$	1	2	3	4	5	6	7
T2-α vs entry 99	0.0042	0.0092	0.0125	0.0056	0.0099	0.0088	0.0127	0.0099
T2-β vs entry 28	0.0157	0.0156	0.0159	0.0224	0.0334	0.0396	0.0357	0.0454
T2-γ vs entry 62	0.0020	0.0080	0.0128	0.0155	0.0153	0.0250	0.0296	0.0391
T2-δ vs entry 09	0.0610	0.0884	0.1267	0.0676	0.0915	0.0801	0.0733	0.0388
T2-ϵ vs entry 01	0.0132	0.0152	0.0207	0.0571	0.0514	0.0431	0.0468	0.0550
T2-β' vs entry 09	0.2981	0.2631	0.3718	0.3747	0.2563	0.2360	0.3161	0.3232

Table 5.2: *First five rows:* the L_∞-distances between the first eight pairs of corresponding density functions of physically synthesized T2 crystals (T2-α, T2-β, etc.) and the simulated structures that had predicted them from the CSP output dataset (entry XX). *Last row:* the suspiciously larger numbers revealed the mix-up of the files T2-δ and T2-β' and thus led to depositing the initially omitted Crystallographic Information File of the T2-δ crystal into the Cambridge Structural Database.

Plots of the density functions of correctly matched synthesized and simulated structures can be seen in Figure 5.7. As another application, we expect that the fingerprint will be used to simplify the large output data sets produced by CSP by comparing simulated structures with each other, thus speeding up what is currently a slow process.

5.6 Discussion

The main contribution of this chapter is a fingerprint map from periodic point sets in $\mathbb{R}^3$ (which model crystals) to series of density functions. This map is obviously invariant under isometries, and we prove it is continuous and generically complete. We leave the completeness without genericity assumption as an open question. In this context, it is worth noticing that our proof of generic completeness makes only limited use of the order, k, at which the circumradius of an edge, triangle, or tetrahedron is detected. Recall that the order is the number of points in the respective circumsphere. Is this additional information sufficient to prove general completeness?

A drawback of the bottleneck distance between periodic point sets used in this chapter is its sensitivity to changes of the unit cell; see Lemma 5.2.1. An alternative dissimilarity that may be more relevant in practice considers affine transformations, τ, that minimize the bottleneck distance:

$$d_{\mathrm{AT}}(A,Q) = \inf_{\tau} \max\{\min\{d_B(A,\tau(Q)), d_B(\tau(A),Q)\}, |\log s_1|, |\log s_3|\}, \tag{5.14}$$

in which $s_1 \geq s_2 \geq s_3$ are the three singular values of the matrix of τ. Is the density fingerprint map defined in Section 5.1 continuous with respect to this dissimilarity?

We close this chapter with three extensions of the results presented in this chapter. Different types of atoms are often modeled as balls with different radii. A possible geometric formalism

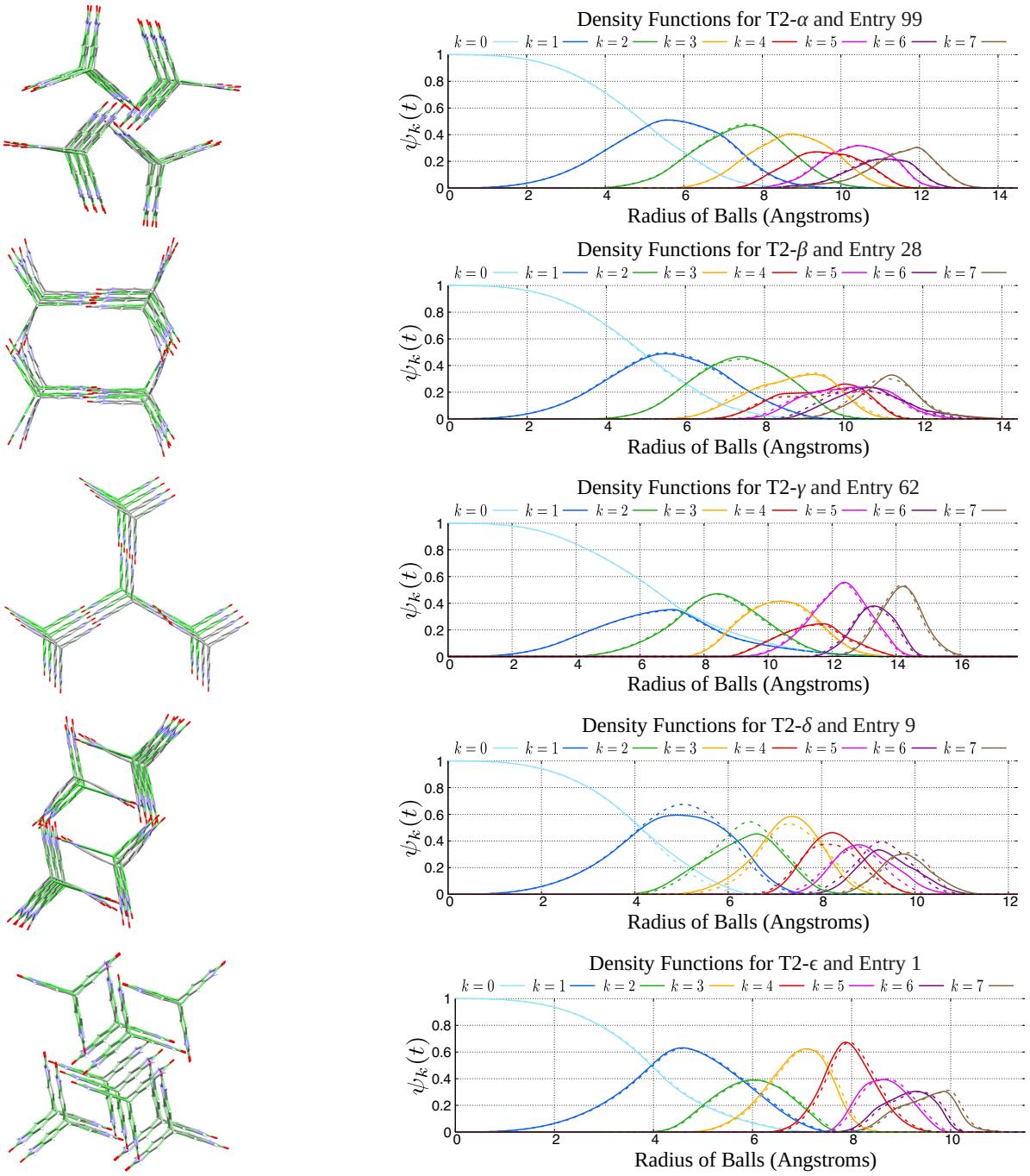

Figure 5.7: *Left*: experimental T2 crystals (curved gray molecules) and their simulated versions (straight green molecules) overlaid. *Right*: the density functions of the periodic point sets of molecular centres of the experimental T2 crystals (solid curves) vs. simulated crystals (dashed curves).

is that of weighted points and the power distance [Aur87]. Our geometric results generalize to this setting, but some need a careful adaptation. Our continuity result for periodic point sets (Theorem 5.2.2) also generalizes to non-periodic Delone sets that allow for a reasonable definition of density functions. Considering that quasiperiodic crystals can be modeled as such, it might be worthwhile to find out how far such an extension can be pushed. Finally, we mention that our results generalize to arbitrary finite dimension.

CHAPTER 6

Merge Trees of Periodic Filtrations

This chapter is based on [EH24]. Some of the results have already been described in the Master's thesis of Chiara Martyka [Mar23], where she cites an unpublished version of [EH24] under the title "The decorated barcode of a periodic map". In her Master's thesis, Chiara Martyka worked on an earlier version of the algorithm for periodic merge trees together with Herbert Edelsbrunner and me.

Motivated by applications to crystalline materials, we generalize the merge tree and the related barcode of a filtered complex to the periodic setting in Euclidean space. They are invariant under isometries, changing bases, and indeed changing lattices. In addition, we prove stability under perturbations and provide an algorithm that under mild geometric conditions typically satisfied by crystalline materials takes $O((n+m)\log n)$ time, in which n and m are the numbers of vertices and edges in the quotient complex, respectively.

Prior work. Prior approaches to persistent homology for periodic data relied primarily on one of two heuristics: work with a sufficiently large finite subset of the data, e.g. [HNH+16, KHM20], or compactify using the torus topology, which is a classic mathematical concept and supported by modern software, including the implementations of Delaunay triangulations in CGAL [C+96]. The first heuristic suffers from unwanted boundary effects and unnecessarily high computation costs, while the latter loses information about the patterns in the periodic setting. Our approach is related to both and reaps the benefits of the compactness of the torus while carefully recording the relation to the infinite periodic setting; see Figure 6.1. Indeed, each connected component on the torus has preimages (shadows) in the Euclidean space—perhaps infinitely many—and similar to Onus and Robins [OR22], we count these preimages but in a fine-grained manner that keeps track of growth-rates as well as densities (exponents and coefficients of the shadow monomials). The growth-rate distinguishes between connected clumps, strings, sheets, and blocks, while the density quantifies how densely these are distributed. Both pieces of information are needed for detailed statements about how materials are composed. The periodicity lattice—which is used in the definition of shadow monomials—has been introduced before under different names, and used to compute 0-homology [CM90, DD89, OR22]. However, using it to quantify the growth-rate and density of the components is to the best knowledge of the authors new.

Shifting the focus from counting components to recording when components merge, we mention the work of Ingrid Hotz and her group [TAS+23]. In collaborations with domain scientists, they study the electron density of 3-dimensional layered materials, and found

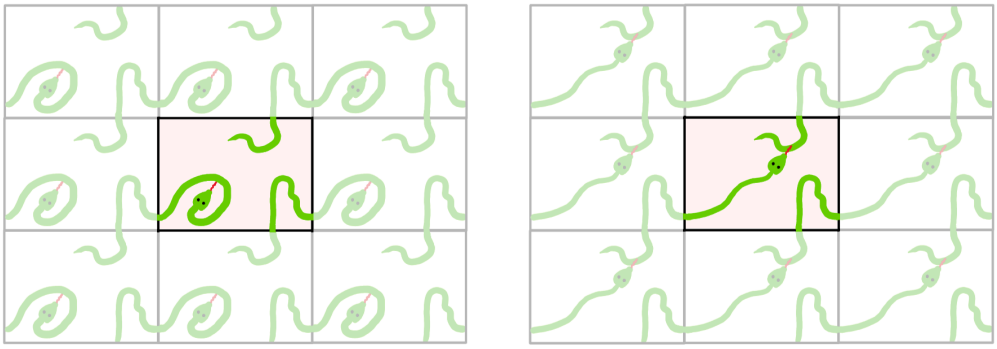

Figure 6.1: In a unit cell with periodic boundary conditions (the torus), we see a single snake that bites itself, both in the *left* and the *right panel*. There is however a significant difference in the periodically tiled plane, since the snakes on the *right* connect in infinite diagonal lines, while the snakes on the *left* remain isolated, a distinction we will quantify with the novel concept of a *shadow monomial*. The material properties of the two examples would indeed be rather different, with higher resistance to tearing on the *right*.

that the timing of the merge events is useful in the quick comparison of materials. They compromise the (infinite) periodic setting by considering merge events within an array of $2 \times 2 \times 2$ unit cells with periodic boundary conditions. Our method removes the restriction to a finite portion of the data and provides extra information, such as how many and which components join at a merge event.

Outline. Section 6.1 extends the concept of a merge tree to the periodic setting. Section 6.2 explains the algorithm for constructing periodic merge trees. Section 6.3 proves the invariance and stability of equivalence classes of periodic merge trees. Section 6.4 introduces the periodic 0-th barcode and proves invariance and stability. Section 6.5 illustrates the concepts using a 3-dimensional periodic graph as an example. Section 6.6 concludes this chapter.

6.1 The Periodic Merge Tree

This chapter is written for arbitrary but fixed dimension, d, with $d = 3$ being the most important case for applications, including to crystalline materials. We limit ourselves to the discrete setting of complexes or graphs, and stress that the ideas also apply to smooth and piecewise smooth functions.

6.1.1 Merge Tree

We begin with standard definitions. Recall the definition of a cell complex K from Section 2.1. Note that in this chapter, we allow for infinitely many points and cells but require that K be locally finite. The complex is *connected* if the graph that consists of its vertices and edges is connected. Since this chapter is primarily concerned with 0-dimensional homology—whose classes correspond to the connected components—we will mostly ignore cells of dimension 2 and higher. After removing such cells, we are left with a graph consisting of vertices and edges, allowing for edges that start and end at the same vertex, and multiple edges connecting the same two vertices. However, in anticipation of an extension of our methods to higher homology dimensions, and because the overhead is modest, we nonetheless write this chapter using the more general terminology of complexes.

Recall from Section 2.1 that a filter of K is a function $F\colon K \to \mathbb{R}$ that satisfies $F(\sigma) \leq F(\tau)$ whenever σ is a face of τ. And recall that the ordered sequence of sublevel set complexes $K_t = F^{-1}(-\infty, t]$ is the filtration defined by F. As K_s is a subcomplex of K_t, whenever $s \leq t$, every connected component of K_s includes into a connected component of K_t. Think of the sublevel set as an object that evolves as the threshold grows continuously. With this picture in mind, the merge tree of F keeps track of the connected components and how they merge. Intuitively, this tree has a point for each connected component of each sublevel set. We draw the tree from left to right, nevertheless calling the threshold for the sublevel set the *height* of the point. As long as a connected component stays the same or just grows, we draw the point along a left-to-right trajectory, but when components merge, then the corresponding trajectories meet, and afterwards continue as a single trajectory again in left-to-right direction. Since trajectories meet but never bifurcate, and—assuming K is connected—eventually combine to a single trajectory, we indeed have a tree, as suggested by the name. The number of points in a vertical section of this tree is the number of connected components of the corresponding sublevel set, and the connectivity in the left-to-right direction reflects the inclusion between connected components in different sublevel sets; see Figure 6.2, which shows two merge trees of the same periodic point set. Each tree is drawn as a collection of horizontal intervals, called *beams*, which we connect with short vertical line segments to represent the merging of two components.

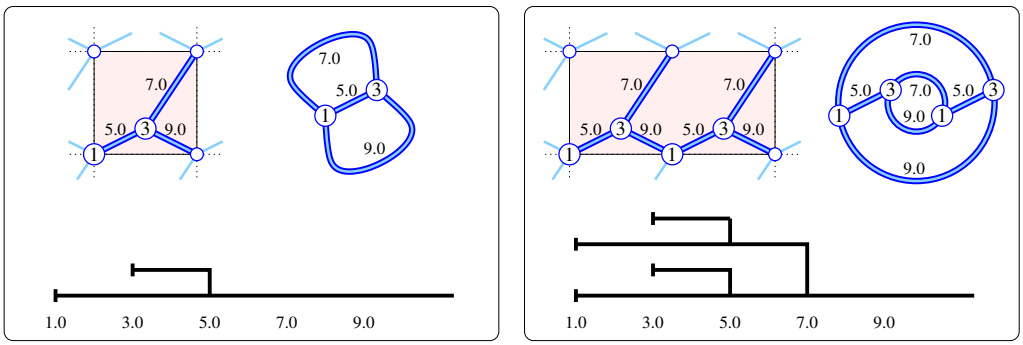

Figure 6.2: *Left panel:* a periodic graph with two vertices and three edges inside a unit cell in the shape of the unit square in the *upper left portion*, and the corresponding graph in the 2-dimensional torus in the *upper right portion of the panel*. The filter maps the vertices to their (real) labels and the edges to the values shown, which defines the merge tree at the *bottom in the panel*. *Right panel:* the same periodic graph as in the *left panel*, but now represented by a sublattice with a rectangular unit cell of twice the area. Correspondingly, the graph in the 2-dimensional torus has twice the number of vertices and edges, and the merge tree is richer than in the *left panel*.

Note the asymmetry, in which we always draw the vertical segment from the shorter to the longer beam (comparing their lengths to the left of the bifurcation point). As a result, the shorter beam ends, and the longer beam continues without interruption to the right. While not being part of the definition of merge tree, this asymmetry will come handy when we transition from merge trees to barcodes in Section 6.4. We call this the *elder rule* according to which the younger component gets absorbed into the older component; see [EH10, page 150]. Ties are broken arbitrarily, and we will see in Section 6.4 that different ways to break a tie does not affect the barcode.

To give a more formal definition of the merge tree, we introduce the *sublevel space* of F, denoted $[F]$, which consists of all points $(x, s) \in K \times \mathbb{R}$ with $x \in K_s$, in which x may be a

vertex or a point on an edge or higher-dimensional cell in K. Call two of its points *equivalent*, written $(x, s) \sim (y, t)$, if $s = t$ and x, y belong to the same connected component of $K_s = K_t$. The *quotient*, denoted $[F]_\sim$, effectively combines all equivalent points into one and inherits the *quotient topology* from the Euclidean topology of $\mathbb{R}^{d+1}$ restricted to $[F] \subseteq \mathbb{R}^{d+1}$.

Definition 6.1.1 (Merge Tree)**.** The *merge tree* of a filter $F\colon K \to \mathbb{R}$ is the quotient of the sublevel space of F, denoted $\mathbb{M} = \mathbb{M}(F) = [F]_\sim$, together with the corresponding quotient topology. It is equipped with the *height function*, $\mathrm{h}\colon \mathbb{M} \to \mathbb{R}$, which maps each point to the value of the sublevel set in which the point is a connected component.

Whenever we talk about a merge tree, we will tacitly assume that it comes with a height function. We will furthermore simplify language by saying that points α and β of the merge tree not only represent but in fact *are* connected components of their respective sublevel sets, namely of $K_{\mathrm{h}(\alpha)}$ and $K_{\mathrm{h}(\beta)}$, respectively. We say β *covers* α if $\mathrm{h}(\alpha) \leq \mathrm{h}(\beta)$ and $\alpha \subseteq \beta$.

In this chapter, we are particularly interested in periodic filters on periodic complexes. If the periodic filter is not constant, the merge tree consists of infinitely many beams, making it an impractical descriptor. Instead we can use the quotient filter $F/\Lambda\colon K/\Lambda \to \mathbb{R}$ as a finite representation of the periodic filter $F : K \to \mathbb{R}$ and compute the (finite) merge tree $\mathbb{M}(F/\Lambda)$, as done in Figure 6.2.

Unfortunately, this representation is not unique since, for example, K and F are also periodic with respect to 2Λ, or really any sublattice of Λ; compare the two panels in Figure 6.2 in which the respective unit cells (a square on the left and a rectangle of twice the area on the right) define the same periodic complex but yield different merge trees. This motivates the introduction of a quantified version of the merge tree; see Figure 6.4. We first explain how we quantify and then how we apply this idea to modify the merge tree.

6.1.2 Shadow Monomial

Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, and $F\colon K \to \mathbb{R}$ a Λ-periodic filter on a Λ-periodic complex. Write $\phi\colon \mathbb{R}^d \to \mathbb{R}^d/\Lambda$ for the projection that maps every x to $x + \Lambda$. Taking the inverse, a sublevel set of the quotient filter lifts to the corresponding sublevel set of the filter: $K_t = \phi^{-1}((K/\Lambda)_t)$. Given a component $\Gamma \subseteq (K/\Lambda)_t$, we call a component γ of $\phi^{-1}(\Gamma)$ a *shadow* of Γ. As an example, consider the sublevel set of the graph in Figure 6.2 before adding the edge with value 9.0. The quotient is a loop whose shadows are infinite polygonal lines that run parallel to each other in a diagonal direction, as shown in Figure 6.3. To count the distinct shadows of Γ, we introduce the *periodicity lattice* of all vectors that keep the shadows invariant:

$$\Lambda_\Gamma = \{u \in \Lambda \mid \gamma = \gamma + u \text{ for at least one and therefore every shadow } \gamma \text{ of } \Gamma\}. \tag{6.1}$$

For example, translating the shadow in Figure 6.3 along the vector $(1, 1)$ maps the polygonal line to itself. All other vectors for which this holds are integer multiples of this vector. Hence, the periodicity lattice of the loop is the 1-dimensional lattice spanned by $(1, 1)$. With this, counting the shadows of Γ reduces to counting the elements of Λ/Λ_Γ. If the cardinality of this quotient is finite, then this gives the number of shadows. On the other hand, if the cardinality is infinite, we wish to count the elements that appear within a spherical window of radius R. To this end, we introduce a monomial in the variable R, such that the coefficient gives the density and the exponent gives the growth-rate. Write ν_q for the q-dimensional volume of the unit ball in $\mathbb{R}^q$, so $\nu_0 = 1$, $\nu_1 = 2$, $\nu_2 = \pi$, $\nu_3 = \frac{4\pi}{3}$, etc.

Definition 6.1.2 (Shadow Monomial)**.** Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, and $F\colon K \to \mathbb{R}$ a Λ-periodic filter. Letting $\Gamma \in \mathbb{M}(F/\Lambda)$ be a connected component of a sublevel set of the quotient filter, and p the dimension of the corresponding periodicity lattice, Λ_Γ, the *shadow monomial* of Γ is

$$\mathrm{m}_\Gamma(R) = \frac{\mathrm{vol}_p\,(\Lambda_\Gamma)}{\mathrm{vol}_d\,(\Lambda)} \cdot \nu_{d-p} R^{d-p}. \tag{6.2}$$

Up to lower-order terms, this monomial gives the number of elements of Λ/Λ_Γ that have points inside the sphere of radius R centered at the origin in $\mathbb{R}^d$. For example, the loop whose shadows are the polygonal lines in Figure 6.3, we have $\mathrm{vol}_1\,(\Lambda_\Gamma) = \sqrt{2}$, $\mathrm{vol}_2\,(\Lambda) = 1$, and $\nu_1 = 2$, so $\mathrm{m}_\Gamma(R) = 2\sqrt{2}R^1$, and up to a constant error term this is the number of polygonal lines that pass through a disk of radius R. This property is fundamental to our results, so we give a formal statement and a proof.

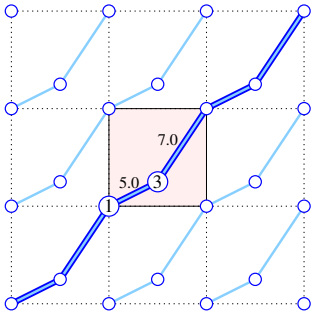

Figure 6.3: Each shadow of the loop in the quotient complex is an infinite polygonal line with periodicity lattice spanned by the vector $(1, 1)$. The length of its unit cell is $\sqrt{2}$, which implies that its shadow monomial is $2\sqrt{2}R$.

Lemma 6.1.3 (Counting Inside Sphere)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice and $\Lambda_\Gamma \subseteq \Lambda$ a p-dimensional sublattice. Then the number of elements of Λ/Λ_Γ that have a non-empty intersection with the ball of radius R centered at $0 \in \mathbb{R}^d$ is $\mathrm{m}_\Gamma(R) + O(R^{d-p-1})$.*

Proof. We begin with the two extreme cases. In the zero-dimensional case, when $p = 0$, we have $\Lambda_\Gamma = \{0\}$ and $\Lambda/\Lambda_\Gamma = \Lambda$. We thus count the points of Λ inside the d-ball of radius R centered at 0, which up to lower-order terms is the volume of the ball divided by the volume of the unit cell: $\nu_d R^d/\mathrm{vol}_d\,(\Lambda) = \mathrm{m}_\Gamma(R)$ because $\mathrm{vol}_0\,(\Lambda_\Gamma) = 1$. Observe that this estimate also holds if we count the points inside the d-ball with radius $R - 1$. In the full-dimensional case, when $p = d$, we have $\dim \Lambda_\Gamma = \dim \Lambda$, so Λ/Λ_Γ is finite, with cardinality $\mathrm{vol}_d\,(\Lambda_\Gamma)\,/\mathrm{vol}_d\,(\Lambda)$. Indeed, any unit cell of Λ_Γ contains a point from each class in the quotient, so for a large enough constant R, the d-ball with radius R contains a point from each class. Furthermore, $\mathrm{vol}_d\,(\Lambda_\Gamma)\,/\mathrm{vol}_d\,(\Lambda) = \mathrm{m}_\Gamma(R)$ because $\nu_0 R^0 = 1$.

In the general case, let Λ' be the maximal sublattice of Λ that satisfies $\mathrm{span}(\Lambda') = \mathrm{span}(\Lambda_\Gamma)$. By construction, $p = \dim \Lambda' = \dim \Lambda_\Gamma$, so we can apply the result for the full-dimensional case, in which we count exactly $\mathrm{vol}_p\,(\Lambda_\Gamma)\,/\mathrm{vol}_p\,(\Lambda')$ translates of Λ_Γ within Λ'. Next project Λ orthogonally onto the linear subspace orthogonal to $\mathrm{span}(\Lambda')$, which gives a $(d-p)$-dimensional lattice, Λ''. The unit cells satisfy $\mathrm{vol}_p\,(\Lambda') \cdot \mathrm{vol}_{d-p}\,(\Lambda'') = \mathrm{vol}_d\,(\Lambda)$. Using the result for the zero-dimensional case, we count $\nu_{d-p} R^{d-p}/\mathrm{vol}_{d-p}\,(\Lambda'') + O(R^{d-p-1})$ points of

Λ'' within the ball of radius R, and multiplying the two counts while ignoring the lower order term, we get

$$\frac{\mathrm{vol}_p\,(\Lambda_\Gamma)}{\mathrm{vol}_p\,(\Lambda')} \cdot \nu_{d-p} R^{d-p} \frac{1}{\mathrm{vol}_{d-p}\,(\Lambda'')} = \frac{\mathrm{vol}_p\,(\Lambda_\Gamma)}{\mathrm{vol}_p\,(\Lambda')} \frac{\mathrm{vol}_p\,(\Lambda')}{\mathrm{vol}_d\,(\Lambda)} \cdot \nu_{d-p} R^{d-p} = \mathtt{m}_\Gamma(R). \tag{6.3}$$

This expression measures the product of the $(d-p)$-ball with radius R and the p-ball with constant radius. This product is not contained in the d-ball of radius R, but if we shrink the $(d-p)$-ball to radius $R-1$, then it is contained in the d-ball, assuming R is sufficiently large. This changes the count only by a lower-order term, which implies the claim in the general case. □

6.1.3 Periodic Merge Tree

To cope with the potentially infinitely many connected components of a periodic complex, we construct the merge tree for a finite representation in the torus and decorate this tree with a shadow monomial at every point. Each shadow monomial is specified by a real coefficient and an integer exponent, and we write $\mathcal{M}[R]$ for the set of such monomials.

Definition 6.1.4 (Periodic Merge Tree)**.** Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, and $F\colon K \to \mathbb{R}$ a Λ-periodic filter. The *periodic merge tree* of F with respect to Λ, denoted $\mathbb{M}(F,\Lambda)$, is the merge tree of the quotient filter, $\mathbb{M} = \mathbb{M}(F/\Lambda) = [F/\Lambda]_\sim$, together with its height function, $\mathtt{h}\colon \mathbb{M} \to \mathbb{R}$, and the *frequency function*, $\mathtt{f}\colon \mathbb{M} \to \mathcal{M}[R]$, which maps each point $\Gamma \in \mathbb{M}$ to its shadow monomial.

Whenever we talk about a periodic merge tree, we will tacitly assume that it comes with its height and frequency functions. The notational ambiguity between merge trees and periodic merge trees is deliberate and emphasizes that every periodic merge tree is also a merge tree. See Figure 6.4 for two examples, in which we write the shadow monomials above the beams of the two periodic merge trees. For notational reasons, we write $\mathtt{m}_\Gamma(R) = \mathtt{f}(\Gamma)(R)$ when a value of the frequency function is applied to a radius, R. Since connected components tend to grow with increasing threshold, it is plausible that the shadow monomial can only get smaller when it changes as we move from left to right in the merge tree. To make this intuition concrete, we write $tR^{d-q} < sR^{d-p}$ if either $d-q < d-p$ or $d-q = d-p$ and $t < s$, and we write $tR^{d-q} \leq sR^{d-p}$ if equality is allowed.

Lemma 6.1.5 (Monotonicity)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, $F\colon K \to \mathbb{R}$ a Λ-periodic filter, and $\mathbb{M} = \mathbb{M}(F,\Lambda)$ the periodic merge tree of F with frequency function $\mathtt{f}\colon \mathbb{M} \to \mathcal{M}[R]$. If a point $\mathrm{B} \in \mathbb{M}$ covers another point $\mathrm{A} \in \mathbb{M}$, then $\mathtt{f}(\mathrm{B}) \leq \mathtt{f}(\mathrm{A})$.*

Proof. Since B covers A, the periodicity lattice of A is a sublattice of that of B: $\Lambda_\mathrm{A} \subseteq \Lambda_\mathrm{B}$. Indeed, if $\alpha = \alpha + u$ for every shadow α of A, then $\beta = \beta + u$ for every shadow β of B, but not necessarily the other way round. The exponents of $\mathtt{f}(\mathrm{A})$ and $\mathtt{f}(\mathrm{B})$ are $d - \dim \Lambda_\mathrm{A}$ and $d - \dim \Lambda_\mathrm{B}$. If $\dim \Lambda_\mathrm{A} < \dim \Lambda_\mathrm{B}$, then $\mathtt{f}(\mathrm{B}) < \mathtt{f}(\mathrm{A})$, and if $\dim \Lambda_\mathrm{A} = \dim \Lambda_\mathrm{B}$, then the coefficients determine the order, and the only difference between them is the volume of the periodicity lattice, which is at least as large for A as it is for B. We thus get $\mathtt{f}(\mathrm{B}) \leq \mathtt{f}(\mathrm{A})$ in general. □

Since the shadow monomial can only decrease along a left-to-right trajectory in the merge tree, its exponent progresses from d down to 0, possibly skipping some integers. We

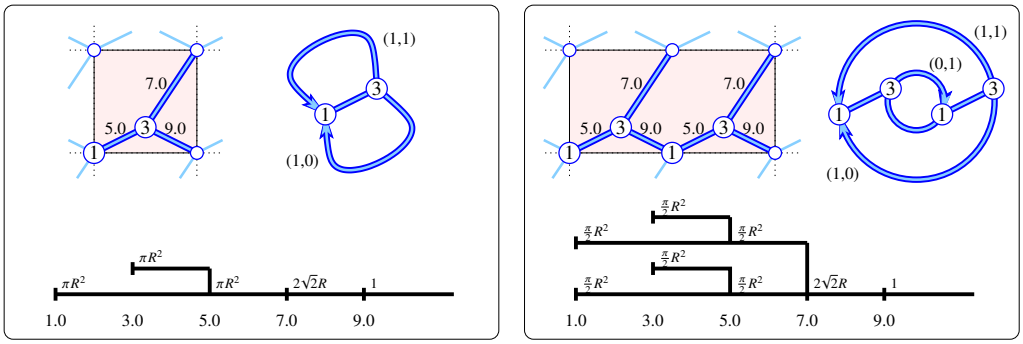

Figure 6.4: The same graphs and periodic merge trees as in Figure 6.2 but with additional information. Edges with non-zero shift vectors (to be defined in Section 6.2.2) are drawn as (directed) arcs and labeled with their shift vectors, while edges with zero shift vectors remain undirected and without vector. In addition to the appearances at $t = 1.0, 3.0$ and the merger at $t = 5.0$, there are two catenations at $t = 7.0, 9.0$ that define the shadow monomials decorating the beams of the periodic merge trees in the *left panel*. Note that the tree in the *right panel* has twice as many subtrees rooted at the point labeled 7.0, and that the shadow monomials compensate for the increased number of beams. Indeed, we have two events at each of the first four values defining the periodic merge tree, with a merger followed by a catenation at $t = 7.0$.

therefore define the $(d-p)$*-ary era* of a beam as the interval of points whose shadow monomials have exponent $d-p$. For example, the longest beams of the periodic merge trees in Figure 6.4 have three eras each, characterized by the exponent of R, which shrinks from left to right. We further subdivide each era into *epochs*, which are maximal intervals in which the coefficient of the shadow monomial stays constant. From one epoch to the next on the same beam, the periodicity lattice grows to a superlattice, which implies a strengthening of Lemma 6.1.5, namely that the coefficient drops by division with an integer larger than 1. The graph in Figure 6.4 is unfortunately too small to have eras consisting of more than one epoch, but see Section 6.5 for a more elaborate example in $\mathbb{R}^3$.

6.2 The Algorithm

This section presents the algorithm we use to construct the periodic merge tree of a Λ-periodic filter, $F\colon K \to \mathbb{R}$. This tree is but the merge tree of the quotient filter, $F/\Lambda\colon K/\Lambda \to \mathbb{R}$, equipped with the frequency function. The construction of the merge tree has been amply studied prior to this chapter; see e.g. [SM20], so we will focus on the frequency function.

6.2.1 Critical Events

Similar to Kruskal's classic algorithm for minimum spanning trees [Kru56], we process the cells of K/Λ in the order of the filter and update the periodic merge tree at each step. The only relevant cells are the vertices and edges, and we distinguish the edges in the quotient complex whose endpoints lie in the same and in different connected components. In the former case, we consider a shadow of the edge and further distinguish when its endpoints belong to the same or to different shadows of the connected component. We therefore have three types of critical events that characterize the evolution of the connected components:

- *appearance:* a vertex is added, which starts a new connected component or, equivalently, a new beam of the periodic merge tree;
- *merger:* an edge with endpoints in two different connected components is added, which merges these components into one or, equivalently, ends a beam by joining it to another in the merge tree;
- *catenation:* an edge with both endpoints in the same connected component is added and the endpoints of its shadow belong to two different shadows of that component, which enlarges the periodicity lattice of the component and shrinks its shadow monomial;

see Figure 6.4, where the tree in the left panel experiences two appearances, one merger, and two catenations, while the tree in the right panel experiences four appearances, three mergers, and two catenations. To add a new vertex in a appearance, we set its periodicity lattice to $\{0\}$. To add a new edge, e, let A, B be the connected components that contain its endpoints. If $\mathrm{A} = \mathrm{B}$, then e together with a path connecting its endpoints form a loop in the quotient complex, and the shadow of this loop can either be a loop in the periodic complex, or a path that connects different shadows of the same vertex. Hence, the addition of the edge has either no effect or it enlarges the periodicity lattice of the component and thus decreases the shadow monomial. Writing Λ_e for the periodicity lattice of the loop, and writing Γ for the component A after adding e, the periodicity lattice of Γ is the smallest common superlattice: $\Lambda_\Gamma = \Lambda_\mathrm{A} + \Lambda_e$. On the other hand, if $\mathrm{A} \neq \mathrm{B}$, the addition of e merges the two components, and we write Γ for the new component, which is the union of A, B, and e. The periodicity lattice of Γ is again the smallest common superlattice: $\Lambda_\Gamma = \Lambda_\mathrm{A} + \Lambda_\mathrm{B}$.

6.2.2 Periodicity Lattice and Loops

There is a direct relation between the periodicity lattice of a connected component and its loops. Since our primary concern are components (and not cycles of dimension 1 or higher), we may assume that a component, Γ, is an undirected graph. Nonetheless, we need to reason about loops in this graph, and this is done more conveniently in the representation of Γ as a directed graph, in which the (undirected) *edge* connecting x to y is replaced by two (directed) *arcs*, one from x to y and the other from y to x.

Definition 6.2.1 (Paths and Loops)**.** In a directed graph, a *path* from x to y is a sequence of arcs, a_i from x_i to x_{i+1}, for $1 \leq i \leq k$, with $x_1 = x$ and $x_{k+1} = y$. This path is a *loop* if, in addition, $x = y$. A path or loop is *simple* if no arc is repeated.

In our context, each arc is labeled with an integer vector that records the relative position of the unit cells where a trajectory that is a shadow of the arc starts and ends. Specifically, let a be an arc from x to y, write x_0, y_0 for the shadows of the endpoints inside a common unit cell, and let $x_0 + u, y_0 + w$ be the endpoints of a shadow of the arc. Then the *shift vector* of the arc is $\mathrm{Shift}(a) = U^{-1}(w - u)$. Correspondingly, the shift vector of the arc from y to x is $\mathrm{Shift}(-a) = -\mathrm{Shift}(a)$. The *drift vector* of a path is the sum of shift vectors of its arcs, and similarly for a loop. Assuming the directed graph represents a connected undirected graph, Γ, we can designate one of its vertices as the *root* and extend every loop so it starts and ends at the root without changing its drift vector. Any two loops can therefore be concatenated while adding their drift vectors. The collection of drift vectors thus satisfies the properties of a lattice: for every vector we also have its integer multiples, and for every

two vectors we also have their sum. Hence, this collection is isomorphic to a sublattice of $\mathbb{Z}^d$, and we write $V \subseteq \mathbb{Z}^d$ for its basis. The periodicity lattice of the undirected graph, Λ_Γ, has basis UV.

EXAMPLE. Consider the right panel of Figure 6.4. The edges of the quotient graph with non-zero shift vectors are drawn directed to avoid possible ambiguities. The counterclockwise version of the *upper loop* consists of the edges with filter values $5.0, 7.0, 5.0, 7.0$ and drift vector $v_1 = (0,0) + (0,1) + (0,0) + (1,1) = (1,2)$, which means this loop moves one unit cells to the right and two unit cells up. The basis vectors that span the rectangular unit cell are $u_1 = (2.0, 0.0)$ and $u_2 = (0.0, 1.0)$, so this drift vector corresponds to

$$U \cdot v_1 = \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix}, \tag{6.4}$$

which is a vector in Λ. Its length is $2\sqrt{2}$, which explains the shadow monomial decorating the epoch that starts at $t = 7.0$ and ends at $t = 9.0$. After adding the two edges with filter value 9.0, we get additional loops: the *inner loop* with drift vector $v_2 = (0,0) - (0,1) = (0,-1)$, and the *lower loop* with drift vector $v_3 = (0,0,) + (0,0) + (0,0) - (1,0) = (-1,0)$. All other loops are concatenations of these three loops, and thus their drift vectors are sums of these three drift vectors. Writing V for the matrix whose columns are v_1, v_2, v_3, the corresponding periodicity lattice is thus spanned by

$$U \cdot V = \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -1 \\ 2 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 2.0 & 0.0 & -2.0 \\ 2.0 & -1.0 & 0.0 \end{bmatrix}. \tag{6.5}$$

The last vector is redundant, so we may choose the first two as a basis. They span a unit cell of area 2.0, which is the same area as the unit cell of Λ (the shaded rectangle). This explains the shadow monomial decorating the last epoch in the periodic merge tree.

6.2.3 Implementing with Union-Find

For the computation of the periodicity lattice, it suffices to look at the drift vectors of a basis of the loop space. We work with a basis defined by a spanning tree of Γ: each edge not in this tree belongs to a unique simple cycle in which all other edges are taken from the spanning tree. Choosing a direction for this edge, we get a corresponding simple loop anchored at the root of Γ, and the basis consists of one such loop for each edge not in the spanning tree. It is convenient to use the spanning tree that consists of the edges corresponding to past merge events for this purpose. The construction of this tree can be reduced to a sequence of *find* and *union* operations—the former are used to decide whether the two endpoints of an edge belong to the same or to different components, and the latter merge the two components that contain the endpoints, assuming they are different.

To put the theory into practice, we use a data structure that supports constant time find and amortized logarithmic time union operations. Faster implementations of the *union-find* data type exist; see e.g. the survey in [GI91], but they are not required for constructing the periodic merge tree within the desired time bound. The data structure stores the vertices of Γ in a linked list whose first vertex is the root r of Γ. We write $\mathrm{Next}(x)$ for the successor of x, which is `null` if x is last. In addition, x stores a link to the root, $\mathrm{Root}(x) = r$, and the drift vector of the unique simple path from r to x inside the spanning tree, denoted $\mathrm{Drift}(x)$. Finally, the root stores the number of vertices in the component, denoted $\mathrm{Size}(r)$, the vertex

with smallest filter value in Γ, denoted $\mathrm{Old}(r)$, as well as an integer matrix, $\mathrm{Basis}(r) = V$, such that UV is a basis of the periodicity lattice of Γ.

Given an edge with endpoints x, y, the find operation returns their respective roots: $r = \mathrm{Root}(x)$ and $s = \mathrm{Root}(y)$. The vertices belong to the same component if and only if $r = s$, so we can distinguish between the cases in which the edge forms a loop or connects two components in constant time. In the former case, we get a new path from r to y, possibly with a different drift vector, which requires updating the periodicity lattice. Let a be the directed version of the edge, leading from x to y:

Algorithm 3 Possible catenation event

```
01  if r = s then v = Drift(x) + Shift(a) − Drift(y);
02                Basis(r) = REDUCE(Basis(r), v)
03  endif.
```

In the latter case, we form the union of the two components. To do this efficiently, we pick the smaller component, update the information at all its vertices, and insert the updated list right after the root of the larger component. In addition, we compute the periodicity lattice of the union:

Algorithm 4 Merger event

```
01  if r ≠ s then assume Size(s) ≤ Size(r);
02      Old(r) = min{Old(r), Old(s)}; Basis(r) = REDUCE(Basis(r), Basis(s));
03      v = Drift(x) + Shift(a) − Drift(y); z = s;
04      while z ≠ null do
05          Root(z) = r; Drift(z) = v + Drift(z); last = z; z = Next(z)
06      endwhile;
07      Size(r) = Size(r) + Size(s); Next(last) = Next(r); Next(r) = s
08  endif.
```

The algorithm for updating the periodicity lattice, including its running time, will be discussed in the next subsection. The running time of all other operations is easily analyzed. The periodic merge tree of a Λ-periodic filter, $F\colon K \to \mathbb{R}$, is constructed incrementally, by adding one vertex or edge at a time, and each addition gives rise to at most two find operations and at most one union operation. Letting n be the number of vertices of K/Λ, there are $n-1$ union operations in total. Each vertex that receives a new drift vector ends up in a component at least twice the size of its old component. Hence, every vertex changes its root and drift vector at most $\log_2 n$ times. Letting m be the number of edges of K/Λ, we observe that each find operation takes constant time, which implies that $O(n \log n + m)$ time suffices for this part of the algorithm.

6.2.4 Euclid's Algorithm for Lattices

Next, we explain how the bases of two periodicity lattices can be processed to give the basis of their sum. Given sublattices, $\Lambda', \Lambda'' \subseteq \Lambda$, the vectors in their bases span $\Lambda' + \Lambda''$,

which is the smallest sublattice of Λ that contains Λ' and Λ''. Since there are possibly more vectors than necessary, we reduce the matrix of basis vectors to its *Hermite normal form*; see Schrijver [Sch98] for general background. Working with the corresponding integer matrices, this form can be computed in time polynomial in the size of the matrix and its entries; see [KB79] for the historically first such algorithm. To be self-contained, we give a simple algorithm that reduces the matrix by repeated application of Euclid's algorithm for the greatest common divisor (gcd) of two integers; see also [HMM98]. We focus on constant size matrices and sacrifice the polynomial running time for the simplicity of the algorithm.

Let M be a $d \times c$ integer matrix, and interpret its columns as vectors that span a sublattice of $\mathbb{Z}^d$. The algorithm reduces M using only three types of column operations: multiply a column with -1, exchange two columns, and subtract one column from another. An easy but important observation is that these operations preserve the lattice spanned by the columns. We reduce M from left to right such that the non-zero columns of the resulting lower triangular matrix define a basis of the lattice:

Algorithm 5 Hermite Normal Form reduction algorithm based on Euclid's algorithm

```
01  i = j = 1;
02  while i ≤ d do
03     if ∃ℓ with j ≤ ℓ ≤ c and M[i,ℓ] ≠ 0 then assume M[i,j] > 0;
04        for k = j + 1 to c do assume M[i,k] ≥ 0;
05           while M[i,j] > 0 and M[i,k] > 0 do
06              subtract ⌊M[i,j]/M[i,k]⌋ times column k from column j;
07              exchange columns j and k
08           endwhile
09        endfor;
10        for k = 1 to j − 1 do
11           subtract ⌊M[i,k]/M[i,j]⌋ times column j from column k
12        endfor; j = j + 1
13     endif; i = i + 1
14  endwhile.
```

The `while`-loop in lines 05 to 08 effectively computes the gcd of the i-th entries of columns j and k; that is: $M[i,j] = \gcd(M[i,j], M[i,k])$ and $M[i,k] = 0$. This is prepared by ascertaining $M[i,j] > 0$ and $M[i,k] \geq 0$ in lines 03 and 04 by the possible exchange of columns j and ℓ and by possible multiplication with -1. After finishing the `for`-loop in lines 04 to 09, all entries that succeed $M[i,j]$ in its row are zero, and after finishing the `for`-loop in lines 10 to 12, all entries that precede $M[i,j]$ in its row are non-negative and smaller than $M[i,j]$. After running the algorithm, each column has strictly more leading zeros than the preceding column. Hence, there are at most d non-zero columns left, and they form a basis of the lattice defined by M. Indeed, the matrix is in Hermite normal form, as defined in [Mad00] for the not necessarily full-rank case.

The running time depends, among other things, on the entries in M. To study this dependence, we introduce the *magnitude* of M as the maximum absolute entry in the matrix, denoted $\|M\|_\infty$. Similarly, we talk about the magnitude of an integer vector and the magnitude of a basis of integer vectors. Setting $N = \|M\|_\infty$, we show that the algorithm reduces

M in time at most logarithmic in N. Note however that there are faster variants of this algorithm; see e.g. [HMM98, Sto00].

A key ingredient of the proof is Bézout's identity, which is a statement about Euclid's algorithm for computing the gcd of positive integers $s \leq t$. Specifically, it asserts that there are integers x and y such that $xs + yt = \gcd(s,t)$ and $|x|, |y| \leq t$. These coefficients are computed by the *extended Euclidean algorithm* from the quotients that appear during the execution of Euclid's algorithm. The latter substitutes 0 and $\gcd(s,t)$ for s and t, so it is not surprising that there are also integers x_0 and y_0 such that $x_0 s + y_0 t = 0$ and $|x_0|, |y_0| \leq t$, and they can similarly be computed from the quotients in Euclid's algorithm. In summary, the actions of Euclid's algorithm are equivalent to substituting $xs + yx$ and $x_0 s + y_0 t$ for s and t, and the magnitude of the four coefficients does not exceed t.

Lemma 6.2.2 (Time for Reduction)**.** *Given a lattice spanned by the columns of a $d \times c$ matrix of integer entries with absolute values at most N, the reduction algorithm takes time at most $O(d^2 c^d \log(2N))$ to construct a basis of the lattice.*

Proof. The algorithm consists of four loops, of which the outer `while`-loop iterates at most d times, and the two `for`-loops together iterate $c - 1$ times. To bound the number of iterations of the inner `while`-loop, let μ_i be the maximum absolute entry that appears in row i at any time during the algorithm. We have $\mu_1 \leq N$, because the reduction of the first row does not increase its entries. However, it may increase entries in the other rows. To understand by how much, we consider the side effect of Euclid's algorithm computing the gcd of $s = M[1,j]$ and $t = M[1,k]$, with $s \leq t$ or $t \leq s$. It substitutes $\gcd(s,t) = xs + yt$ for s in $M[1,j]$ and $0 = x_0 s + y_0 t$ for t in $M[1,k]$. Since the reduction works on columns, it also substitutes $xM[2,j] + yM[2,k]$ for $M[2,j]$ and $x_0 M[2,j] + y_0 M[2,k]$ for $M[2,k]$. We have $|x|, |y|, |x_0|, |y_0| \leq N$, so the first gcd computation in row 1 increases the magnitude of row 2 from at most N to at most $2N^2$. Similarly, every additional gcd computation in row 1, increases the magnitude of row 2 by at most a factor $2N$, so after $c - 1$ such computations, we have $\mu_2 \leq N(2N)^{c-1} = \frac{1}{2}(2N)^c$. We iterate this argument and get $\mu_i \leq \frac{1}{2}(2\mu_{i-1})^c \leq \frac{1}{2}(2N)^{c^{i-1}}$ for $2 \leq i \leq d$. Taking the binary logarithm of the final and largest upper bound, we get

$$\log_2 \mu_d \leq \log_2 \left(\tfrac{1}{2}(2N)^{c^{d-1}} \right) \leq c^{d-1} \log_2(2N). \tag{6.6}$$

Since Euclid's algorithm takes at most logarithmically many steps to compute the gcd of two integers, this is an upper bound on the number of iterations of the inner `while`-loop. Multiplying with $d(c-1)$, we get $dc^d \log_2(2N)$ as an upper bound on the total number of iterations. Each iteration takes $O(d)$ time to subtract an integer multiple of a column from another, or to exchange two columns, which implies the claimed time bound. □

Remark on magnitude. While the above bound on the size of temporary entries in the matrix is rather high, it is known that the magnitude of the matrix in Hermite normal form is bounded from above by $(\sqrt{d}N)^d$ [Sto00, pages 91, 92]. Since the matrix in this form is unique [Mad00, Theorem 1.5.2], it follows that this upper bound also applies to the reduced matrix computed by the above algorithm.

6.2.5 Magnitude and Time for Construction

The size of the entries in the matrices plays an important role in the running time for constructing a periodic merge tree. We thus express N in terms of the size of K/Λ.

Considering a (directed) edge of the quotient complex, a, we recall that $\mathrm{Shift}(a)$ records the relative position of the endpoints of a shadow. The magnitude of this vector is its largest absolute entry, and we write D for the maximum magnitude over all shift vectors of edges in K/Λ, noting that it depends on the chosen basis of Λ.

It is not difficult to see that there are cases in which an unfortunate choice of basis can cause an arbitrarily large D, even independent of the number of vertices and edges in K/Λ. On the other hand, if K is a Delaunay triangulation and the basis vectors are pairwise orthogonal, then $D \leq 1$. This follows from [DH97, Theorem 3.1] but can also be proved directly, as illustrated in Figure 6.5. Let x and y be the endpoints of an edge in the Delaunay triangulation, and u_i a basis vector. The four points, $x, x+u_i, y, y-u_i$ are the vertices of a parallelogram, and the angles at $x+u_i$ and $y-u_i$ are equal and at most $90°$. Indeed, if they exceed $90°$, then every $(d-1)$-sphere that passes through x and y encloses at least one of $x+u_i$ and $y-u_i$, which contradicts that x and y are the endpoints of an edge in the Delaunay triangulation. Since this holds for all basis vectors and their negatives, $\pm u_i$ for $1 \leq i \leq d$, we conclude that y is contained in the hyper-rectangle with vertices $x \pm u_1 \pm u_2 \pm \ldots \pm u_d$. But this hyper-rectangle overlaps only the unit cell that contains x and the $3^d - 1$ neighboring unit cells. It follows that x and y lie in neighboring unit cells, so $D \leq 1$, as claimed.

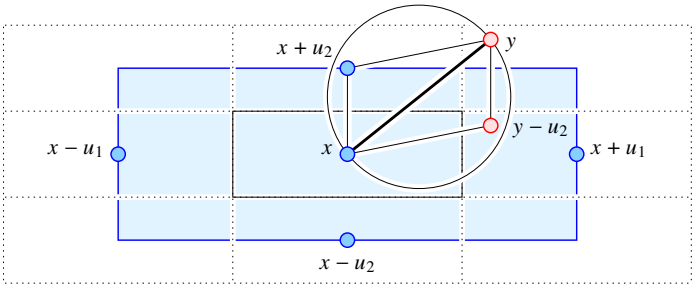

Figure 6.5: The edge connecting x to y cannot be in the Delaunay triangulation if y lies outside the hyper-rectangle whose facets are centered at the points $x \pm u_i$. This hyper-rectangle has the volume of 2^d unit cells and overlaps 3^d of them, which for the displayed 2-dimensional case are drawn with *dotted lines*.

Lemma 6.2.3 (Magnitude of Basis)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice with given matrix of basis vectors, U, let $F\colon K \to \mathbb{R}$ be a Λ-periodic filter, let m be the number of edges in K/Λ, and let D the be maximum magnitude of their shift vectors. For every $\Gamma \in \mathbb{M}(F,\Lambda)$, let V be the Hermite normal form of the lattice $U^{-1}(\Lambda_\Gamma) \subseteq \mathbb{Z}^d$. Then, the basis, UV, of the periodicity lattice of Γ satisfies $\|V\|_\infty \leq (\sqrt{d}Dm)^d$.*

Proof. Take a spanning tree of Γ, and for each edge not in this tree, take the drift vector of the simple loop for which this edge is the sole edge not in the tree. There are at most m edges in this loop, so its drift vector has magnitude at most Dm. We get fewer than m such vectors, which define the lattice of the periodicity lattice of Γ. Letting M be the matrix whose columns are these drift vectors, we compute a basis by reducing M to Hermite normal form. Since $\|M_\infty\| \leq Dm$, this implies that the magnitude of the reduced matrix is at most $(\sqrt{d}Dm)^d$. □

The overall algorithm constructs a periodic merge tree by incrementally adding the vertices and edges of K/Λ. Whenever an edge connects two components (a merge event) or forms a loop (a possible catenation), we use the reduction algorithm to compute a basis from two

bases. By Lemma 6.2.3, the magnitude of the input matrices is at most $(\sqrt{d}Dm)^d$. Indeed, as the Hermite normal form of a lattice is unique, the iteratively computed Hermite normal form from our algorithm agrees with the Hermite normal form from Lemma 6.2.3.

Theorem 6.2.4 (Time for Construction)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice with given basis, $F\colon K \to \mathbb{R}$ a Λ-periodic filter, n and m the number of vertices and edges in the quotient complex, K/Λ, and D the maximum magnitude of the shift vectors. Then the periodic merge tree of F can be constructed in time $O(n \log n + m 2^d d^{d+3} \log(\sqrt{d}Dm))$.*

Proof. The union-find data structure is maintained in constant time per edge and amortized logarithmic time per vertex, so $O(n \log n + m)$ time in total. The reduction algorithm is used at most once for each edge, and each time reduces an integer matrix with d rows and at most $2d$ columns. By Lemma 6.2.3, its magnitude is at most $(\sqrt{d}Dm)^d$, and by Lemma 6.2.2, the time to reduce the matrix is at most $O(d^2(2d)^d \log(2(\sqrt{d}Dm)^d))$. Assuming $(\sqrt{d}Dm)^d \geq 2$, we can simplify the bound to $O(d^3(2d)^d \log(\sqrt{d}Dm))$. After multiplying with the number of edges, we get $O(md^3(2d)^d \log(\sqrt{d}Dm))$ time in total, which implies the claimed bound. □

In the important application of our algorithm to crystalline materials, the dimension is a constant, $d = 3$. Using the filtration of alpha complexes in the torus, it is typical that each vertex (atom center) is connected to at most some constant number of other vertices, so the number of edges is $m = O(n)$. Similarly typical is that entries in the shift vectors are bounded by a small constant, D. In this case, the algorithm constructs the periodic merge tree in time at most $O(n \log n)$.

6.3 Properties

This section studies the invariance under different choices of lattices and the stability with respect to perturbations. We begin with the introduction of a notion of distance between periodic merge trees, and follow up with an equivalence relation among periodic merge trees and the corresponding quotient pseudo-metric.

6.3.1 Interleaving Distance

We adapt the interleaving distance between ordinary merge trees introduced in [MBW13] to periodic merge trees by including the frequency function, which maps each point of the tree to its shadow monomial. Recall that a point $\mathrm{B} \in \mathbb{M}$ covers $\mathrm{A} \in \mathbb{M}$ if $\mathtt{h}(\mathrm{A}) \leq \mathtt{h}(\mathrm{B})$ and $\mathrm{A} \subseteq \mathrm{B}$.

Definition 6.3.1 (Interleaving Distance)**.** Let $\mathbb{M}$ and $\mathbb{M}'$ be two periodic merge trees, with height functions $\mathtt{h}\colon \mathbb{M} \to \mathbb{R}$, $\mathtt{h}'\colon \mathbb{M}' \to \mathbb{R}$ and frequency functions $\mathtt{f}\colon \mathbb{M} \to \mathcal{M}[R]$, $\mathtt{f}'\colon \mathbb{M}' \to \mathcal{M}[R]$. For $\varepsilon \geq 0$, continuous maps, $\varphi\colon \mathbb{M} \to \mathbb{M}'$ and $\psi\colon \mathbb{M}' \to \mathbb{M}$ are *ε-compatible* if for all $\Gamma \in \mathbb{M}$, $\Gamma' \in \mathbb{M}'$,

(i) $\mathtt{h}' \circ \varphi(\Gamma) = \mathtt{h}(\Gamma) + \varepsilon$ and $\mathtt{h} \circ \psi(\Gamma') = \mathtt{h}'(\Gamma') + \varepsilon$;

(ii) $\psi \circ \varphi(\Gamma)$ covers Γ and $\varphi \circ \psi(\Gamma')$ covers Γ';

(iii) $\mathtt{f}' \circ \varphi(\Gamma) \leq \mathtt{f}(\Gamma)$ and $\mathtt{f} \circ \psi(\Gamma') \leq \mathtt{f}'(\Gamma')$.

The *interleaving distance* between $\mathbb{M}$ and $\mathbb{M}'$, denoted $I(\mathbb{M}, \mathbb{M}')$, is the infimum of the $\varepsilon \geq 0$ for which there exist ε-compatible maps from $\mathbb{M}$ to $\mathbb{M}'$ and back.

We focus on *finite* periodic merge trees, for which the infimum in Definition 6.3.1 is a minimum. Throughout this section, we will therefore assume finite trees, with the benefit of simpler arguments involving the interleaving distance.

As an example, consider the two periodic merge trees in Figure 6.4. Writing $\mathbb{M}$ for the tree in the left panel and $\mathbb{M}'$ for the tree in the right panel, we need continuous maps φ and ψ whose compositions increase the height of each point by 2ε in its own tree (i.e. move the point to the right in our drawing). For small ε, this is not possible because $\mathbb{M}'$ has two subtrees rooted at the point with height 7.0, while $\mathbb{M}$ has only one such subtree. The minimum height of any point is 1.0, and since ε is at least half the height difference, we get $\varepsilon \geq 3.0$. We get another constraint from Condition (iii): the shadow monomial cannot increase from a point to its image. All points with height less than 7.0 in $\mathbb{M}$ have shadow monomials larger than those of the points with height 1.0 in $\mathbb{M}'$, hence $\varepsilon \geq 6.0$. Indeed, for $\varepsilon = 6.0$ we get maps φ and ψ that are ε-compatible, which implies $I(\mathbb{M}, \mathbb{M}') = 6.0$.

We extend the proof of the interleaving distance being a metric for merge trees [MBW13] to periodic merge trees, but note that [MBW13] missed the verification of positivity, which we add in our extension.

Lemma 6.3.2 (Interleaving Distance is a Metric)**.** *Let $\mathbb{M}, \mathbb{M}', \mathbb{M}''$ be periodic merge trees. Then*

- $I(\mathbb{M}, \mathbb{M}') \geq 0$ *and* $I(\mathbb{M}, \mathbb{M}') > 0$ *if and only if* $\mathbb{M} \neq \mathbb{M}'$ *(positivity);*
- $I(\mathbb{M}, \mathbb{M}') = I(\mathbb{M}', \mathbb{M})$ *(symmetry);*
- $I(\mathbb{M}, \mathbb{M}') + I(\mathbb{M}', \mathbb{M}'') \geq I(\mathbb{M}, \mathbb{M}'')$ *(triangle inequality).*

Proof. Positivity: Clearly, $I(\mathbb{M}, \mathbb{M}') = 0$ if $\mathbb{M} = \mathbb{M}'$. It remains to show $I(\mathbb{M}, \mathbb{M}') > 0$ if $\mathbb{M} \neq \mathbb{M}'$. Recall that a complex in the d-dimensional torus is necessarily finite, so there are only finitely many critical events in the construction of the corresponding periodic merge tree. Let $t_0 < t_1 < \ldots < t_k$ be the values of the critical events that arise in the construction of $\mathbb{M}$ and $\mathbb{M}'$. They decompose both trees into a finite collection of intervals, each connecting points at consecutive critical values. There is an adjacency preserving bijection between the two collections of intervals that preserves heights and shadow monomials if and only if $\mathbb{M} = \mathbb{M}'$. But if there is no such bijection, then there are also no ε-compatible maps for ε smaller than the minimum difference between consecutive critical values. Hence, $I(\mathbb{M}, \mathbb{M}') > 0$.

Symmetry: Since we can switch φ and ψ in Definition 6.3.1, we have $I(\mathbb{M}, \mathbb{M}') \leq \varepsilon$ if and only if $I(\mathbb{M}', \mathbb{M}) \leq \varepsilon$, for every $\varepsilon \geq 0$. Hence, $I(\mathbb{M}, \mathbb{M}') = I(\mathbb{M}', \mathbb{M})$.

Triangle inequality: Writing $\varepsilon = I(\mathbb{M}, \mathbb{M}')$ and $\varepsilon' = I(\mathbb{M}', \mathbb{M}'')$, there exist ε-compatible maps $\varphi\colon \mathbb{M} \to \mathbb{M}'$, $\psi\colon \mathbb{M}' \to \mathbb{M}$ and ε'-compatible maps $\varphi'\colon \mathbb{M}' \to \mathbb{M}''$, $\psi'\colon \mathbb{M}'' \to \mathbb{M}'$. It is easy to see that $\varphi' \circ \varphi\colon \mathbb{M} \to \mathbb{M}''$ and $\psi \circ \psi'\colon \mathbb{M}'' \to \mathbb{M}$ satisfy (i) and (iii) of Definition 6.3.1 for $\varepsilon + \varepsilon'$. To see (iii), we continuously advance Γ to the point $\Pi \in \mathbb{M}$ at height $\mathrm{h}(\Pi) = \mathrm{h}(\Gamma) + 2\varepsilon'$. By continuity of φ, and because there is only one point of $\mathbb{M}'$ whose height is $\mathrm{h}'(\varphi(\Gamma)) + 2\varepsilon'$ that covers $\varphi(\Gamma)$, this point must be $\varphi(\Pi) = \psi' \circ \varphi' \circ \varphi(\Gamma)$. Hence, $\psi \circ \varphi(\Pi) = \psi \circ \psi' \circ \varphi' \circ \varphi(\Gamma)$. Since $\psi \circ \varphi(\Pi)$ covers Π and therefore Γ, this implies that $\varphi' \circ \varphi$ and $\psi \circ \psi'$ satisfy (ii) for $\mathbb{M}$ and similarly for $\mathbb{M}''$, which implies $I(\mathbb{M}, \mathbb{M}'') \leq \varepsilon + \varepsilon'$. □

6.3.2 Splintering Periodic Merge Trees

While the periodic merge tree is independent of the choice of basis of a given lattice, it heavily depends on the lattice used for the filter; see Figure 6.4, where we get interleaving distance $\varepsilon = 6.0$ just because we switch from the integer lattice on the left to a sublattice that drops every other column of integer points on the right. We desire a notion of distance that tolerates different lattices, provided the filter is periodic with respect to both. To this end, we introduce an equivalence relation and work with the corresponding quotient of the interleaving distance. For each point $\Gamma \in \mathbb{M}$, write $\mathbb{M}_\Gamma$ for the subtree with topmost point Γ; that is: $\mathbb{M}_\Gamma$ consists of all points $A \in \mathbb{M}$ covered by Γ.

Definition 6.3.3 (Equivalence by Splintering)**.** Let $\mathbb{M}$ and $\mathbb{M}'$ be two periodic merge trees with height functions $\mathtt{h}\colon \mathbb{M} \to \mathbb{R}$, $\mathtt{h}'\colon \mathbb{M}' \to \mathbb{R}$ and frequency functions $\mathtt{f}\colon \mathbb{M} \to \mathcal{M}[R]$, $\mathtt{f}'\colon \mathbb{M}' \to \mathcal{M}[R]$. We say $\mathbb{M}'$ *splinters* $\mathbb{M}$ if there exists a continuous surjection $\omega\colon \mathbb{M}' \to \mathbb{M}$ such that

(i) the surjection preserves height: $\mathtt{h} \circ \omega = \mathtt{h}'$;

(ii) it splits subtrees evenly: for every $\Gamma \in \mathbb{M}$ and all $A, B \in \omega^{-1}(\Gamma)$, the subtrees at A and B are equal, $I(\mathbb{M}'_A, \mathbb{M}'_B) = 0$, ω maps $\mathbb{M}'_A$ as well as $\mathbb{M}'_B$ surjectively onto $\mathbb{M}_\Gamma$, and the shadow monomials are $\mathtt{f}'(A) = \mathtt{f}'(B) = \mathtt{f}(\Gamma)/\#\omega^{-1}(\Gamma)$.

We call two periodic merge trees *equivalent*, denoted $\mathbb{M} \simeq \mathbb{N}$, if there exists a periodic merge tree that splinters both, and we write $[\mathbb{M}]$ for the class of periodic merge trees equivalent to $\mathbb{M}$.

To be certain that "$\simeq$" is indeed an equivalence relation, we need to verify that it is transitive. First note that splintering is transitive: if $\mathbb{M}'$ splinters $\mathbb{M}$ and $\mathbb{M}''$ splinters $\mathbb{M}'$, then $\mathbb{M}''$ also splinters $\mathbb{M}$. Second observe that two splinterings of the same tree can be composed: if $\mathbb{M}'$ and $\mathbb{N}'$ both splinter $\mathbb{N}$, then there exists $\mathbb{M}''$ that splinters both $\mathbb{M}'$ and $\mathbb{N}'$. This operation acts like point-wise multiplication: if $\Gamma \in \mathbb{N}$ has i preimages in $\mathbb{M}'$ and j preimages in $\mathbb{N}'$, then it has ij preimages in $\mathbb{M}''$. To see that this is well defined, we traverse the points of the two trees in parallel and in the order of decreasing height, making sure that the split is even whenever we encounter a point of bifurcation. With this we argue the transitivity of "$\simeq$": assuming $\mathbb{M} \simeq \mathbb{N} \simeq \mathbb{O}$, there exist trees $\mathbb{M}'$ and $\mathbb{N}'$ splintering $\mathbb{M}, \mathbb{N}$ and $\mathbb{N}, \mathbb{O}$, respectively. Since $\mathbb{M}'$ and $\mathbb{N}'$ both splinter $\mathbb{N}$, there exists $\mathbb{M}''$ that splinters $\mathbb{M}', \mathbb{N}'$. By transitivity of splintering, $\mathbb{M}''$ also splinters $\mathbb{M}$ as well as $\mathbb{O}$, so $\mathbb{M} \simeq \mathbb{O}$ as required.

Splintering happens, for example, when we construct the periodic merge tree of a Λ-periodic filter for a sublattice of Λ; see Figure 6.4. By combining Definitions 6.3.1 and 6.3.3, we obtain a pseudo-distance function that considers equivalent such trees the same. For this, we use the quotient pseudo-metric of a metric space with respect to an equivalence relation [BBI01, Definition 3.1.12]. The proof that this yields a pseudo-metric is both standard and trivial.

Definition 6.3.4 (Interleaving Pseudo-distance)**.** Let $[\mathbb{M}]$ and $[\mathbb{M}']$ be two equivalence classes of periodic merge trees. The *interleaving pseudo-distance* between them is the infimum, over all sequences of equivalence classes and two trees per class, of the sum of interleaving distances:

$$J([\mathbb{M}], [\mathbb{M}']) = \inf \left\{ \sum_{i=1}^{k} I(\mathbb{M}_i, \mathbb{M}'_i) \right\}, \tag{6.7}$$

in which $\mathbb{M}_1 \in [\mathbb{M}]$, $\mathbb{M}'_k \in [\mathbb{M}']$, and $\mathbb{M}'_i \simeq \mathbb{M}_{i+1}$ for all $1 \leq i \leq k-1$.

As an example, consider again the two periodic merge trees in Figure 6.4. Writing $\mathbb{M}$ and $\mathbb{M}'$ for the tree in the left and right panels, respectively, we recall that the interleaving distance is $I(\mathbb{M}, \mathbb{M}') = 6.0$. The interleaving pseudo-distance is however zero, because $\mathbb{M}'$ splinters $\mathbb{M}$ and thus they are in the same equivalence class. Indeed, we get $\mathbb{M}'$ by duplicating the subtree of $\mathbb{M}$ below the point at height 7.0 and assigning half the shadow monomial to each point of the two copies of the subtree.

6.3.3 Invariance of Equivalence Classes

As mentioned earlier, splintering occurs if we enlarge the unit cell by substituting a sublattice for the original lattice in the construction of the periodic merge tree. We formalize this claim and prove it.

Lemma 6.3.5 (Splintering from Sublattice)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, $F\colon K \to \mathbb{R}$ a Λ-periodic filter, and $\Lambda' \subseteq \Lambda$ a d-dimensional sublattice. Then $\mathbb{M}(F, \Lambda')$ splinters $\mathbb{M}(F, \Lambda)$.*

Proof. To show that $\mathbb{M}' = \mathbb{M}(F, \Lambda')$ splinters $\mathbb{M} = \mathbb{M}(F, \Lambda)$, we study the projection $\phi\colon \mathbb{R}^d/\Lambda' \to \mathbb{R}^d/\Lambda$ defined by mapping every point $x+\Lambda'$ to $x+\Lambda$. Since $\dim \Lambda' = \dim \Lambda$, the degree of the map is finite, namely $\deg \phi = \mathrm{vol}_d(\Lambda')/\mathrm{vol}_d(\Lambda)$, and the image of ϕ covers every point in the d-dimensional torus $\deg \phi$ times. The projection induces a surjection, $\omega\colon \mathbb{M}' \to \mathbb{M}$, that maps every point $\Gamma' \in \mathbb{M}'$ to the point $\Gamma = \omega(\Gamma') \in \mathbb{M}$ with $\mathrm{h}(\Gamma) = \mathrm{h}'(\Gamma')$ and $\phi(\Gamma') = \Gamma$, where we note that Γ and Γ' are connected components of sublevel sets. The continuity of ω follows from the continuity of ϕ and the definition of quotient topology as the final topology with respect to the quotient map. By construction, ω preserves height, so it satisfies Condition (i) of Definition 6.3.3.

To prove Condition (ii), we note that all preimages of Γ are translates of each other and thus have the same periodicity lattice and shadow monomials. Similarly, their subtrees are identical. To show that ω maps subtrees surjectively to the appropriate subtrees, recall that $\phi(\Gamma') = \Gamma$. Hence, for every component A in the subtree at Γ, we have $\phi|_{\Gamma'}^{-1}(\mathrm{A}) \neq \emptyset$, and hence at least one component at height $\mathrm{h}(\mathrm{A})$ in the subtree at Γ' that ω maps to A. The only part of Condition (ii) left to prove is the shadow monomial, $\mathtt{f}(\Gamma')$. To this end, note that the cardinality of the preimage, $\#\omega^{-1}(\Gamma)$, depends on $\deg \phi$ but also on the periodicity lattices of Γ and Γ'. By construction, these two lattices have the same dimension, p. The number of times the projection of Γ' covers Γ is thus finite, namely $\mathrm{vol}_p(\Lambda_{\Gamma'})/\mathrm{vol}_p(\Lambda_\Gamma)$. Therefore, out of the $\deg \phi$ preimages of every point in Γ, $\mathrm{vol}_p(\Lambda_{\Gamma'})/\mathrm{vol}_p(\Lambda_\Gamma)$ stem from the same component, so the number of different components is

$$\#\omega^{-1}(\Gamma) = \deg \phi \cdot \frac{\mathrm{vol}_p(\Lambda_\Gamma)}{\mathrm{vol}_p(\Lambda_{\Gamma'})}. \tag{6.8}$$

This yields the following calculation for the shadow monomial:

$$\mathtt{f}'(\Gamma') = \frac{\mathrm{vol}_p(\Lambda_{\Gamma'})}{\mathrm{vol}_d(\Lambda')} \cdot \nu_{d-p} R^{d-p} \tag{6.9}$$

$$= \frac{\mathrm{vol}_p(\Lambda_\Gamma)}{\mathrm{vol}_d(\Lambda)} \cdot \nu_{d-p} R^{d-p} \Big/ \left[\frac{\mathrm{vol}_d(\Lambda')}{\mathrm{vol}_d(\Lambda)} \cdot \frac{\mathrm{vol}_p(\Lambda_\Gamma)}{\mathrm{vol}_p(\Lambda_{\Gamma'})} \right] = \frac{\mathtt{f}(\Gamma)}{\#\omega^{-1}(\Gamma)}, \tag{6.10}$$

in which the left-hand side of (6.10) is obtained using trivial substitutions, and the right-hand side follows because the expression in squared brackets is $1/\#\omega^{-1}(\Gamma)$. We conclude that ω satisfies all conditions of Definition 6.3.3. □

This implies the first main property of periodic merge trees we set out to prove in this section: that their equivalence classes are invariant under different lattices and different bases of the same lattice used in the construction of the trees.

Theorem 6.3.6 (Invariance of Equivalence Classes of Periodic Merge Trees)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, $F\colon K \to \mathbb{R}$ a Λ-periodic filter, and $\Lambda', \Lambda'' \subseteq \Lambda$ two d-dimensional sublattices. Then $\mathbb{M}(F,\Lambda')$ and $\mathbb{M}(F,\Lambda'')$ are equivalent and thus the interleaving pseudo-distance between their equivalence classes vanishes: $J([\mathbb{M}(F,\Lambda')],[\mathbb{M}(F,\Lambda'')]) = 0$.*

Proof. We show that the two trees are equivalent by finding a common splintering. For this, consider the lattice $\Lambda''' = \Lambda' \cap \Lambda''$. It is a sublattice of both Λ' and Λ''. By Lemma 6.3.5, $\mathbb{M}(F,\Lambda''')$ splinters both, $\mathbb{M}(F,\Lambda')$ and $\mathbb{M}(F,\Lambda'')$, which implies that the interleaving pseudo-distance vanishes. □

6.3.4 Stability of Periodic Merge Trees

The second property addressed in this section is the stability of periodic merge trees under perturbations of the filter. More specifically, we show that the interleaving pseudo-distance between the equivalence classes of two periodic merge trees is bounded from above by the L_∞-distance between the two filters.

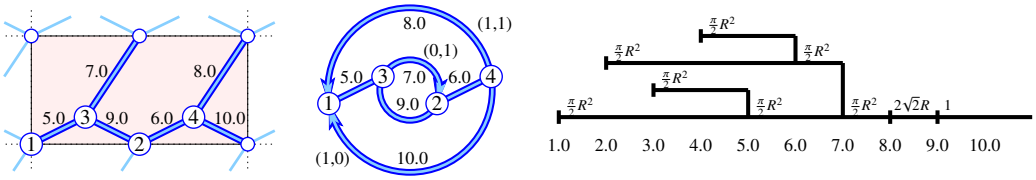

Figure 6.6: *From left to right:* a perturbation of the periodic graph in the left panel of Figure 6.4, the corresponding quotient graph shown with filter values and shift vectors, and the resulting periodic merge tree. Its interleaving pseudo-distance to the periodic merge tree in the left panel of Figure 6.4 is 1.0. This can be seen by first splintering the tree in the left panel to obtain the tree in the right panel of Figure 6.4, and then comparing the latter to the periodic merge tree in this figure. The intermediate step is necessary because after the perturbation, the graph is no longer periodic with respect to the initially used square lattice.

Theorem 6.3.7 (Stability of Periodic Merge Trees)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, and $F, G\colon K \to \mathbb{R}$ two Λ-periodic filters. Then $I(\mathbb{M}(F,\Lambda),\mathbb{M}(G,\Lambda)) \leq \|F-G\|_\infty$ and hence $J([\mathbb{M}(F,\Lambda)],[\mathbb{M}(G,\Lambda)]) \leq \|F-G\|_\infty$.*

Proof. Writing $\mathbb{M} = \mathbb{M}(F,\Lambda)$ and $\mathbb{M}' = \mathbb{M}(G,\Lambda)$ for the two periodic merge trees, we prove $I(\mathbb{M},\mathbb{M}') \leq \|F-G\|_\infty$, and by definition of the interleaving pseudo-distance in terms of the interleaving distance, we have $J([\mathbb{M}],[\mathbb{M}']) \leq I(\mathbb{M},\mathbb{M}')$, which then implies the claimed inequality. Write $\varepsilon = \|F-G\|_\infty$, and let A and A' be connected components of the sublevel sets of F/Λ and G/Λ at height $\mathrm{h}(\mathrm{A}) = \mathrm{h}'(\mathrm{A}') = t$, respectively. Since ε is the L_∞-distance between F and G, there are unique connected components B and B' of the sublevel sets of

F/Λ and G/Λ at height $t+\varepsilon$, respectively, such that $\mathrm{A} \subseteq \mathrm{B}'$ and $\mathrm{A}' \subseteq \mathrm{B}$. By the stability theorem for merge trees in [MBW13], the maps $\varphi\colon \mathbb{M} \to \mathbb{M}'$ and $\psi\colon \mathbb{M}' \to \mathbb{M}$ defined by $\varphi(\mathrm{A}) = \mathrm{B}'$ and $\psi(\mathrm{A}') = \mathrm{B}$ are ε-compatible as far as the two merge trees are concerned; that is: they satisfy Conditions (i) and (ii) of Definition 6.3.1.

It remains to prove Condition (iii). Since $\mathrm{A} \subseteq \mathrm{B}'$, an argument analogous to the proof of Lemma 6.1.5 yields $\mathtt{f}'(\mathrm{B}') \leq \mathtt{f}(\mathrm{A})$ and, by symmetry, $\mathtt{f}(\mathrm{B}) \leq \mathtt{f}'(\mathrm{A}')$. Therefore, φ and ψ are ε-compatible with respect to the two periodic merge trees, so $I(\mathbb{M}, \mathbb{M}') \leq \varepsilon$ by definition of the interleaving distance between these trees. □

See Figure 6.6 for an illustration of the stability in which the perturbation forces a coarser lattice and thus an intermediate step of splintering. With the help of Theorem 6.3.6, the statement about the interleaving pseudo-distance in Theorem 6.3.7 can be extended to periodic filters $F\colon K \to \mathbb{R}$ and $G\colon L \to \mathbb{R}$, provided K and L have the same underlying space and a common periodic refinement.

6.4 The Periodic Barcode

Motivated by simplifying the computation of distance, we strip information off the periodic merge tree to construct the periodic analog of the 0-th barcode or persistence diagram. Following the structure of the previous section, we introduce a new notion of distance right after defining the concept and prove invariance and stability thereafter.

6.4.1 Definition by Construction

In a nut-shell, we get the periodic 0-th barcode by calling the beams of the periodic merge tree bars while dropping the vertical segments that connect them. Recall, however, that different epochs along a beam have different shadow monomials, which decrease from left to right, and we need to account for them by possibly substituting more than one bar for each beam. Specifically, when the monomial decreases from $s\nu_{d-p}R^{d-p}$ to $t\nu_{d-q}R^{d-q}$ at some interior point of a beam, this point marks the end of an era, if $d-q < d-p$, and the end of an epoch inside an era, if $d-q = d-p$. This change is brought about by $s\nu_{d-p}R^{d-p} - t\nu_{d-q}R^{d-q}$ deaths giving rise to polynomial decorations of bars; see the step from the top left panel to the bottom left panel in Figure 6.7.

While the auction algorithm [KMN17] for computing Wasserstein distances between persistence diagrams can deal with polynomial multiplicities, some vectorization methods cannot (e.g. persistence images [AEK+17]). We therefore choose to decompose further. We replace a bar with decorating polynomial $s\nu_{d-p}R^{d-p} - t\nu_{d-q}R^{d-q}$ by two bars with the same birth and death, but decorated by the monomials $s\nu_{d-p}R^{d-p}$ and $-t\nu_{d-q}R^{d-q}$. These monomials can then be viewed as real valued multiplicities (given by their coefficients) in $d+1$ different barcodes (separated by their exponents). This enables the classification of the bars into $d+1$ eras, which is desirable since different growth-rates render the coefficients of the corresponding monomials incommensurable. We thus obtain the 0-th barcode as an ensemble of $d+1$ collections of bars, one for each era; see the middle panels in Figure 6.7. The alternative visualization as an ensemble of $d+1$ points is displayed in the right panels of the same figure. Within each era we, drop the factor $\nu_{d-p}R^{d-p}$ of the decoration that all bars share.

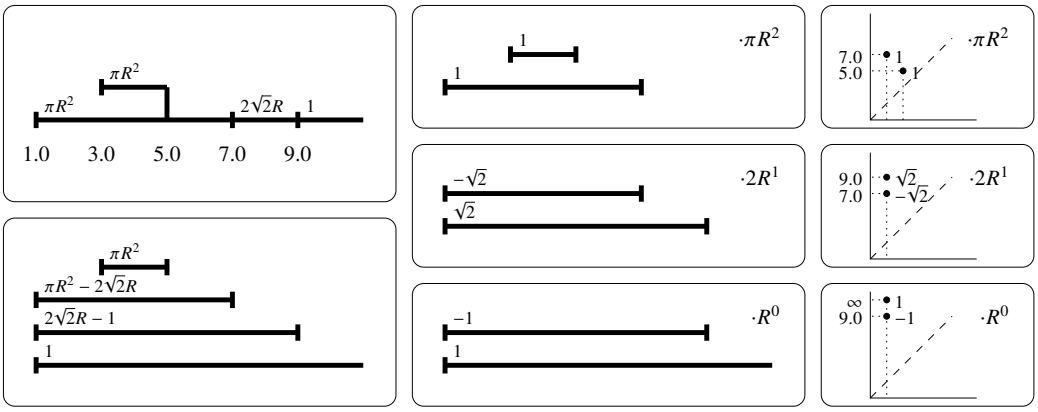

Figure 6.7: *Left:* the periodic merge tree of the left panel in Figure 6.4 and below its decomposition into labeled bars. *Middle:* The periodic barcode split into groups of bars per era. *Right:* the corresponding ensemble of periodic persistence diagrams.

With this in mind, we compute the periodic barcode from the periodic merge tree one epoch at a time. In particular, consider a beam with left endpoint $\Gamma \in \mathbb{M}$, and an epoch on this beam that starts at $\mathrm{A} \in \mathbb{M}$ and ends entering $\mathrm{B} \in \mathbb{M}$. Letting $\mathtt{f}(\mathrm{A}) = sv_{d-p}R^{d-p}$ be the monomial decorating this epoch, we introduce two bars into the $(d-p)$*-ary* 0*-th barcode*: $[\mathtt{h}(\Gamma), \mathtt{h}(\mathrm{B}))$ with *multiplicity* s and $[\mathtt{h}(\Gamma), \mathtt{h}(\mathrm{A}))$ with *multiplicity* $-s$, but note that the second bar might be empty, in which case we skip it. The first bar tells us that $sv_{d-p}R^{d-p}$ (minus the monomial of the next epoch) of the shadows born at $\mathtt{h}(\Gamma)$ die at $\mathtt{h}(\mathrm{B})$, and the second bar tells us that the monomial of the previous epoch minus $sv_{d-p}R^{d-p}$ of the shadows born at $\mathtt{h}(\Gamma)$ die at $\mathtt{h}(\mathrm{A})$. This straightforward conversion takes time proportional to the number of epochs and gives a collection of bars with possibly negative multiplicities. We refer to the collection of the $d+1$ different $(d-p)$-ary 0-th barcodes as the *periodic* 0*-th barcode*, or the *periodic* 0*-th persistence diagram*, if we choose to visualize the information with points rather than bars. Similar to the notation for the periodic merge tree, we write $\mathbb{B}_0(F, \Lambda)$ for the periodic 0-th barcode of F and Λ.

Setting $\bar{\mathbb{R}} = (-\infty, \infty]$, we note that for each era, the corresponding bars with multiplicities can be viewed as a function $\mathbb{R} \times \bar{\mathbb{R}} \to \mathbb{R}$, which maps a point (b, d) to the (possibly negative) sum of the multiplicities of all bars $[b, d)$. This sum is 0 if there are no such bars. While some vectorization methods for barcodes require the multiplicities to be non-negative integers, some other standard vectorization methods, such as persistence images [AEK+17], do not need such requirements and can thus be used for periodic barcodes.

6.4.2 Alternating Wasserstein Distance

While the interleaving distance between periodic merge trees is difficult to find, it is straightforward to compute a meaningful distance between periodic 0-the barcodes, as we will shortly see. We begin by formulating the Wasserstein distance between two periodic barcodes as a solution to an optimal transport problem. The bars in classic barcodes come with positive integer multiplicities, for which the Wasserstein distance is a solution to an optimal assignment problem; see [EH10, Section VIII.4] for an algorithm. Contrast this to the periodic setting, in which we have real and possibly negative multiplicities, which requires a solution to the more general optimal transport problem; see [PC19] for comprehensive background. We will work with the linear version, commonly referred to as the 1*-Wasserstein distance*, which is a weighted sum of distances between points. The main reason is that

this version satisfies $W_1(\xi, \eta) = W_1(\xi + \zeta, \eta + \zeta)$, and a similar relation does not hold if we work with powers $q > 1$ of the distances. This property allows for the removal of negative multiplicities by adding them on both sides, which is also the intuition behind the definition of the alternating 1-Wasserstein distance in (6.13). Recall that the *support* a function $f\colon \mathbb{R} \times \bar{\mathbb{R}} \to \mathbb{R}$, denoted $\operatorname{supp}(f)$, are the points in the domain for which f is non-zero.

Definition 6.4.1 (Alternating Wasserstein Distance)**.** Let $\xi, \eta\colon \mathbb{R} \times \bar{\mathbb{R}} \to \mathbb{R}$ be two non-negative multiplicity functions, each with finite support. Then the 1-*Wasserstein distance* between ξ and η is

$$W_1(\xi, \eta) = \inf_{T,X,Y} \left[\sum_{x,y} T(x, y)\|x - y\|_1 + \sum_x X(x)\delta(x) + \sum_y Y(y)\delta(y) \right], \tag{6.11}$$

in which $\delta(x) = |x_2 - x_1|$ is the vertical distance of $x = (x_1, x_2)$ to the diagonal, and the infimum is taken over all functions $T\colon \operatorname{supp}(\xi) \times \operatorname{supp}(\eta) \to [0, \infty)$, $X\colon \operatorname{supp}(\xi) \to [0, \infty)$, and $Y\colon \operatorname{supp}(\eta) \to [0, \infty)$ that satisfy

$$X(x) + \sum_{y \in \operatorname{supp}(\eta)} T(x, y) = \xi(x) \quad \text{and} \quad Y(y) + \sum_{x \in \operatorname{supp}(\xi)} T(x, y) = \eta(y), \tag{6.12}$$

for all $x \in \operatorname{supp}(\xi)$ and $y \in \operatorname{supp}(\eta)$. Without assuming the non-negativity of the multiplicities, we write $\xi = \xi^+ - \xi^-$ and $\eta = \eta^+ - \eta^-$, in which $\xi^+, \xi^-, \eta^+, \eta^-$ are non-negative. The more general *alternating* 1-*Wasserstein distance* between ξ and η is

$$W_1^{\pm}(\xi, \eta) = W_1(\xi^+ + \eta^-, \xi^- + \eta^+). \tag{6.13}$$

Note that the relation $W_1(\xi, \eta) = W_1(\xi + \zeta, \eta + \zeta)$ implies that $W_1^{\pm}(\xi, \eta)$ does not depend on the choice of $\xi^+, \xi^-, \eta^+, \eta^-$. We may therefore assume that ξ^+ and ξ^- have disjoint supports, and similar for η^+ and η^-. We will use this property in the proof of the triangle inequality for the alternating 1-Wasserstein distance.

Write Δ for the diagonal in $\mathbb{R}^2$; that is: the points $x = (x_1, x_2)$ with $x_1 = x_2$. For non-negative multiplicity functions whose support avoids Δ, the 1-Wasserstein distance is an extended metric, meaning it can also take ∞ as a value, namely between persistence pairs with finite and infinite deaths. We use this property to prove that the alternating 1-Wasserstein distance is also an extended metric.

Lemma 6.4.2 ($W_1^{\pm}$ is Extended Metric)**.** *Let $\xi, \eta, \zeta\colon \mathbb{R} \times \bar{\mathbb{R}} \to \mathbb{R}$ be multiplicity functions, each with finite support avoiding Δ. Then*

- $W_1^{\pm}(\xi, \eta) \geq 0$ *and* $W_1^{\pm}(\xi, \eta) > 0$ *if and only if* $\xi \neq \eta$ *(positivity);*
- $W_1^{\pm}(\xi, \eta) = W_1^{\pm}(\eta, \xi)$ *(symmetry);*
- $W_1^{\pm}(\xi, \eta) + W_1^{\pm}(\eta, \zeta) \geq W_1^{\pm}(\xi, \zeta)$ *(triangle inequality).*

Proof. Write $\xi = \xi^+ - \xi^-$, $\eta = \eta^+ - \eta^-$, $\zeta = \zeta^+ - \zeta^-$, in which ξ^+, ξ^- are non-negative with disjoint supports, and similar for η^+, η^- and ζ^+, ζ^-. To show positivity, we note that the right-hand side of (6.13) can only vanish when $\xi^+ + \eta^- = \xi^- + \eta^+$. Since $\operatorname{supp}(\xi^+) \cap \operatorname{supp}(\xi^-) = \operatorname{supp}(\eta^+) \cap \operatorname{supp}(\eta^-) = \emptyset$, this implies $\xi^+ = \eta^+$ and $\xi^- = \eta^-$ and therefore also $\xi = \eta$.

Symmetry is implied by the symmetry of Definition 6.4.1. Finally, we use the triangle inequality for positive multiplicities to prove the same for possibly negative multiplicities:

$$\begin{aligned} W_1^{\pm}(\xi,\eta) + W_1^{\pm}(\eta,\zeta) &= W_1(\xi^+ + \eta^-, \xi^- + \eta^+) + W_1(\eta^+ + \zeta^-, \eta^- + \zeta^+) && (6.14)\\ &= W_1(\xi^+ + \eta^- + \zeta^-, \xi^- + \eta^+ + \zeta^-) && (6.15)\\ &\quad + W_1(\xi^- + \eta^+ + \zeta^-, \xi^- + \eta^- + \zeta^+) && (6.16)\\ &\geq W_1(\xi^+ + \eta^- + \zeta^-, \xi^- + \eta^- + \zeta^+) && (6.17)\\ &= W_1(\xi^+ + \zeta^-, \xi^- + \zeta^+), && (6.18) \end{aligned}$$

in which the last term is equal to $W_1^{\pm}(\xi,\zeta)$. In summary, the alternating Wasserstein distance satisfies positivity, symmetry, and the triangle inequality and is therefore an extended metric. □

We define the alternating Wasserstein distance of two periodic 0-th barcodes as the sum over the alternating Wasserstein distances of the $(d-p)$-ary barcodes.

6.4.3 Invariance of Periodic Barcodes

In this subsection, we prove that periodic barcodes do not depend on the lattice used in their construction. This is in contrast to periodic merge trees, for which the splintering relation is needed to get invariant equivalence classes of such trees. We begin by showing that splintering does not affect the periodic barcode derived from a periodic merge tree.

Lemma 6.4.3 (Splintering Preserves Periodic Barcodes)**.** *Let $\mathbb{B}$ and $\mathbb{B}'$ be the periodic 0-th barcodes derived from the periodic merge trees $\mathbb{M}$ and $\mathbb{M}'$. If $\mathbb{M}'$ splinters $\mathbb{M}$, then $\mathbb{B}' = \mathbb{B}$.*

Proof. To get $\mathbb{B}$, we collect two bars from each epoch of $\mathbb{M}$, one with positive and the other with negative multiplicity, but note that the second bar may be empty. Let $\mathrm{A}, \mathrm{B} \in \mathbb{M}$ delimit an epoch decorated with $s\nu_{d-p}R^{d-p}$, and let $\Gamma \in \mathbb{M}$ be the leftmost point on the same beam. By the elder rule applied in the construction of the merge tree, $\mathrm{h}(\Gamma)$ is the minimum height of any point in the subtree rooted at A. The two bars are $[\mathrm{h}(\Gamma), \mathrm{h}(\mathrm{B}))$ with multiplicity s and $[\mathrm{h}(\Gamma), \mathrm{h}(\mathrm{A}))$ with multiplicity $-s$. If we subdivide the epoch into intervals we call *ages* and construct the bars from each age, we get the same result because all bars ending at points strictly between A and B cancel in pairs. This motivates us to refine the epochs of $\mathbb{M}$ and $\mathbb{M}'$ into ages such that

(i) none of the ages contains the value of critical event in its interior;

(ii) the preimages of an age in $\mathbb{M}$ are ages in $\mathbb{M}'$.

Consider an age with shadow monomial $s\nu_{d-p}R^{d-p}$ in $\mathbb{M}$, let N be the number of its preimages in $\mathbb{M}'$, and observe that each of these preimages is an age with shadow monomial $\frac{s}{N}\nu_{d-p}R^{d-p}$. The bars generated by the age in $\mathbb{M}$ and its preimages in $\mathbb{M}$ are the same, except that the former have multiplicities $\pm s$, while the latter have multiplicities $\pm\frac{s}{N}$. Adding the contributions of the N preimages amounts to the same two bars with the same multiplicities. Since this is true for all ages, we conclude that $\mathbb{B} = \mathbb{B}'$. □

The invariance of periodic barcodes is an immediate consequence of Lemma 6.4.3 and Lemma 6.3.5. Indeed, if $\Lambda', \Lambda'' \subseteq \Lambda$ are two d-dimensional sublattices, and $\mathbb{M}(F, \Lambda')$, $\mathbb{M}(F, \Lambda'')$ are the corresponding periodic merge trees, then they both splinter $\mathbb{M}(F, \Lambda)$ by Lemma 6.3.5. By Lemma 6.4.3, the periodic barcodes derived from these periodic merge trees satisfy $\mathbb{B}_0(F, \Lambda') = \mathbb{B}_0(F, \Lambda) = \mathbb{B}_0(F, \Lambda'')$. We formulate for later reference:

Corollary 6.4.4 (Invariance of Periodic Barcodes)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, $F\colon K \to \mathbb{R}$ a Λ-periodic filter, and $\Lambda', \Lambda'' \subseteq \Lambda$ two d-dimensional sublattices. Then $\mathbb{B}_0(F, \Lambda') = \mathbb{B}_0(F, \Lambda'')$.*

6.4.4 Size of Multiplicities

To prepare the proof of stability for periodic barcodes, this section studies how large the multiplicities can get. We thus consider the maximum multiplicity of a bar in the 0-barcode of F and Λ,

$$\mu(F, \Lambda) = \max \left\{ \frac{\mathrm{vol}_p\,(\Lambda_\Gamma)}{\mathrm{vol}_d\,(\Lambda)} \mid \Gamma \in \mathbb{M}(F, \Lambda) \right\}, \tag{6.19}$$

and prove an upper bound that depends on the dimension, d, the number of edges in the quotient complex, m, the magnitude of the shift vectors, D, but also on the basis of Λ. As before, we write U for the matrix whose columns are the vectors in the basis of Λ. Recall that U^{-1} maps Λ to $\mathbb{Z}^d$, and that the *operator norm*, $\|U^{-1}\|_{\mathrm{op}} = \max_{x \in \mathbb{R}^d} \|U^{-1}(x)\|_2 / \|x\|_2$, is the largest singular value of U^{-1} and quantifies the maximum stretching experienced by any vector $x \in \mathbb{R}^d$.

Lemma 6.4.5 (Maximum Multiplicity)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, U the matrix of basis vectors, $F\colon K \to \mathbb{R}$ a Λ-periodic filter on a Λ-periodic complex, m the number of edges, and D the maximum magnitude of their shift vectors in K/Λ. Then $\mu(F, \Lambda) \leq (d^{2.5} D m \|U^{-1}\|_{\mathrm{op}})^d$.*

Proof. Write $\mathbb{M} = \mathbb{M}(F, \Lambda)$, let $\Gamma \in \mathbb{M}$ be arbitrary, set $p = \dim \Lambda_\Gamma$, and let $w_1, w_2, \ldots, w_p$ be a basis of Λ_Γ. Write $v_i = U^{-1}(w_i)$. There are different ways to choose the basis. For example, the columns of the reduced matrix in Hermite normal form would be a valid choice. However, to get a better bound, we choose v_i as the output of a different reduction algorithm, described in [KR23], which we apply to the matrix of size $d \times m$ whose columns are the drift vectors of the simple cycles that correspond to the fewer than m catenation events in the subtree rooted at Γ. As argued in the proof of Lemma 6.2.3, the magnitude of this matrix is at most Dm, and according to [KR23], the magnitude of the reduced matrix—which is not necessarily in Hermite normal form—is at most $d^2 Dm$. It follows that the Euclidean norm of each column vector satisfies $\|v_i\|_2 \leq \sqrt{d}(d^2 Dm) = d^{2.5} Dm$. In order to write $\mathrm{vol}_p\,(\Lambda_\Gamma)$ as a d-dimensional volume, let $w_{p+1}, w_{p+2}, \ldots, w_d$ be unit vectors orthogonal to each other and to Λ_Γ, so that $\mathrm{vol}_p\,(\Lambda_\Gamma) = \mathrm{vol}_d\,(\Lambda'_\Gamma)$, with Λ'_Γ spanned by $w_1, w_2, \ldots, w_d$. Since a linear transformation changes the d-dimensional volume of any subset of $\mathbb{R}^d$ in the same way, we get

$$\frac{\mathrm{vol}_d\,(\Lambda'_\Gamma)}{\mathrm{vol}_d\,(\Lambda)} = \frac{\mathrm{vol}_d\,(U^{-1}(\Lambda'_\Gamma))}{\mathrm{vol}_d\,(U^{-1}(\Lambda))} = \mathrm{vol}_d\left(U^{-1}(\Lambda'_\Gamma)\right) \leq \prod_{i=1}^{d} \|v_i\|_2 \tag{6.20}$$

$$\leq \left(d^{2.5} Dm\right)^p \cdot \|U^{-1}\|_{\mathrm{op}}^{d-p} \leq \left(d^{2.5} Dm \|U^{-1}\|_{\mathrm{op}}\right)^d, \tag{6.21}$$

in which we get the third term in (6.20) by recalling that $U^{-1}(\Lambda) = \mathbb{Z}^d$, and we get the first term in (6.21) by using Lemma 6.2.3 for the first p and the operator norm for the last $d - p$ basis vectors. □

Remark about dependence on basis. Both D and $\|U^{-1}\|_{\text{op}}$ depend on the basis we choose for Λ, but by Corollary 6.4.4, the barcode and therefore the multiplicities of its bars do not. To gain a better understanding of the size of the multiplicities, it would therefore we useful to know how small D and $\|U^{-1}\|_{\text{op}}$ can be made by careful choice of a basis. Indeed, the bound in Lemma 6.4.5 seems excessively pessimistic. It would be interesting to find out what the maximum multiplicity is in practice, e.g. on average for the periodic structures collected in popular crystallographic databases.

6.4.5 Stability of Periodic Barcodes

The *cellular ℓ_1-distance* between filters $F/\Lambda, G/\Lambda \colon K/\Lambda \to \mathbb{R}$ is the sum of absolute differences over all simplices, $\|F/\Lambda - G/\Lambda\|_1 = \sum_{\sigma \in K/\Lambda} |F/\Lambda(\sigma) - G/\Lambda(\sigma)|$, but note that this is different from the L_1-distance between the two functions, which is be the integral of their point-wise differences. By comparing the 1-Wasserstein distance of two non-periodic barcodes with the cellular ℓ_1-distance between the corresponding filters, Skraba and Turner [ST20] prove stability with Lipschitz constant 1. Following this approach, we prove stability with respect to the cellular ℓ_1-distance in the periodic setting, but with worse Lipschitz constant. Observe first that the periodic 0-th barcode is invariant under changing how ties in the ordering of the simplices are broken. Indeed, the periodic merge tree is invariant, and the asymmetry introduced by the elder rule has no effect on the periodic barcode computed from the periodic merge tree. This invariance is of course necessary for the stability of periodic barcodes, which we prove next.

Theorem 6.4.6 (Stability of Periodic Barcodes)**.** *Let $\Lambda \subseteq \mathbb{R}^d$ be a d-dimensional lattice, U the matrix of a basis of Λ, and $F, G \colon K \to \mathbb{R}$ two Λ-periodic filters. Then*

$$W_1^{\pm}(\mathbb{B}_0(F, \Lambda), \mathbb{B}_0(G, \Lambda)) \leq 2(d+1)\mu_0 \cdot \|F/\Lambda - G/\Lambda\|_1, \tag{6.22}$$

in which $\mu_0 = (d^{2.5} D m \|U^{-1}\|_{\text{op}})^d$ is the upper bound on the maximum multiplicity from Lemma 6.4.5, with m the number of edges of K/Λ, and D the maximum magnitude of any of their shift vectors.

Proof. The proof is analogous to that of [ST20, Lemma 4.7 and Theorem 4.8], which itself follows the strategy of the proof of [CSEM06, Combinatorial Stability Theorem]. It consists of two steps: the first assumes that F and G are compatible to a common ordering of the simplices, and the second analyzes the transpositions necessary to make them compatible to such an ordering, if they are not.

Step 1. Assume the filters F and G allow for a total order of the simplices that is compatible to both; that is: $\sigma_i \subseteq \sigma_j$, $F(\sigma_i) < F(\sigma_j)$, and $G(\sigma_i) < G(\sigma_j)$ all imply $i < j$. Let $\text{index} \colon K/\Lambda \to \mathbb{R}$ be the filter that assigns to each simplex its index in this order, and construct $\mathbb{M}_{\text{index}} = \mathbb{M}(\text{index}, \Lambda)$. Replacing the height $\text{h}(\Gamma) = i$ of a critical event by $\text{h}(\Gamma) = F(\sigma_i)$ and $\text{h}(\Gamma) = G(\sigma_i)$, respectively, we obtain two periodic merge trees, $\mathbb{M}_F$ and $\mathbb{M}_G$, whose only difference is the height function. However, since $F(\sigma_i) = F(\sigma_j)$ is possible even if $i \neq j$, some epochs in $\mathbb{M}_{\text{index}}$ may correspond to *empty epochs* in $\mathbb{M}_F$, and similarly for $\mathbb{M}_G$. When we construct the periodic barcodes, $\mathbb{B}_F = \mathbb{B}_0(F, \Lambda)$ and

$\mathbb{B}_G = \mathbb{B}_0(G, \Lambda)$, we use the empty epochs as well, while noting that each contributes two bars that cancel each other. The bijection between the epochs of $\mathbb{M}_F$ and $\mathbb{M}_G$ (both empty and non-empty) thus gives a bijection between $\mathbb{B}_F$ and $\mathbb{B}_G$, which is a transportation plan, $T\colon \operatorname{supp}(\xi) \times \operatorname{supp}(\eta) \to \mathbb{R}$, as in Definition 6.4.1. [1] [2] We prove that the cost of this particular transportation plan is bounded from above by $2(d+1)\mu_0 \cdot \|F/\Lambda - G/\Lambda\|_1$, which will imply

$$W_1^{\pm}(\mathbb{B}_F, \mathbb{B}_G) \leq 2(d+1)\mu_0 \cdot \|F/\Lambda - G/\Lambda\|_1. \tag{6.23}$$

By construction, we compare two bars if they are born and die at the hands of the same two simplices. Their contribution to T is therefore the difference in birth values plus the difference in death values, times the absolute multiplicity. Depending on the type of critical event a simplex causes, its value may mark more or fewer births or deaths. If σ causes a catenation or merge event, its value marks the death of at most three bars, with the absolute multiplicity of each bounded by μ_0. So the total cost for transporting the corresponding death values from F to G is bounded by $3\mu_0|F(\sigma) - G(\sigma)|$. If σ causes an appearance, its value may mark the births of any number of bars. However, within each era, absolute multiplicities form a telescoping series whose sum is bounded by $2\mu_0$. Since there are $d+1$ eras, the cost for transporting the corresponding birth values from F to G is bounded by $2(d+1)\mu_0 \cdot |F(\sigma) - G(\sigma)|$. Adding the cost for the death and the birth values implies (6.23).

Step 2. If F and G are not necessarily compatible with the same total order, we look at the straight-line homotopy between the two filters: $F_t = (1-t)F + tG$ for $t \in [0,1]$ and observe that

- $F_0 = F$ and $F_1 = G$;
- each F_t is a periodic filter with respect to Λ;
- for each $0 \leq s \leq t \leq 1$, $\|F_s/\Lambda - F_t/\Lambda\|_1 = (t-s)\|F/\Lambda - G/\Lambda\|_1$;
- there are only finitely many values $0 = t_0 < t_1 < \ldots < t_k = 1$ at which the total order compatible with F_t changes.

With this, we finish the proof by taking the sum of the bounds in (6.23) for each interval between two such values:

$$\begin{aligned} W_1^{\pm}(\mathbb{B}_F, \mathbb{B}_G) &\leq \sum\nolimits_1^k W_1^{\pm}(\mathbb{B}_0(F_{t_{i-1}}, \Lambda), \mathbb{B}_0(F_{t_i}, \Lambda)) && (6.24)\\ &\leq 2(d+1)\mu_0 \sum\nolimits_1^k \|F_{t_{i-1}}/\Lambda - F_{t_i}/\Lambda\|_1 && (6.25)\\ &= 2(d+1)\mu_0 \cdot \|F/\Lambda - G/\Lambda\|_1, && (6.26) \end{aligned}$$

which is the claimed inequality. □

Remark about invariance. When a perturbation breaks a symmetry and thus forces a coarser lattice, like in Figure 6.6, Theorem 6.4.6 can be applied to the filter before and after

[1] Whenever the transportation plan maps to or from a specific point on the diagonal, we could instead use X or Y to or from the closest point on the diagonal. For the sake of a simpler argument, we refrain from using X and Y and rely solely on T, which leads to a possible over-estimation of the 1-Wasserstein distance.

[2] Note also that instead of canceling the two bars contributed by an empty epoch, we keep both. This corresponds to looking at $W_1(\xi + \zeta, \eta + \zeta)$ instead of $W_1(\xi, \eta)$, but recall that they agree.

the perturbation, both considered under the coarse lattice. By Corollary 6.4.4, passing to an unnecessarily coarse lattice does not change the periodic barcode.

Remark about distortion. For a Λ-periodic $F\colon K \to \mathbb{R}$, and an arbitrarily small $\varepsilon > 0$, a uniform scaling of K and Λ by $1 + \varepsilon$ that preserves the filter values yields a periodic merge tree with slightly smaller shadow monomials. Hence, there is no arbitrarily small interleaving with the original periodic merge tree. However, the 1-Wasserstein distance between the corresponding periodic barcodes is less sensitive to small differences in the multiplicities and therefore the better choice in coping with the effect of slight scalings, or more general affine linear transformations with singular values close to 1.

Remark about Delaunay triangulations. Note that the perturbation of the vertices of a 2-dimensional Delaunay triangulation may change the filter values but possibly also the underlying graph, which happens only when four points lie on a common circle so one diagonal of a convex quadrangle *flips* to the other diagonal. In this case, neither diagonal contributes to the merge trees, because their endpoints are connected via shorter edges of the quadrangle. Hence, a diagonal flip does not affect the periodic merge tree. Since Theorems 6.3.7 and 6.4.6 apply to each part of the straight line homotopy between Delaunay triangulations during which the graph structure does not change—and there is no contribution due to the diagonal flips—the two theorems can be used to bound the distance between our descriptors of two Delaunay triangulations with possibly non-isomorphic edge-skeletons.

6.5 A Working Example

This section illustrates the concepts introduced in Sections 6.2 to 6.4 using the periodic graph in Figure 6.9 as a running example. Its quotient in the 3-dimensional torus has five vertices, labeled from 1 to 5, and eight edges, each labeled with its filter value; see Figure 6.8. If an edge has shadows whose endpoints lie in different copies of the unit cell, we choose an arbitrary direction and label the directed edge with the shift vector that locates the unit cell of the target vertex relative to that of the source vertex. In Figure 6.8, two of the blue edges connect to the same vertex twice, and they correspond to two families of parallel lines in $\mathbb{R}^3$: with diagonal direction defined by the drift vector $(1, 1, 0)$, and with vertical direction defined by the drift vector $(0, 0, 1)$. The loop formed by the two edges

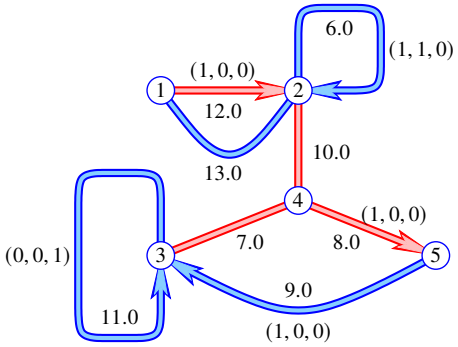

Figure 6.8: The quotient of the graph in Figure 6.9. Each edge is labeled by its value in the filter, and some also by their shift vectors. The spanning tree edges, which merge components, are drawn straight and *red*, and the others are drawn curved and *blue*. Each edge whose shadow has endpoints in different copies of the unit cell has a non-zero shift vector associated to a directed version of the edge.

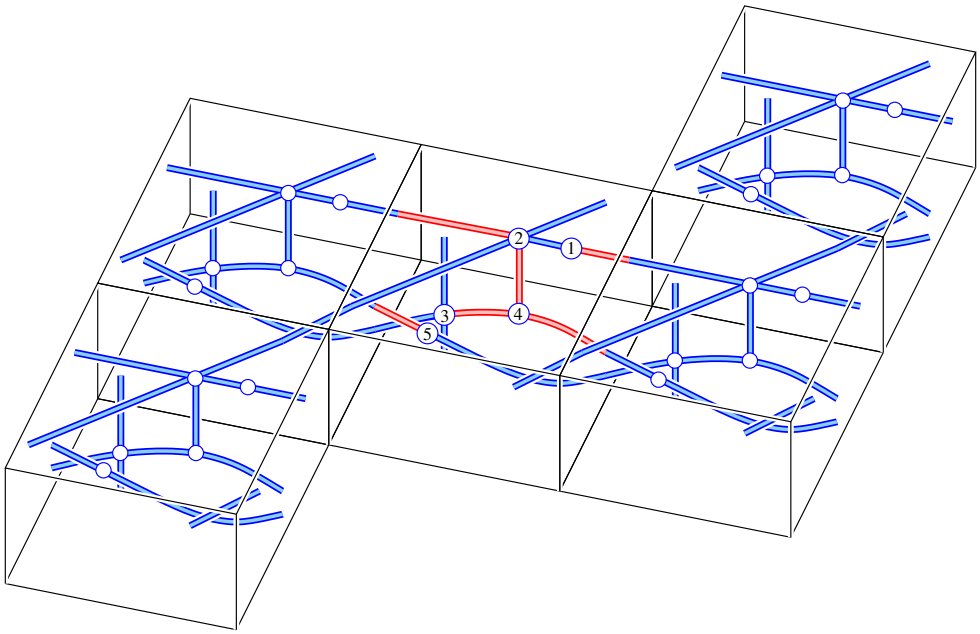

Figure 6.9: A periodic graph in $\mathbb{R}^3$ serving as a running example to illustrate the periodic merge tree and its construction. The portion of the graph inside five copies of the unit cell is shown. Inside the middle unit cell, the vertices are shown with labels and the edges in the spanning tree are drawn *red*. The motif within this cell consists of a cross above a double helix, with a single edge connecting the two, and, in addition, a vertical line sharing a point with the helix. This motif repeats periodically. The two lines of the cross meet at vertex 2, and together with their periodic translates form a series of 2-dimensional grids. The two strands of the double helix are periodic translates of each other and pass from left to right through copies of vertices 3, 4, 5, in this cyclic sequence. For convenience, the vertex labels are also used as filter values, and the values of the edges are shown in Figure 6.8, which displays the corresponding quotient, i.e. the graph in the 3-dimensional torus.

connecting vertices 1 and 2 corresponds to another family of lines with direction defined by $(1, 0, 0)$. More interesting is the loop formed by the three edges connecting vertices 3, 4, and 5. The drift vector is $(2, 0, 0)$, which implies that two copies of the strand (drawn curved in Figure 6.9 for better visibility) form a double helix that passes through a linear sequence of unit cells.

Next we go through the motion of constructing the periodic merge tree incrementally, by adding one vertex or edge at a time in the order of increasing filter value. The top panel of Figure 6.10 shows the final tree. We pay special attention to the periodicity lattices, which are maintained incrementally and used to compute the shadow monomials decorating the beams of the tree. For simplicity, we assume that the underlying lattice is $\Lambda = \mathbb{Z}^3$, whose unit cell is the unit cube with unit volume.

STEPS 1 TO 5: adding the five vertices, we get five connected components, each with trivial periodicity lattice, $\Lambda_1 = \Lambda_2 = \Lambda_3 = \Lambda_4 = \Lambda_5 = \{0\}$, in which we use the labels of the vertices as indices. Each of these lattices has 0-dimensional volume 1, which implies that the shadow monomial is $\frac{4\pi}{3}R^3$; see the left ends of the beams in the top panel of Figure 6.10.

STEP 6: the edge with filter value 6.0 connects vertex 2 to itself, and its addition changes the periodicity lattice of the corresponding component to $\Lambda_2 = \Lambda((1, 1, 0))$. Its unit

cell has 1-dimensional volume $\sqrt{2}$. Note that the dimension of the periodicity lattice increases from 0 to 1, and the shadow monomial decreases from cubic to quadratic. The drop of the exponent means the component leaves the cubic and enters the quadratic era.

STEPS 7 TO 9: the edges with filter values $7.0, 8.0, 9.0$ form the helix in the periodic graph. The addition of the first edge merges the components of vertices 3 and 4, and the addition of the second edge merges this component with that of vertex 5. These mergers do not affect the periodicity lattices. However, the third edge completes the helix with periodicity lattice $\Lambda_3 = \Lambda((2,0,0))$ and shadow monomial $2\pi R^2$. The factor 2 indicates that shifting a strand by $(1,0,0)$ produces a second strand, so we are dealing with a double and not a single helix.

STEP 10: the edge with filter value 10.0 connects the line from Step 6 with the helix from Step 9. The corresponding periodicity lattice is spanned by the vectors $(1,1,0)$ and $(2,0,0)$, which after reduction becomes $\Lambda_2 = \Lambda((1,1,0),(0,2,0))$. The 2-dimensional volume of its unit cell is 2, and so is the 1-dimensional volume of the unit ball in $\mathbb{R}^1$, which is why the shadow monomial is $4R$; see Figure 6.10. Note that the component leaves the quadratic and enters the linear era.

STEP 11: the edge with filter value 11.0 connects vertex 3 to itself. Its addition combines the 2-dimensional structures from Step 10 to form a bigger, 3-dimensional structure. Indeed, the periodicity lattice is spanned by vectors $(1,1,0)$, $(0,2,0)$, and $(0,0,1)$, whose unit cell has 3-dimensional volume 2. The fact that the volume is 2 and not 1 suggests that the two strands of the double helix are still in separate components. The corresponding shadow monomial is 2, so the component leaves the linear and enters the constant era.

STEPS 12 AND 13: until now, vertex 1 was isolated, with its own beam in the periodic merge tree. The first edge with endpoints 1 and 2 merges the two components, keeping the larger of the two periodicity lattices unchanged, but assigning it now to beam 1, by the elder rule. The second such edge forms a loop with drift vector $(1,0,0)$. We use the reduction algorithm to select a basis from the three vectors spanning the lattice constructed in Step 11 and the drift vector of the new loop. The result is $\Lambda_1 = \Lambda((1,0,0),(0,1,0),(0,0,1))$, and the shadow monomial is 1, so this step moves the one remaining component from a first to the second and final epoch in the constant era; see the right end of the tree in the top panel of Figure 6.10.

Remark about stability. If we alter the value of the second vertex from 2 to $1+\varepsilon$, for an arbitrarily small $\varepsilon > 0$, the periodic merge tree and periodic barcode do not change combinatorially, but if we alter it to $1-\varepsilon$, then this vertex swaps with the current first vertex and thus changes the combinatorics of the tree and barcode. While the adjustment from $1+\varepsilon$ to $1-\varepsilon$ mostly affects birth values in the arbitrarily small interval between these value, there is one change worth noting: the two epochs with shadow monomial 2—which belong to different beams in the tree shown in Figure 6.10—merge into one epoch on one beam. Therefore, the points $(2,12)$ with multiplicity 2 and $(1,12)$ with multiplicity -2 disappear. This illustrates the importance of negative multiplicities, which are essential for stability as they facilitate the matching of such pairs using the alternating 1-Wasserstein distance.

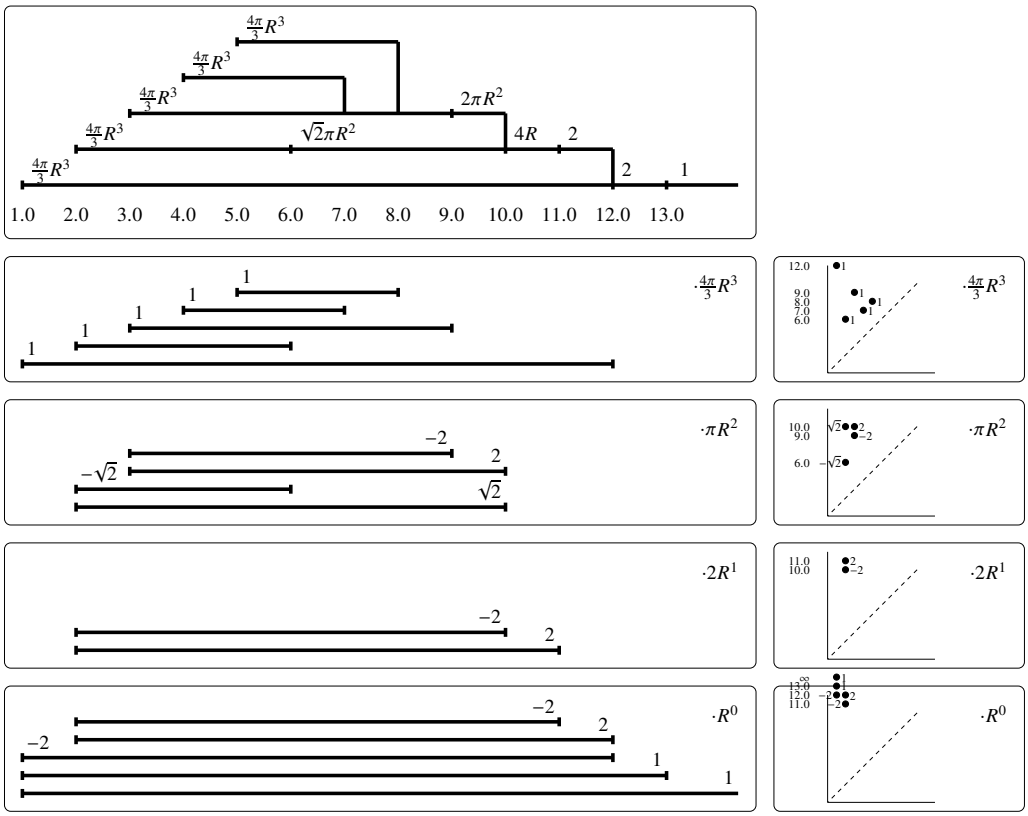

Figure 6.10: *Top:* the periodic merge tree of the filtered graph in Figure 6.9. The left endpoints of the beams correspond to vertices of the quotient graph, and the vertical segments correspond to edges in its spanning tree. The other edges form loops and thus do not affect the set of components but possibly alter the shadow monomials, which decorate the beams. *Below the top on the left:* the four collections of bars labeled with their possibly negative multiplicities. *On the right:* the corresponding four persistence diagrams with similarly labeled points. The bottommost constant diagram has a point with infinite vertical coordinate, which is drawn above its window.

6.6 Discussion

The main contribution of this chapter is the extension of the persistent homology framework to the periodic setting, albeit only for connected components; that is: for homological dimension 0. The crucial ingredient in this extension is the notion of a *shadow monomial*, which quantifies the growth-rate and density of the translates of a component. The two new data structures are the *periodic merge tree* and the *periodic* 0*-barcode*, with the former containing strictly more information and the latter offering computational advantages and an easier connection to machine learning tools; see the text by Peyré and Cuturi [PC19] for optimal transport algorithms and their application to data science. Both data structures are invariant under isometries and a change of basis, which can be seen directly from the definition of the shadow monomial. While the periodic merge tree splinters when we pass to a sublattice of the original lattice, the periodic 0-th barcode is invariant under such changes, and so are the equivalence classes of periodic merge trees. Both data structures are stable under perturbations.

When defining a pseudo-distance between periodic merge trees, we chose the path of first defining an interleaving distance that is sensitive to splintering and then in a second

step making it insensitive to splintering via the equivalence relation. The idea is that equivalent representations of periodic merge trees carry the same information, but some representations might be more suitable for comparison. In particular, we get an unnecessary large distance, when the two representations differ in how finely they are splintered; see the example of Figure 6.4, which get distance 6 assigned in the interleaving distance, but distance 0 after the correction via equivalence classes. An alternative approach would have been to try to define directly a pseudo-distance between periodic merge trees that is insensitive to splintering. A promising approach in this direction seems to replace the ε-compatible maps in the definition of interleaving distance by some kind of ε-compatible transport plans. However, the details of this approach would need to be worked out, and it is not clear if it would make the exposition clearer.

Possible applications of the method are for example

- Any application where connections in a periodic or periodically-simulated material or in its complement are guiding physical or chemical properties of the material; see, for example, [KHM20].
- In the search for a crystalline fingerprint [EHK+21, WK22], facilitating fast comparison between crystalline materials, the density fingerprint has proven useful [EHK+21, HMW+24]. Once extended to higher homological dimensions, the periodic barcode applied to order-k Delaunay filtrations for several values of k has potential to be a more concise fingerprint with similar information content as the density fingerprint.
- Usually we do not evaluate the shadow monomial for a specific value of R but rather treat R as a parameter going to infinity. However, in some applications it makes sense to replace R by a specific real number, evaluating all the monomials to real numbers. In that case, there is no need to treat the $d+1$ different eras separately, yielding just one barcode, instead of $d+1$ separate ones. An example for an application where this would make sense, is when wanting to compare a finite piece of an imperfect periodic material (of unknown periodicity) observed in nature, with a database of perfectly periodic simulated materials (of known periodicity). For simplicity, let us assume the piece has the shape of a ball, so that we can replace R by the radius of the piece.

The reported work suggests several avenues of future study aimed at broadening and deepening the capabilities of the method and thus make it more compelling in applications to the sciences.

- For the sake of convenience, we restricted ourselves to d-periodic filtrations in $\mathbb{R}^d$. However, the theory extends to the setting of k-periodic filtrations in $\mathbb{R}^d$, by replacing all d by k in the definition of the shadow monomial. The non-periodic case, $k=0$, then yields shadow monomials equal to 1 and thus the classical merge tree and 0-th persistent homology. To what extent can the theory, and in particular Definition 6.1.2 and Lemma 6.1.3, be further extended to groups acting on topological spaces different from $\mathbb{R}^d$?
- Further extend the results of this chapter to homological dimensions beyond 0. Some such capability can already be achieved by applying the periodic merge tree to the dual filtration, describing the complements of the original filtration. While this idea is not new [DE95], a key difference is that the infinite periodic space is not compact, and

therefore Alexander duality does not hold in the usual way, which can be seen in the right half of Figure 7.1 in Chapter 7. As a consequence, we not only get information about $(d-1)$- but also about $(d-2)$-dimensional homology. In $\mathbb{R}^3$, 2-dimensional homology (voids) of the original filtration corresponds to the cubic eras of the dual filtration. In contrast, the quadratic eras, for example, describe infinitely long tunnels, so a particular kind of 1-dimensional homology.

- Assuming success in extending periodic barcodes to higher homological dimension, it will be interesting to combine the periodic with the chromatic setting recently developed in [CdMDES24]. Indeed, crystalline materials are periodic and typically consist of more than one type of atoms, which suggests both concepts in its analysis.

We conclude this chapter with a concrete geometric question about the Delaunay triangulation of a locally finite Λ-periodic point set in $\mathbb{R}^d$: does there exist a constant $D = D(d)$ and a basis of Λ such that the shift vector of every edge in the Delaunay triangulation has magnitude at most D? We recall that this constant is $D = 1$ if Λ has a basis with pairwise orthogonal vectors, as proved in Section 6.2.5. However, the proof does not extend to the case in which the basis vectors enclose arbitrary angles. Is there a similar bound for nearly orthogonal bases constructed with the Lenstra–Lenstra–Lovász (LLL) lattice basis reduction algorithm?

CHAPTER 7

Conclusion

This thesis provides answers to four different research questions arising when trying to apply topological data analysis methods to materials science data.

- Research Question 1 is answered thoroughly, by showing that the choice of digital topology can be made purely based on the needs of the application, not limited by available software. This is because software can be modified easily to produce results for the other digital topology. Our results have already been used by the software cubical ripser [KSA20].
- Research Question 2 about the choice of fine enough resolution for digital images does not have a clear answer. Depending on the specific application, different assumptions about the data can be made. We therefore provide guidance for various different assumptions.
- Research Question 3 is answered by providing a specific fingerprint for crystals and proving its properties. However, the search for fingerprints with even better properties (completeness not only generically, faster computations, smaller Lipschitz constant, etc.) continues [WK22]. In particular, one limitation of the density fingerprint is that in practice, it is only computed for finitely many radius values, while the proof for generic completeness relies on being able to take derivatives with respect to the radius. Nevertheless, the density fingerprint has been successfully applied already [HMW$^+$24].
- Research Question 4 is answered in part, by providing a definition, an algorithm, and proven properties for 0-dimensional persistent homology of periodic data. In addition, we suggest to use dualities to infer some but not all information about other homological dimensions.

One of the aspects that makes materials science data difficult is that it is often periodic. For example, crystalline materials are periodic. The fact that a perfectly periodic structure consists of infinitely many repetitions makes it lose properties that finite structures have. For example, periodic filters are not tame, which is a problem for the definition of persistent homology. We solve this problem in Chapter 6 for homological dimension 0, but it is still open for other homological dimensions. Another example showing that periodic filters

do not have certain properties finite filters have, has already been hinted at in Section Section 1.1. Now, with the language of Chapter 6 available, we can go into more detail. The duality result (Theorem 3.1.4) of Chapter 3 does not hold in the infinite periodic setting of Chapter 6 because of its assumption that the complexes are finite. It is easy to construct a counter-example; see Figure 7.1.

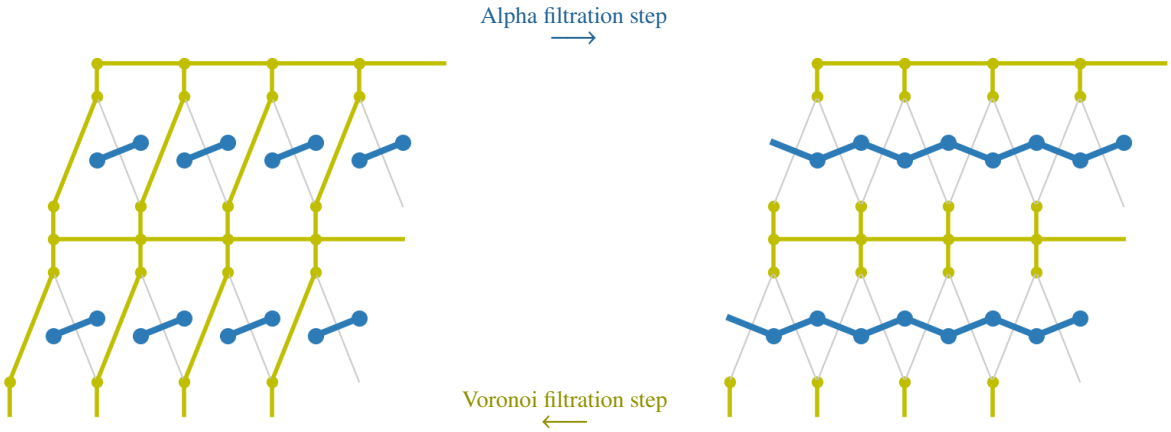

Figure 7.1: The *blue* complexes are consecutive Alpha complexes of the *blue* periodic point set. From the left to the right Alpha complex, one blue edge (along with its periodic copies) is added. This addition causes a quadratic amount of 0-th persistence bars to die and there is no 1-homology being born. The *yellow* complexes are consecutive complexes in the dual Voronoi filtration. (The missing Voronoi edges are in light gray.) From the right to the left complex, one yellow edge (along with its periodic copies) is added. This addition causes a linear amount of 0-th persistence bars to die (along with the births of a quadratic amount of 1-homology). This shows that the linearly many yellow death edges are dual to blue death edges, which is not allowed in Theorem 3.1.4, where death is always dual to birth.

Another problem about periodic data is that a small perturbation can cause the fundamental domain (or primitive unit cell) to change, for example, to twice its size; see Figure 6.6. In Chapter 6 this causes the complication of splinterings (see Theorem 6.3.5) which shows that the state-of-the-art approach of computing the merge tree (or persistent homology) of the quotient filter on the torus is not stable under such perturbations. We solve this problem by introducing the shadow monomial, decorating the merge tree of the quotient complex. With the shadow monomial, it is not a problem that splintering replaces a persistence bar by several copies of it, because the shadow monomials of the new bars sum up to the original shadow monomial. Therefore, splintering preserves the multiplicities of the persistence bars (see Theorem 6.4.3). In Chapter 5, we solve this problem by not using homology as a descriptor but instead a volumetric descriptor that is naturally invariant under such changes of the fundamental domain.

The importance of the presented research stems from the role that materials science plays in mitigating climate change. The new methods provided in this thesis can help in the search for new materials with better chemical or physical properties, by gaining more insights about the shape of the material and how it determines these properties. Improving the design of new materials can help in making batteries more efficient or improving carbon capture, and thus contributes to the mitigation of one of humanities most important problems: climate change.

On the application side, I plan to extend this current line of research together with my chemistry and topology collaborators from Aalborg University. We plan on applying the methods

of Chapter 6 to estimate the conductivity of their periodic simulations of different aperiodic glass materials. Estimating the conductivity of these materials efficiently with topological methods allows for finding the few most promising candidates for efficient batteries, which can then be studied more thoroughly with expensive simulations on supercomputers or even in the laboratory. The first, easiest step in that direction is to implement the algorithm given as pseudo-code in Section 6.2.

On the theory side, the obvious next step is extending the theory of Chapter 6 to arbitrary dimensional persistent homology. While we are well aware of the problems facing us in this endeavour, we have ideas how to potentially overcome them.

Another interesting direction of further research, related to Chapter 5 and Chapter 6, is proving whether the collection of persistence diagrams of all order-k Delaunay mosaics characterizes a point set uniquely, or at least uniquely up to a measure zero set. The question would both be interesting for finite and infinite periodic point sets. If the answer is positive, the result would be similar in spirit to the injectivity of the persistent homology transform [TMB14]. If the answer is negative, maybe the 2-parameter persistence module (where one parameter is k and the other is the radius r) is enough to characterize point sets uniquely.

Bibliography

[ABI88] David Avis, Binay K. Bhattacharya, and Hiroshi Imai. Computing the volume of the union of spheres. *The Visual Computer*, 3(6):323–328, 1988.

[ABL+13] Ann S. Almgren, John B. Bell, Mike J. Lijewski, Zarija Lukić, and Ethan Van Andel. Nyx: A massively parallel amr code for computational cosmology. *The Astrophysical Journal*, 765(1):39, feb 2013.

[ACDL00] Nina Amenta, Sunghee Choi, Tamal K. Dey, and Naveen Leekha. A Simple Algorithm for Homeomorphic Surface Reconstruction. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, SCG '00, pages 213–222. ACM, 2000.

[AEK+17] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

[Ale15] J. W. Alexander, II. A proof of the invariance of certain constants of analysis situs. *Transactions of the American Mathematical Society*, 16(2):148–154, 1915.

[ALS13] Dominique Attali, André Lieutier, and David Salinas. Vietoris-rips complexes also provide topologically correct reconstructions of sampled shapes. *27th Annual Symposium on Computational Geometry*, 46, 05 2013.

[ARC14] Aaron Adcock, Daniel Rubin, and Gunnar Carlsson. Classification of hepatic lesions using the matching metric. *Computer Vision and Image Understanding*, 121:36–42, 2014.

[Aur87] Franz Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[BBI01] Dmitri Burago, Yuri Burago, and Sergei Ivanov. *A course in metric geometry*, volume 33. American Mathematical Society Providence, 2001.

[BEK10] Paul Bendich, Herbert Edelsbrunner, and Michael Kerber. Computing robustness and persistence for images. *IEEE Transactions on Visualization and Computer Graphics*, 16:1251–1260, 2010.

[BGH+22] Bea Bleile, Adélie Garin, Teresa Heiss, Kelly Maggs, and Vanessa Robins. The persistent homology of dual digital image constructions. In Ellen Gasparovic, Vanessa Robins, and Katharine Turner, editors, *Research in Computational Topology 2*, pages 1–26. Springer International Publishing, Cham, 2022.

[BK04] Mireille Boutin and Gregor Kemper. On reconstructing n-point configurations from the distribution of distances or areas. *Advances in Applied Mathematics*, 32(4):709–735, 2004.

[Blu67] Harry Blum. A transformation for extracting descriptors of shape. In *Models for the Perception of Speech and Visual Forms*, pages 362–380. MIT Press, 1967.

[BM21] Peter Bubenik and Nikola Milićević. Homological algebra for persistence modules. *Foundations of Computational Mathematics*, Jan 2021.

[BS16] Pieter W.S.K. Botha and Adrian P. Sheppard. Mapping permeability in low-resolution micro-CT images: A multiscale statistical approach. *Water Resources Research*, 52(6):4377–4398, June 2016.

[C+96] CGAL Consortium et al. CGAL, computational geometry algorithms library. `https://www.cgal.org`, 1996.

[Cas97] John William Scott Cassels. *An introduction to the geometry of numbers*. Springer, 1997.

[CCSL09] Frédéric Chazal, David Cohen-Steiner, and André Lieutier. A Sampling Theory for Compact Sets in Euclidean Space. *Discrete Comput Geom*, 41(3):461–479, April 2009.

[CdMDES24] Sebastiano Cultrera di Montesano, Ondřej Draganov, Herbert Edelsbrunner, and Morteza Saghafian. Chromatic alpha complexes. *arXiv preprint arXiv:2212.03128*, 2024.

[CDS+23] Zhimin Chen, Tao Du, Søren S. Sørensen, Rasmus Christensen, Qi Zhang, Lars R. Jensen, Oxana V. Magdysyuk, Maria Diaz-Lopez, Mathieu Bauchy, Yuanzheng Yue, and Morten M. Smedskjaer. Structure dependence of fracture toughness and ionic conductivity in lithium borophosphate glassy electrolytes for all-solid-state batteries. *Journal of Power Sources*, 553:232302, 2023.

[CDSGO16] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. Springer, 2016.

[CM90] Edith Cohen and Nimrod Megiddo. Recognizing properties of periodic graphs. In *Applied geometry and discrete mathematics*, pages 135–146, 1990.

[CM05] James A. Chisholm and Sam Motherwell. Compack: a program for identifying crystal structure similarity using distances. *Journal of Applied Crystallography*, 38(1):228–231, 2005.

[CSEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & computational geometry*, 37(1):103–120, 2007.

[CSEH09] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*, 9:79–103, Feb 2009.

[CSEM06] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, SCG '06, page 119–126, New York, NY, USA, 2006. Association for Computing Machinery.

[CSMR+23] Salvador Chulián, Bernadette J Stolz, Álvaro Martínez-Rubio, Cristina Blázquez Goñi, Juan F Rodríguez Gutiérrez, Teresa Caballero Velázquez, Águeda Molinos Quintana, Manuel Ramírez Orellana, Ana Castillo Robleda, José Luis Fuster Soler, et al. The shape of cancer relapse: Topological data analysis predicts recurrence in paediatric acute lymphoblastic leukaemia. *PLOS Computational Biology*, 19(8):e1011329, 2023.

[DD89] Warren Dicks and Martin John Dunwoody. *Groups acting on graphs*, volume 17. Cambridge University Press, 1989.

[DE95] Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995.

[DFR15] Olaf Delgado-Friedrichs and Vanessa Robins. diamorse: Digital image analysis using discrete morse theory and persistent homology. `https://github.com/AppliedMathematicsANU/diamorse`, 2015.

[DH97] Nikolai P. Dolbilin and Daniel H. Huson. Periodic Delone tilings. *Periodica Mathematica Hungarica*, 34:57–64, 1997.

[DLS98] Nikolai P. Dolbilin, Jeffrey C. Lagarias, and Marjorie Senechal. Multiregular point systems. *Discrete & Computational Geometry*, 20(4):477–498, 1998.

[DRS14] Olaf Delgado-Friedrichs, Vanessa Robins, and Adrian Sheppard. Morse theory and persistent homology for topological analysis of 3D images of complex materials. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4872–4876, October 2014.

[DRS15] Olaf Delgado-Friedrichs, Vanessa Robins, and Adrian Sheppard. Skeletonization and partitioning of digital images using discrete Morse theory. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):654–666, 2015.

[DSG07] Vin De Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

[DSMVJ11] Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27, July 2011.

[DW18] Paweł Dłotko and Thomas Wanner. Rigorous cubical approximation and persistent homology of continuous functions. *Computers & Mathematics with Applications*, 75(5):1648–1666, March 2018.

[Ede87] Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10. Springer-Verlag Berlin Heidelberg, 1987.

[Ede93] Herbert Edelsbrunner. The union of balls and its dual shape. In *Proceedings of the ninth annual symposium on Computational geometry*, pages 218–231, 1993.

[EF94] Herbert Edelsbrunner and Ping Fu. Measuring space filling diagrams and voids. In *Molecular Biophysic Report UIUC-BI-MB-94–01*. University of Illinois at Urbana-Champaign, 1994.

[EGG+24] Herbert Edelsbrunner, Alexey Garber, Mohadese Ghafari, Teresa Heiss, Morteza Saghafian, and Mathijs Wintraecken. Brillouin zones of integer lattices and their perturbations. *SIAM Journal on Discrete Mathematics*, 38(2):1784–1807, 2024.

[EH08] Herbert Edelsbrunner and John Harer. Persistent homology—a survey. In *Surveys on discrete and computational geometry*, volume 453 of *Contemp. Math.*, pages 257–282. Amer. Math. Soc., Providence, RI, 2008.

[EH10] Herbert Edelsbrunner and John L. Harer. *Computational topology: an introduction*. American Mathematical Society, 2010.

[EH24] Herbert Edelsbrunner and Teresa Heiss. Merge trees of periodic filtrations. *arXiv preprint arXiv:2408.16575*, 2024.

[EHK+21] Herbert Edelsbrunner, Teresa Heiss, Vitaliy Kurlin, Philip Smith, and Mathijs Wintraecken. The density fingerprint of a periodic point set. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[EIH18] Herbert Edelsbrunner and Mabel Iglesias-Ham. Multiple covers with balls i: inclusion–exclusion. *Computational Geometry*, 68:119–133, 2018.

[EÖ19] Herbert Edelsbrunner and Katharina Ölsböck. Holes and dependences in an ordered complex. *Computer Aided Geometric Design*, 73:1–15, 2019.

[EÖ20] Herbert Edelsbrunner and Katharina Ölsböck. Tri-partitions and bases of an ordered complex. *Discrete & Computational Geometry*, pages 1–17, 2020.

[ES12] Herbert Edelsbrunner and Olga Symonova. The adaptive topology of a digital image. In *Proceedings of the 2012 9th International Symposium on Voronoi Diagrams in Science and Engineering, ISVD 2012*, pages 41–48. IEEE, June 2012.

[Fed59] Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–491, 1959.

[FOT+16] Gemma de la Flor, Danel Orobengoa, Emre Tasci, Juan M. Perez-Mato, and Mois I. Aroyo. Comparison of structures applying the tools available at the bilbao crystallographic server. *Journal of Applied Crystallography*, 49(2):653–664, 2016.

[GHM+20] Adélie Garin, Teresa Heiss, Kelly Maggs, Bea Bleile, and Vanessa Robins. Duality in persistent homology of images. *Extended abstract SoCG YRF*, pages 15–19, 2020.

[GI91] Zvi Galil and Giuseppe F. Italiano. Data structures and algorithms for disjoint set union problems. *ACM Computing Surveys (CSUR)*, 23(3):319–344, 1991.

[GM95] F. Alberto Grünbaum and Calvin C. Moore. The use of higher-order invariants in the determination of generalized patterson cyclotomic sets. *Acta Crystallographica*, A51(3):310–323, 1995.

[Hat02] Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002.

[HHA+13] Anna L. Herring, Elizabeth J. Harper, Linnéa Andersson, Adrian Sheppard, Brian K. Bay, and Dorthe Wildenschild. Effect of fluid topology on residual nonwetting phase trapping: Implications for geologic CO2 sequestration. *Advances in Water Resources*, 62, Part A:47–58, December 2013.

[HHS21] Ruotong Huang, Anna L. Herring, and Adrian Sheppard. Effect of Saturation and Image Resolution on Representative Elementary Volume and Topological Quantification: An Experimental Study on Bentheimer Sandstone Using Micro-CT. *Transport in Porous Media*, 137(3):489–518, April 2021.

[HLCK21] Chuan-Shen Hu, Austin Lawson, Yu-Min Chung, and Kaitlin Keegan. Two-parameter persistence for images via distance transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 4176–4184, October 2021.

[HMM98] George Havas, Bohdan S. Majewski, and Keith R. Matthews. Extended gcd and hermite normal form algorithms via lattice basis reduction. *Experimental Mathematics*, 7(2):125–136, 1998.

[HMW+24] Chuan-Shen Hu, Rishikanta Mayengbam, Min-Chun Wu, Kelin Xia, and Tze Chien Sum. Geometric data analysis-based machine learning for two-dimensional perovskite design. *Communications Materials*, 5(1):106, 2024.

[HNH+16] Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.

[HRS19] Anna L. Herring, Vanessa Robins, and Adrian P. Sheppard. Topological Persistence for Relating Microstructure and Capillary Fluid Trapping in Sandstones. *Water Resources Research*, 55(1):555–573, January 2019.

[HTS+21a] Teresa Heiss, Sarah Tymochko, Brittany Story, Adélie Garin, Hoa Bui, Bea Bleile, and Vanessa Robins. The impact of changes in resolution on the persistent homology of images. *arXiv preprint arXiv:2111.05663*, 2021.

[HTS+21b] Teresa Heiss, Sarah Tymochko, Brittany Story, Adélie Garin, Hoa Bui, Bea Bleile, and Vanessa Robins. The impact of changes in resolution on the persistent homology of images. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3824–3834, 2021.

[KB79] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *siam Journal on Computing*, 8(4):499–507, 1979.

[KHM20] Aditi S. Krishnapriyan, Maciej Haranczyk, and Dmitriy Morozov. Topological descriptors help predict guest adsorption in nanoporous materials. *The Journal of Physical Chemistry C*, 124(17):9360–9368, 2020.

[KMN17] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. Geometry helps to compare persistence diagrams. *ACM J. Exp. Algorithmics*, 22, September 2017.

[Kov89] Vladimir A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46(2):141–161, May 1989.

[KR23] Kim-Manuel Klein and Janina Reuter. Simple lattice basis computation – the generalization of the euclidean algorithm. *arXiv preprint arXiv:2311.15902*, 2023.

[Kru56] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[KSA20] Shizuo Kaji, Takeki Sudo, and Kazushi Ahara. Cubical ripser: Software for computing persistent homology of image and volume data. *arXiv preprint arXiv:2005.12692*, 2020.

[KSC+20] Jisu Kim, Jaehyeok Shin, Frédéric Chazal, Alessandro Rinaldo, and Larry Wasserman. Homotopy reconstruction via the cech complex and the vietoris-rips complex. In *36th International Symposium on Computational Geometry (SoCG 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[LBD+17] Yongjin Lee, Senja D. Barthel, Paweł Dłotko, S. Mohamad Moosavi, Kathryn Hess, and Berend Smit. Quantifying similarity of pore-geometry in nanoporous materials. *Nature communications*, 8(1):1–8, 2017.

[LSS03] Paul Lemke, Steven S. Skiena, and Warren D. Smith. *Reconstructing sets from interpoint distances*. Springer Berlin Heidelberg, 2003.

[Mad00] Adolf Mader. *Almost completely decomposable groups*. CRC Press, 2000.

[Mar23] Chiara Martyka. Periodic persistent 0-homology. Master's thesis, University of Vienna, Vienna, Austria, July 2023.

[MBW13] Dmitriy Morozov, Kenes Beketayev, and Gunther Weber. Interleaving distance between merge trees. *Discrete and Computational Geometry*, 49(22-45):52, 2013.

[MEM+06] Clare F. Macrae, Paul R. Edgington, Patrick McCabe, Elna Pidcock, Greg P. Shields, Robin Taylor, Matthew Towler, and Jacco van de Streek. Mercury: visualization and analysis of crystal structures. *Journal of Applied Crystallography*, 39(3):453–457, 2006.

[Mil20] Ezra Miller. Homological algebra of modules over posets. *arXiv preprint arXiv:2008.00063*, 2020.

[Mun84] James R. Munkres. *Elements of algebraic topology*. Addison-Wesley Publishing Company, Menlo Park, CA, 1984.

[Nig28] Paul Niggli. *Krystallographische und strukturtheoretische Grundbegriffe*, volume 1. Akademische verlagsgesellschaft mbH, 1928.

[NS04] Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. *In "Proceedings of the 6th International Algorithmic Number Theory Symposium"*, pages 338–357, 2004.

[NSW08] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the Homology of Submanifolds with High Confidence from Random Samples. *Discrete Comput Geom*, 39(1-3):419–441, March 2008.

[NZ01] M. E. J. Newman and Robert M. Ziff. Fast monte carlo algorithm for site or bond percolation. *Phys. Rev. E*, 64:016706, Jun 2001.

[OR22] Adam Onus and Vanessa Robins. Quantifying the homology of periodic cell complexes. *arXiv preprint arXiv:2208.09223*, 2022.

[PAB+19] Pratyush Pranav, Robert J. Adler, Thomas Buchert, Herbert Edelsbrunner, Bernard J. T. Jones, Armin Schwartzman, Hubert Wagner, and Rien van de Weygaert. Unexpected topology of the temperature fluctuations in the cosmic microwave background. *Astronomy & Astrophysics (A&A)*, 627:A163:1–30, 2019.

[PC14] Jose A. Perea and Gunnar Carlsson. A klein-bottle-based dictionary for texture representation. *International journal of computer vision*, 107:75–97, 2014.

[PC19] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[Pea17] Angeles Pulido et al. Functional materials discovery using energy–structure–function maps. *Nature*, 543(7647):657–664, 2017.

[PEv+16] Pratyush Pranav, Herbert Edelsbrunner, Rien van de Weygaert, Gert Vegter, Michael Kerber, Bernard J. T. Jones, and Mathijs Wintraecken. The topology of the cosmic web in terms of persistent Betti numbers. *Monthly Notices of the Royal Astronomical Society*, 465(4):4281–4310, 11 2016.

[PS30] Linus Pauling and Maple D. Shappell. 8. the crystal structure of bixbyite and the c-modification of the sesquioxides. *Zeitschrift fü Kristallographie - Crystalline Materials*, 75(1):128–142, 1930.

[RNS+17] Michael W. Reimann, Max Nolte, Martina Scolamiero, KatharineTurner, Rodrigo Perin, Giuseppe Chindemi, Paweł Dłotko, Ran Levi, Kathryn Hess, and Henry Markram. Cliques of neurons bound into cavities provide a missing link between structure and function. *Frontiers in Computational Neuroscience*, 11, 2017.

[RSDS16] Vanessa Robins, Mohammad Saadatfar, Olaf Delgado-Friedrichs, and Adrian P. Sheppard. Percolating length scales from topological persistence analysis of micro-CT images of porous materials. *Water Resources Research*, 52(1):315–329, January 2016.

[RWS11] Vanessa Robins, Peter J. Wood, and Adrian P. Sheppard. Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1646–1658, 05 2011.

[Sch98] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[Sen08] Marjorie Senechal. A point set puzzle revisited. *European Journal of Combinatorics*, 29(8):1933–1944, 2008.

[SM20] Dmitriy Smirnov and Dmitriy Morozov. Triplet merge trees. In *Topological Methods in Data Analysis and Visualization V: Theory, Algorithms, and Applications 7*, pages 19–36. Springer, 2020.

[Smi20] Philip Smith. Density functions of a periodic set in c++. `https://github.com/Phil-Smith1/Density_Functions`, 2020.

[SP15] Adrian Sheppard and Masa Prodanovic. Network generation comparison forum. `http://www.digitalrocksportal.org/projects/16`, 2015.

[ST20] Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*, 2020.

[Sto00] Arne Storjohann. *Algorithms for matrix canonical forms*. PhD thesis, ETH Zurich, 2000.

[TAS+23] Signe Sidwall Thygesen, Alexei I. Abrikosov, Peter Steneteg, Talha Bin Masood, and Ingrid Hotz. Level of detail visual analysis of structures in solid-state materials. In *EuroVis (Short Papers)*, pages 55–59, 2023.

[Thä08] Christoph Thäle. 50 Years of sets with positive reach, a survey. *Surveys in Mathematics and its Applications*, 3:123–165, 2008.

[TLT+20] Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Anibal Medina-Mardones, Alberto Dassatti, and Kathryn Hess. giotto-tda: A topological data analysis toolkit for machine learning and data exploration. *arXiv preprint arXiv:2004.02551*, 2020.

[TMB14] Katharine Turner, Sayan Mukherjee, and Doug M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[Tym] Sarah Tymochko. PH-of-Images GitHub Repository. https://github.com/sarahtymochko/PH-of-Images/.

[VYP18] Hanife Varlı, Yağmur Yılmaz, and Mehmetcik Pamuk. Homological properties of persistent homology. *arXiv preprint arXiv:1805.01274*, 2018.

[Wag18] Hubert Wagner. Cubicle: Streaming computations for persistent homology of large greyscale images. https://bitbucket.org/hubwag/cubicle/src/master/, 2018.

[Wag23] Hubert Wagner. Slice, simplify and stitch: Topology-preserving simplification scheme for massive voxel data. In *39th International Symposium on Computational Geometry (SoCG 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[WK22] Daniel Widdowson and Vitaliy Kurlin. Resolving the data ambiguity for periodic crystals. In Sanmi Koyejo, Shakir Mohamed, Alekh Agarwal, Danielle Belgrave, Kyunghyun Cho, and Alice Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24625–24638. Curran Associates, Inc., 2022.

[WS13] Dorthe Wildenschild and Adrian P. Sheppard. X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems. *Advances in Water Resources*, 51:217–246, 2013.

[ZC05] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[Zhi15] Boris Zhilinskii. *Introduction to Louis Michel's lattice geometry through group action*. EDP sciences, CNRS Éditions, 2015.