Approximation Algorithms for Vietoris-Rips and Čech Filtrations

Dissertation

zur Erlangung des Grades des
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

vorgelegt von

Aruni Choudhary

Saarbrücken
2017

Kolloquium

Tag des Kolloquiums:
11 Dezember 2017

Dekan:
Prof. Frank-Olaf Schreyer

Prüfungsausschuss:
Prof. Dr. Raimund Seidel (Vorsitzender)
Prof. Dr. Michael Kerber
Prof. Dr. Kurt Mehlhorn
Dr. Frédéric Chazal
Dr. Antonios Antoniadis (Akademischer Beisitzer)

Berichterstatter:
Prof. Dr. Michael Kerber
Prof. Dr. Kurt Mehlhorn
Dr. Frédéric Chazal

Abstract

Persistent Homology is a tool to analyze and visualize the shape of data from a topological viewpoint. It computes *persistence*, which summarizes the evolution of topological and geometric information about metric spaces over multiple scales of distances. While computing persistence is quite efficient for low-dimensional topological features, it becomes overwhelmingly expensive for medium to high-dimensional features.

In this thesis, we attack this computational problem from several different angles:

- We present efficient techniques to approximate the persistence of metric spaces. Three of our methods are tailored towards general point clouds in Euclidean spaces. We make use of high dimensional lattice geometry to reduce the cost of the approximations. In particular, we discover several properties of the *Permutahedral* lattice, whose Voronoi cell is well-known for its combinatorial properties. The last method is suitable for point clouds with low intrinsic dimension, where we exploit the structural properties of the point set to tame the complexity. In some cases, we achieve a reduction in size complexity by trading off the quality of the approximation. Two of our methods work particularly well in conjunction with dimension-reduction techniques: we arrive at the first approximation schemes whose complexities are only polynomial in the size of the point cloud, and independent of the ambient dimension.

- On the other hand, we provide a lower bound result: we construct a point cloud that requires super-polynomial complexity for a high-quality approximation of the persistence. Together with our approximation schemes, we show that polynomial complexity is achievable for rough approximations, but impossible for sufficiently fine approximations.

- For some metric spaces, the intrinsic dimension is low in small neighborhoods of the input points, but much higher for large scales of distances. We develop a concept of local intrinsic dimension to capture this property. We also present several applications of this concept, including an approximation method for persistence.

This thesis is written in English.

Zusammenfassung

Persistent Homology ist eine Methode zur Analyse und Veranschaulichung von Daten aus topologischer Sicht. Sie berechnet eine topologische Zusammenfassung eines metrischen Raumes, die *Persistence* genannt wird, indem die topologischen Eigenschaften des Raumes über verschiedene Skalen von Abständen analysiert werden. Die Berechnung von Persistence ist für niederdimensionale topologische Eigenschaften effizient. Leider ist die Berechung für mittlere bis hohe Dimensionen sehr teuer.

In dieser Dissertation greifen wir dieses Problem aus vielen verschiedenen Winkeln an:

- Wir stellen effiziente Techniken vor, um die Persistence für metrische Räume zu approximieren. Drei unserer Methoden eignen sich für Punktwolken im euklidischen Raum. Wir verwenden hochdimensionale Gittergeometrie, um die Kosten unserer Approximationen zu reduzieren. Insbesondere entdecken wir mehrere Eigenschaften des *Permutahedral* Gitters, dessen Voronoi-Zelle für ihre kombinatorischen Eigenschaften bekannt ist. Die vierte Methode eignet sich für Punktwolken mit geringer intrinsischer Dimension: wir verwenden die strukturellen Eigenschaften, um die Komplexität zu reduzieren. Für einige Methoden zeigen wir einen Trade-off zwischen Komplexität und Approximationsqualität auf. Zwei unserer Methoden funktionieren gut mit Dimensionsreduktionstechniken: wir präsentieren die erste Methode mit polynomieller Komplexität unabhängig von der Dimension.

- Wir zeigen auch eine untere Schranke. Wir konstruieren eine Punktwolke, für die die Berechnung der Persistence nicht in Polynomzeit möglich ist. Die bedeutet, dass in Polynomzeit nur eine grobe Approximation berechnet werden kann.

- Für gewisse metrische Räume ist die intrinsiche Dimension gering bei kleinen Skalen aber hoch bei großen Skalen. Wir führen das Konzept lokale intrinsische Dimension ein, um diesen Umstand zu fassen, und zeigen, dass es für eine gute Approximation von Persistenz benutzt werden kann.

Diese Dissertation ist in englischer Sprache verfasst.

Acknowledgments

First and foremost, I would like to express my gratitude towards Michael Kerber, who advised the work presented in this thesis. Michael introduced me to the field of Computational Topology and helped me develop a footing in this area. In particular, I have learnt many problem solving techniques and ways to present one's work in a clear, concise manner from Michael. I thank Michael again for supporting me for innumerable academic and non-academic issues.

Next, I thank Kurt for accepting me as a PhD student in the highly vibrant Algorithms group, and for giving me his valuable guidance on various topics. I also thank Kurt for arranging me to visit Graz from time to time, which helped me immensely in continuing research with Michael. I additionally thank the Geometry group at TU Graz for their hospitality.

I thank Frédéric Chazal for agreeing to examine my thesis. I thank the anonymous reviewers who gave useful feedback for our papers; this has helped improve the content and presentation of this thesis considerably. I thank Christina and Ingrid for dealing with technical and administrative issues quickly and efficiently, including making several travel arrangements. I thank the graduate school for all their support. I thank colleagues and friends at MPI for making it a lively and stimulating place to work.

Finally, I want to thank my family for encouraging me to pursue higher academic studies, and for supporting me unwaveringly through pleasant and difficult times.

Contents

Abstract 3

Zusammenfassung 4

Acknowledgments 5

List of Figures 9

1 Introduction 11
1.1 Motivation 11
1.2 Summary of contributions 16
1.3 Related work 19
1.4 Outline of the thesis 19

2 Topological Concepts 21
2.1 Simplicial homology 21
2.2 Persistent homology 27
2.3 Stability of persistence modules 32
2.4 Additional concepts 36

3 Geometric Concepts 39
3.1 The A^* Lattice and the permutahedron 39
3.2 Shifted grids and cubes 50
3.3 Intrinsic dimension 53
3.4 Locality-sensitive hashing 56
3.5 Dimension reduction 59

I Techniques for Euclidean Spaces 61

4 Approximation using the Permutahedron 63
4.1 Approximation scheme 63
4.2 Computational aspects 66
4.3 A lower bound for approximation schemes 79
4.4 Discussion 84

5 Approximation using Grids 85
5.1 Approximation scheme 85
5.2 Computational complexity 90
5.3 Discussion 96

6 Digitization **97**
6.1 Approximation with cubical pixels 97
6.2 Discussion 103

II Techniques for Doubling Spaces **105**

7 Well-Separated Simplicial Decomposition **107**
7.1 Well-separated simplicial decomposition 107
7.2 Čech approximation 114
7.3 Discussion 118

8 Local Doubling Dimension **119**
8.1 Definition and net-forests 119
8.2 Applications 123
8.3 Discussion 126

9 Conclusion **127**

Bibliography **129**

A Strong Interleaving for Barycentric Scheme **137**

B Curriculum Vitae **139**

List of Figures

1.1 A point sample from a doughnut 12
1.2 A persistence diagram 13
1.3 A simplicial complex 13
1.4 An example of Čech and Rips complexes 14
1.5 A filtration 14
1.6 An Alpha complex 16
1.7 Matching of persistence diagrams 17

2.1 Čech and Rips complex 24
2.2 Cycles and Boundaries 26
2.3 Persistence diagrams and bottleneck matching 33

3.1 Hexagonal lattice 40
3.2 Permutahedra in 3D 41
3.3 Grid map 52
3.4 Barycentric subdivision 53
3.5 Covering for doubling dimension 54

4.1 An approximation complex in 2-D 64
4.2 An example of splits 69

5.1 Barycentric complex 87

6.1 Digitization 98
6.2 A claim about permutahedra 104

7.1 A well-separated tuple 108
7.2 WSSD construction 110
7.3 A covering property of Rel-sets 112

8.1 An almost space-filling curve 120
8.2 Disparity between Euclidean and Geodesic distances 121

Chapter 1

Introduction

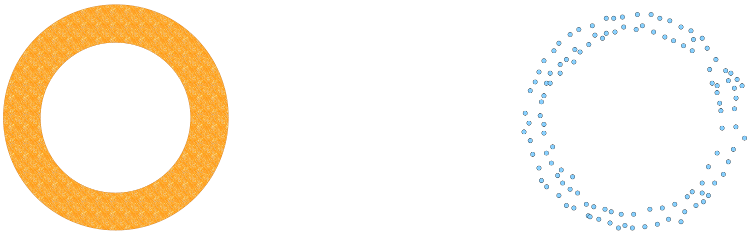
1.1 Motivation

One of the most common buzzwords of our times is *data*. Every conceivable process and object in nature, every action and thought has a wealth of information associated to it. Some of these can usually be represented in computer-readable formats and with suitable processing techniques, one can analyze the data to reveal useful insights. The acquired knowledge can later be utilized for innumerable objectives, some of which include scientific research and commercial interests. The infinite corpus of available data and its practical importance demand efficient techniques for analyzing data.

Often, it has been observed that data has *shape*. This encourages the need to apply geometric and topological lenses to data analysis in order to understand and interpret the shape of information. This has given rise to a fast emerging field called *Topological Data Analysis* (abbreviated as TDA), which aims at extracting and analyzing geometric and topological properties of data, so as to make the underlying information apparent and accessible. Often in practice, data is presented in the form of a metric space, where each data point has some characteristic property associated to it, and there is a notion of distance between data points. Perhaps the simplest case is that of a set of points in Euclidean space, with the usual notion of distance. TDA encompasses a variety of techniques to analyze metric spaces. We will concentrate on the main component, which took shape near the beginning of the twenty-first century and has had a profound effect on the development of this field.

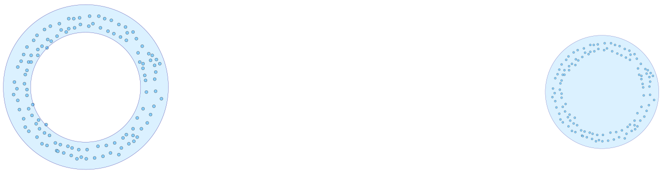
Since data usually has an inherent notion of distance associated to it, it is natural to talk about the shape of information at a given distance parameter. An elementary example could be a point sample from a doughnut: at very low distances, the points seem to be isolated, each in its own cluster. At medium distances, it becomes more clear that the points have some shape. Finally, at large distances, the point sample seems to come from a single cluster and does not seem to represent the original shape. See Figure 1.1 to get an intuition. We usually call this distance parameter as the *scale*.

Homology is a mathematical notion, which can be used to describe the shape of objects in some contexts. For a given shape, it describes the number of connected components, the number of loops, voids, tunnels...(and higher-dimensional analogous topological features) in the shape, using concepts from abstract algebra. In other words, it characterizes a shape using the *holes* present in the shape. In some sense, the homology of a shape depends on the scale of observation. As an example, for the doughnut-sample, homology can describe the shape for any given distance parameter. However, it is usually unclear which scale parameter to choose: the true shape (which is a loop) is apparent only when the scale



(a) The original shape.

(b) At low scales, points are isolated.



(c) At medium scales, the shape of the loop is apparent.

(d) At large scales, all points are in one cluster, so the loop disappears.

Figure 1.1: A point sample from a doughnut, interpreted at different scales of distances.

is chosen in the correct range (Figure 1.1). Moreover, practical data is usually noisy, so at any given scale several loops can be apparent from the point sample, which do not represent the correct shape.

The natural solution is to not focus on a particular scale, but rather a multitude of scales, and to observe the evolution of these topological features as the scale changes. Ideally, this analysis should show that the important features *persist*, that is, they should have long ”lifetimes” over the scales, and noisy features should have short lifetimes. The formal notion for this intuition was introduced by Edelsbrunner, Letscher and Zomorodian in their landmark paper [ELZ02]. They coined the term *Persistent Homology*, which represents the *persistence* of topological features in the homology over multiple scales. They introduced a fast algorithm (which we call the *persistence algorithm*) to compute a topological summary called a *persistence diagram*: this is a collection of points in $\mathbb{R}^2$, each representing the evolution of some topological feature. The co-ordinates of the points represent the scales at which the features appeared and disappeared. For more details, see Figure 1.2.

The persistence algorithm utilizes a collection of discrete structures called *simplices*. A k-dimensional simplex is the convex hull of $(k+1)$ affinely independent points in Euclidean space, which means that it is a polytope of dimension k (perhaps the *simple*~~*x*~~st polytope). A collection of simplices which intersect only in their common faces is commonly called a *simplicial complex*; a simple example is shown in Figure 1.3. We briefly describe two types of simplicial complexes built on point clouds in Euclidean spaces. Most of the concepts however, extend to arbitrary metric spaces.

To apply the tools of persistent homology, the first step is to construct simplicial complexes on the point set. Let P be a finite set of $n > 0$ points in d-dimensional Euclidean space $\mathbb{R}^d$, and let $\alpha \geq 0$ be any scale. Consider the set of Euclidean balls of radius α, centered at the points of P. Whenever there is a common intersection between $(k+1)$ balls, we add a k-simplex on those $(k+1)$ points of P. That means, whenever two balls intersect, we add an edge; whenever three balls intersect, we add a triangle, and so on. The resulting structure is a simplicial complex, commonly known as the *Čech* complex at scale α. See Figure 1.4.

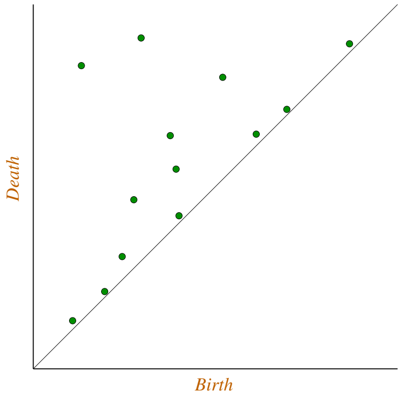


Figure 1.2: A persistence diagram: each point in the diagram corresponds to a topological feature. The x co-ordinate represents the scale at which it appeared (the *birth* scale) and the y co-ordinate represents the scale at which it became disappeared (the *death* scale). Points which are far from the diagonal $y = x$ (shown in the figure) have large persistence, and are typically considered to be important features of the data. On the other hand, noisy features only survive for a short range of scales, and are represented by points close to the diagonal. So, it is easy to identify the important features of data without being misled.

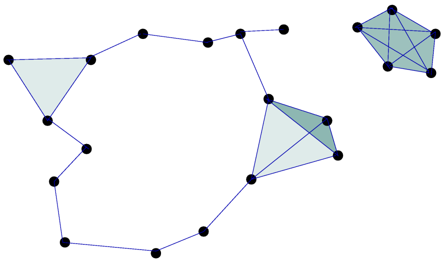
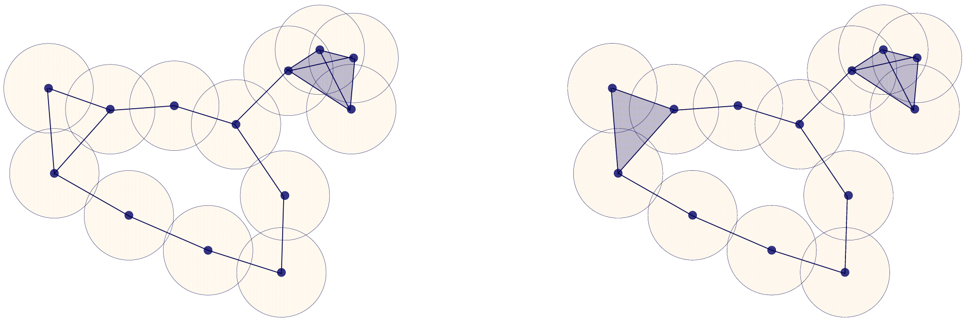


Figure 1.3: A simplicial complex: the 0-simplices are the dark vertices, the 1-simplices are the edges, the 2-simplices are triangles, and so on...

It is well-known that the Čech complex and the union of balls defining it have the same homology. This makes it easy to interpret the homology of any union of balls from the corresponding Čech complex, which is easier to manipulate. A relative of the Čech complex is the simplicial complex which goes by the name of *Vietoris-Rips complex* at scale α, commonly called simply the *Rips complex*: the simplices of this complex are sets of points with diameter at most 2α. Another interpretation is that there is a k-simplex in the Rips complex, if the $(k + 1)$ corresponding α-balls pairwise intersect. See Figure 1.4 for an example in $\mathbb{R}^2$.

Both the Čech and the Rips complexes are commonly applied in TDA. However, the latter is more suited for arbitrary metric spaces which makes it more attractive. At any given scale, both simplicial complexes have homology groups which give information about the underlying point cloud. To visualize the persistence of topological features, the scale α is varied from 0 to ∞. This gives a sequence of nested simplicial complexes called a *filtration*. A simple example of the Rips filtration is shown in Figure 1.5. The persistence algorithm takes as input a filtration and outputs the persistence diagram.

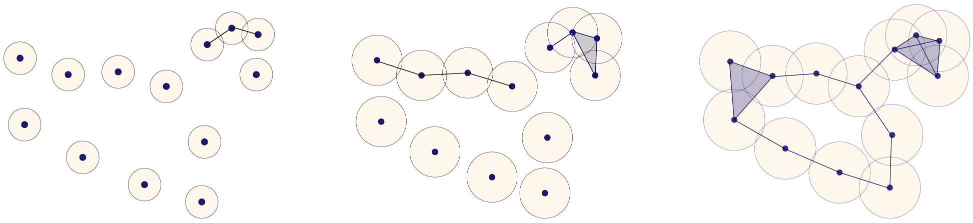
Real world data often includes some amount of noise, usually because of imprecision in measurement or representation. One of the most attractive properties of persistent



(a) Čech complex on the set of input points (dark dots) at a certain scale.

(b) Rips complex, at the same scale.

Figure 1.4: An example of Čech and Rips complexes: the Rips complex is a superset of the Čech complex at the same scale.



(a) Rips complex at some scale

(b) ..at a larger scale.

(c) ...and at an even larger scale.

Figure 1.5: A Rips filtration.

homology, which has led to a proliferation in practical applications, is its stability to noise. For small perturbations in the input data, the corresponding persistence diagram also changes slightly. As a result, the important features of data, which remain mostly unaffected with small perturbations, are still captured in the diagram with large persistence.

Applications Persistent homology (PH) has been used for a wide variety of applications, including those where the input data was of geometric nature, as well as applications where it was not apparent at first that the data had shape. We mention a few results in this area. PH has been used to solve coverage problems in sensor networks [dSG06] and to understand the structure of complex networks [HMR09, PSDV13]. It has been used to cluster point clouds [CGOS13], to recognize shapes from images [BOOC16], and to measure the dimension of fractal shapes [MS12]. In astrophysics, persistent homology has been used to understand the structure of cosmic matter [PEvdW$^+$16, SPK11, Sou11]. There have been many applications of persistent homology in biology. In [PET$^+$14], the influence of drugs on brain networks was studied. The geometry of neural networks in the brain and the effect of stimuli was recently explored in [RNS$^+$17]. The effect of mixing of genetic material in evolution was studied in [CCR13]. PH has been used to study the organization of biological tissues in [JRVM$^+$17]. In [BEK10], the growth of plant root systems was studied. For diagnostics, it has been applied for Autism [CBK09] and for classification of Hepatic lesions [ARC14]. In the field of material science, PH has been used to study the phase transition of glass [KFH16, NHH$^+$15]. It has been used to study crystallized packings in [STR$^+$17].

Challenges Since a filtration consists of a nested sequence of simplicial complexes, the last complex contains the largest number of simplices. The size of a filtration is the number of simplices in the last complex. If M is the size of a filtration, the persistence algorithm takes $O(M^3)$ time to compute the persistence diagram. In practice, it has been observed that the algorithm takes near-linear time in M, which makes it efficient for reasonable values of M.

Unfortunately, M can be very large in practice. For instance, at high scales the balls defining the Čech complex can have many intersections among themselves: every $(k+1)$-subset of balls may intersect, which means that the Čech complex can have up to $n^{O(k+1)}$ k-simplices. The *k-skeleton* of a complex is a subset which contains simplices of dimension up to (and including) k. Naturally, the k-skeleton of the Čech complex can have size $n^{O(k)}$. The same problem occurs for the Rips complexes. While this is tractable for small k, such as the 1-skeleton, going higher in dimensions leads to a combinatorial explosion in the complex size. This makes computing the persistence impractical even for medium dimensional homology, and restricts its practical applications. A common trick is to cap the construction of the filtration at a scale till which computation is feasible. However, this method runs the risk of missing out on important topological features, since they may appear at scales higher than the capped limit.

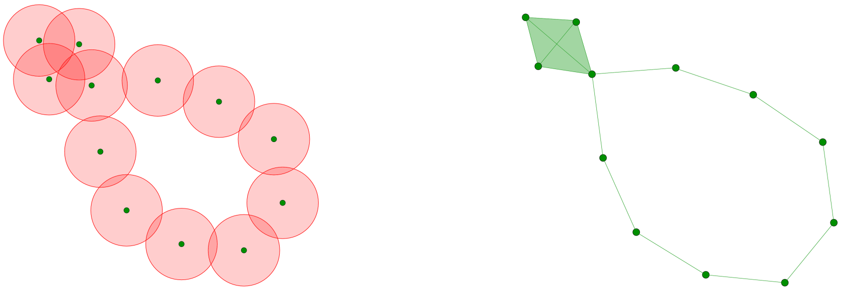
In two and three dimensions, an efficient alternative to the Čech complex is the *Alpha* complex [EH10], which is a subset of the Delaunay triangulation and is defined in a slightly different manner from the Čech complex. Instead of looking directly at the balls defining the Čech complex, we restrict the balls to the individual Voronoi regions of the input points. Then, each non-empty $(k+1)$-wise intersection of the restricted balls gives a k-simplex in the Alpha complex. See Figure 1.6 for an illustration. The Alpha complex has the same homology as the Čech complex; moreover, because the intersections between the balls is restricted, it is a subset of the Čech complex. Alpha complexes can be used to build a filtration and compute persistence. Unfortunately, the size of the Alpha filtration is as large as $n^{O(\lfloor d/2 \rfloor)}$, which is tractable for low dimensions, but faces the same obesity issues as the Čech and Rips filtrations in higher dimensions.

An approach to tackle this issue is to try and compute an approximation of the persistence diagram instead. The approximation is in the sense that there is a partial matching between the original and the approximate diagrams, where points with significant persistence and similar lifelines are matched together. See Figure 1.7 for a more pictorial description. A close matching implies that important topological features are well-represented in the approximate version, so that the crucial features of the data are still accessible and distinguishable from noise. The obvious requirement is that computing the approximation should be much more efficient that the original.

To compute the approximation, a collection of simplicial complexes $\{K_i \mid i \in I\}$ is constructed over a set of scales I. At each scale, these complexes approximately represent the topology and geometry of the original complex. There are homology-preserving maps $\{\theta_i \mid i \in I\}$ between the approximate complexes. Such a collection is called a *tower* and can be represented in the form:

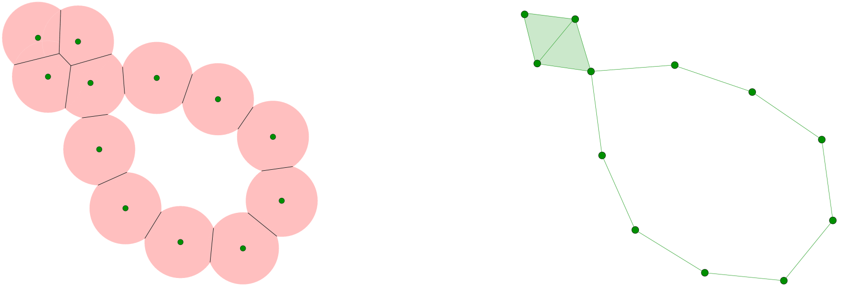
$$K_1 \xrightarrow{\theta_1} K_2 \xrightarrow{\theta_2} K_3 \xrightarrow{\theta_3} \dots \xrightarrow{\theta_{m-1}} K_m.$$

Typically, θ_i maps simplices of K_i to simplices of K_{i+1}, in which case it is known as a *simplicial* map. There are efficient algorithms [DFW14, KS17] to compute the persistence diagram of a tower which is connected with simplicial maps. For efficiency, it is desirable that the size of the tower should be small. Unlike filtrations, a tower's size is not determined by the largest complex in the sequence, since the map θ can contract simplices to make them



(a) The union of balls.

(b) ...and the corresponding Čech complex.



(c) Balls restricted to the Voronoi regions. The Voronoi boundaries are shown in black (dark).

(d) ...the corresponding Alpha complex.

Figure 1.6: The Čech complex and the Alpha complex at the same scale.

disappear. Rather, the number of simplices which are added to the sequence determines the size of the tower.

1.2 Summary of contributions

We present several results pertaining to the construction of approximation towers. Some of our techniques are tailored to work with Euclidean spaces, which we discuss in the first part of the thesis. Further techniques for the special case of point clouds which have low intrinsic dimensions are discussed in the second part of the thesis. The central idea behind computing approximation towers is to either sparsify the point cloud using some coarsening technique, or (perhaps counter-intuitively) in some cases, use additional Steiner points.

The first three contributions use techniques from lattice geometry in Euclidean space. In the first two, we build the approximation complexes using the set of lattice points closest to the input points.

- In **Chapter 4**, we present our first approximation scheme for Rips filtrations in $\mathbb{R}^d$. Our scheme yields a $O(d)$-approximation of the Rips filtration. The k-skeleton of the tower has size $n2^{O(d\log k)}$ per scale, and $n2^{O(d\log d)}$ over all scales. The construction is based on a tessellation of the Euclidean space using Voronoi polytopes of the A_d^* lattice [CSB87]; this polytope is more commonly known as the *Permutahedron*. We present several new results about the geometry and topology of the permutahedral tessellation, that are useful in proving our result.

  We also present a lower bound result on the size of approximation towers. We construct a set of n points in $\Theta(\log n)$ dimensions, such that it gives rise to $n^{\Omega(\log\log n)}$

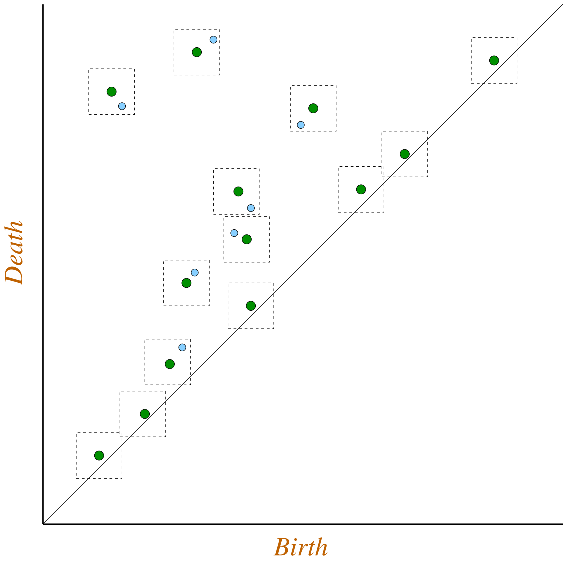


Figure 1.7: The green points (large) come from the original persistence diagram, while the blue ones (small) are from the approximation. Green points are matched to blue points, which are contained in the square centered at the green points. The size of the square determines the approximation ratio. The green points close to the diagonal can be unmatched.

features of non-insignificant persistence. More specifically, we show that for any $0 < \varepsilon < \frac{1}{\log^{1+c} n}$, $c \in (0,1)$, any $(1+\varepsilon)$-approximation of the Čech filtration of this point set has at least that many points in its persistence diagram, which implies that it is impossible to get a polynomial-sized approximation for this input instance.

The contents of this chapter are based on [CKR16] and [CKRb].

- In **Chapter 5**, we improve upon our approximation result of Chapter 4. We present an approximation scheme which has only $n2^{O(d \log k)}$ simplices in the tower for the k-skeleton in the worst case, and 2-approximates the Rips filtration in the metric of the L_∞ norm. This in turn yields a $2d^{1/4}$-approximation for the Rips filtration in the usual L_2 metric. Our construction is based on the *barycentric subdivision* of the grid lattice. We arrive at our approximation result by the use of Steiner vertices, which helps in reducing the size complexity.

  Further, there are two novel techniques which we use in this chapter. The first is the use of *acyclic carriers* for proving our approximation result. In our application, these are maps which relate the Rips complex and the approximation in a relatively simple manner and greatly reduce the complexity of showing the approximation result. Also, they allow for more freedom in designing towers compared to previously known methods. The second technique is what we refer to as *scale balancing*, which is a simple trick to improve the approximation ratio under certain conditions.

  The results in this chapter are based on our paper [CKR17].

For high-dimensional point clouds, a common way to reduce the dimension is to embed them into a lower-dimensional space, while approximately preserving some desirable properties, such as the set of pairwise distances. Two techniques in this area are the well-known Johnson-Lindenstrauss Lemma [JLS86], and its generalization by Matoušek [Mat90]. We apply these dimension-reduction techniques on the input, and then proceed to approximate the filtration of the modified point cloud. With this pipeline, the aforementioned schemes give towers of size $n^{O(1)}$, which is completely independent of the original ambient dimension. These are the first results to achieve true dimension-independent polynomial-sized approximations.

The take-away message from these results is that for relatively rough approximations, it is possible to have true polynomial complexity, but it is impossible for very fine approximations.

Chapter 6 presents an alternative to the above approaches. It relies on digitizing the Euclidean space with very fine cubes (*pixels*). With this approach, we can approximate the Euclidean balls which define the Čech complex with pixels; our approximation complex is built using these pixels. We give a $(1+\varepsilon)$-approximation of the Čech filtration, at the cost of having $n(1/\varepsilon)^{O(d)}2^{O(dk)}$ simplices in the k-skeleton, at any scale in the tower, for any desired $0 < \varepsilon \leq 1$. Compared to the permutahedra-based and grid-based schemes, this gives a much better approximation guarantee, which is independent of the ambient dimension. On the other hand, the complexity is worse, although it is still an improvement over previous work (see Section 1.3). The results of this chapter are based on [CKRa].

In practice, it is often the case that data sets come from low-dimensional metric spaces residing in high dimensional ambient spaces. A common way to capture this structure is the *doubling dimension*, which is a notion of intrinsic dimension of any metric space. It is desirable that algorithms to manipulate data sets with low doubling dimension should have complexities dependent primarily on the doubling dimension. In such cases, using the aforementioned techniques to approximate persistence diagrams for such data sets would be sub-optimal, since the overall complexity is strongly dominated by the ambient dimension. We address this issue in the second part of the thesis, from two different angles:

- A *well-separated pair decomposition* [CK95] is a well-known data structure which approximately captures all pairwise distances of a point set, using only linear space in n. In [KS13], this concept was generalized to approximately represent all simplices on the point set, using only linear space; they call the data structure a *Well-separated simplicial decomposition* (WSSD). Further, using WSSDs, they provided an approximation scheme for Čech filtrations. They used a *quadtree* [HP11] for their construction, which unfortunately does not capture the intrinsic dimension. In **Chapter 7**, we extend their construction to spaces with low doubling dimension. We present a scheme to construct a WSSD whose size is linear in n and exponential in the doubling dimension Υ. We present an $(1+\varepsilon)$-approximation tower for the Čech filtration, whose size is at most $n(1/\varepsilon)^{O(\Upsilon^2)}$. We use a data structure which correctly captures the intrinsic dimension of a space: the *net-tree*.

  The contents of this chapter are based on [CKS].

- In some applications, information is only relevant over a range of scales. The point set may have a much lower intrinsic dimension in that range, than when looking at all scales of distances. To model this setting, in **Chapter 8** we introduce the concept of intrinsic dimension on restricted scales, which we call the *local doubling dimension*. To use this idea in an algorithmic context, we present an algorithm to construct a net-forest, which is the part of the net-tree restricted to the range of relevant scales. This is more efficient than pruning the net-tree. We apply the net-forest to several applications of relevance, including a Čech approximation built up to a desired scale, based on Chapter 7.

  This chapter is based on [CK15].

1.3 Related work

The concept of persistent homology was first introduced by Edelsbrunner and others in the paper [ELZ02]. They also provided a fast algorithm for computing persistence diagrams for filtrations. The concept and the algorithm were later extended by Carlsson and Zomorodian in [CZ05]. Further optimizations in the algorithm were introduced in [BKR14, CK11] and it was later employed in a distributed setting [BKRW17]. The dual of persistent homology, known as *persistent co-homology*, was introduced as an alternative to compute persistence diagrams in [dSMVJ11b], and algorithms were introduced in [BDM15, dSMVJ11a]. The concept of *zig-zag* filtrations, where inclusions can occur in both directions was introduced in [CdS10], and algorithms were presented in [CdSM09, MMv11]. Recently, efficient algorithms for computing barcodes for simplicial towers have been presented in [DFW14, KS17]. Also, persistent homology has recently been applied to non-metric spaces [EW17].

The stability of persistence diagrams is one of the main results in the field of persistent homology. The first result of this kind was shown in [CSEH07] for the notion of *bottleneck distance* between persistence diagrams. This was further extended to a wider class of distance measures in [CSEHM10]. The result was later generalized to arbitrary *persistence modules*, abstracting from the geometric setting in [CCSG$^+$09].

The first approximation scheme for filtrations was given by Hudson et. al in [HMOS10]. For the *Alpha-filtration* of n points $P \subset \mathbb{R}^d$, they give a constant-factor approximation tower of size $n2^{O(d^2)}$. They used the Delaunay triangulation of a superset of P to construct the approximation. The first approximation scheme for Rips filtrations was given by Sheehy [She13]. For any desired $0 < \varepsilon < 1$, he constructs a $(1+\varepsilon)$-approximate tower of the Rips filtration of arbitrary metric spaces. The approximation tower of [She13] consists of a filtration of simplicial complexes, whose k-skeleton has size only $n(\frac{1}{\varepsilon})^{O(\lambda k)}$, where λ is the doubling dimension of the metric. This is a significant improvement over the result of [HMOS10], since the approximation factor can be chosen at will, and the size of the approximation is reduced when considering skeletons of low size. Dey et. al gave an approximation scheme for the Rips filtrations [DFW14] of similar size, along with an algorithm to compute persistence for simplicial towers. They used vertex collapses to reduce the size of the approximation, resulting in a simplicial tower. This is in contrast to [HMOS10] and [She13], whose approximations consisted of filtrations. The scheme was later optimized in [DSW16], to give significant empirical gains over [DFW14] and [She13]. Similar approximation schemes have also been proposed for Čech filtrations [BS15, CJS15, KS13], all with comparable size bounds.

In a different direction, there has been work aiming at reducing the dimension of the point cloud, prior to applying approximation schemes. In analogy to the famous Johnson-Lindenstrauss Lemma [JLS86], it has been shown that an orthogonal projection of P to an $O(\log n/\varepsilon^{O(1)})$-dimensional subspace yields a $(1+\varepsilon)$-approximation of the Čech filtration [KR15, She14], for $0 < \varepsilon < 1$. It is possible to combine dimension reduction with the aforementioned approximation schemes. However, this yields an approximation of size $n^{O(k+1)}$ (ignoring factors of ε) and does not improve upon the exact case.

The above approaches work well for instances where λ and k are small. However, even for medium-sized dimensions and homology, these are prohibitively expensive.

1.4 Outline of the thesis

We start the technical presentation in Chapter 2 by discussing the topological preliminaries. First, we formally discuss the concept of homology for simplicial complexes. With this base, we then explore in length the concept of persistent homology, the central target of this thesis.

Geometric concepts and results are presented in Chapter 3. We start by discussing properties of the A^* lattice (which is later used in Chapter 4), and briefly touch upon some properties of the grid lattice. We further discuss the concepts of doubling dimension, net-trees, pair decompositions and dimension reduction, among others.

The first part of the thesis comprises of Chapters 4 to 6. In Chapter 4 we present the approximation scheme based on the permutahedral lattice, along with the lower bound result for Čech filtrations. We present the approximation scheme based on grid lattices in Chapter 5. Chapter 6 presents an alternative approach based on digitization of Euclidean space.

The second part of the thesis comprises of Chapters 7 and 8 and focuses on point clouds with low doubling dimension. In Chapter 7, we present techniques to construct WSSDs and an approximation method for Čech filtrations. Chapter 8 introduces the notion of local intrinsic dimension and applies it to several applications.

Chapter 9 concludes the presentation.

Chapter 2

Topological Concepts

We start the exposition by providing an overview of the topological concepts used in this thesis. For a more detailed exposure, we direct the reader to the books [EH10, Hat02, Mun84] and articles [BdSS15, Car09, ELZ02] in this area. We assume that the reader is familiar with basic topological concepts. However, for the sake of completeness, we repeat some basic concepts.

We begin by discussing the fundamental concepts of simplicial complexes and the topology carried by such structures in Section 2.1. Then, in Section 2.2 we present the concept of persistent homology which represents the topological information carried by special collections of simplicial complexes. We give details about the topological summary obtained from the pipeline of persistent homology and discuss its computational aspects. An important idea which we use throughout the thesis concerns with the similarity of persistence modules, and is presented in Section 2.3. We end the chapter by discussing some additional concepts in Section 2.4.

Apart from the well-studied topological concepts, we discuss two new contributions, one in Sub-section 2.3.3 and Sub-section 2.4.2.

2.1 Simplicial homology

In this section we discuss about simplicial complexes and their homology they carry. We first discuss a few fundamental concepts related to the similarity of topological spaces.

2.1.1 Topological similarity

We discuss the strongest notion of similarity between topological spaces:

Definition 2.1.1 (homeomorphic spaces). *Two topological spaces X and Y are said to be* homeomorphic *if there exists a bijection $f : X \to Y$ such that f and f^{-1} are continuous functions. We call the map f a* homeomorphism *from X to Y.*

An alternative and equivalent definition is: X and Y are said to be homeomorphic if there exist two continuous maps $f : X \to Y$ and $g : Y \to X$ such that $f \circ g = ID_Y$ and $g \circ f = ID_X$, where ID_X, ID_Y denote the identity maps of X and Y, respectively. Homeomorphic spaces are topologically equivalent to each other.

In general, determining whether two spaces are homeomorphic is a difficult problem. We turn to a weaker form of topological similarity.

Definition 2.1.2 (homotopy). *Let $f_1, f_2 : X \to Y$ be two continuous maps between topological spaces X, Y. A continuous function $H : X \times [0,1] \to Y$ is said to be a* homotopy *between f_1 and f_2, if $H(x,0) = f_1(x)$ and $H(x,1) = f_2(x)$. In this case, f_1 and f_2 are said to be* homotopic *to each other, and we record this relation as $f_1 \overset{h}{\simeq} f_2$.*

Informally, the second parameter of H can be interpreted as time, so that H describes a continuous deformation of f_1 into f_2, as time varies from 0 to 1.

Definition 2.1.3 (homotopy equivalence). *X and Y are said to be* homotopy equivalent *if there exist continuous maps $f : X \to Y$ and $g : Y \to X$ such that $f \circ g \overset{h}{\simeq} ID_Y$ and $g \circ f \overset{h}{\simeq} ID_X$. We denote this relation as $X \overset{h}{\simeq} Y$.*

Intuitively, two spaces are homotopy equivalent if they can be continuously transformed into one another. In comparison to the conditions of homeomorphisms, the condition that f and g are inverses of each other is relaxed. Instead, homotopy equivalence only requires that the compositions of the two functions be homotopic to the identity maps. As such, homotopy equivalence is a weaker form of homeomorphism. Two spaces which are homeomorphic are necessarily homotopy equivalent, but the converse is not true. A special case of homotopy equivalence arises when $Y \subseteq X$.

Definition 2.1.4 (deformation retract). *Let $Y \subseteq X$ and $f : X \to Y$ be a continuous map such that $f(y) = y$, $\forall y \in Y$, and $f \overset{h}{\simeq} ID_X$. Then, $X \overset{h}{\simeq} Y$ and Y is said to be a* retract *of X. The map f is a* deformation retract *realizing the homotopy equivalence.*

Definition 2.1.5 (contractability). *A space is said to be* contractible*, if it is homotopy equivalent to a point.*

2.1.2 Simplicial complexes

We will mostly deal with topological spaces which can be represented discretely. We define the building blocks of these spaces next.

Definition 2.1.6 (simplex). *Given a finite collection of elements S, an* (abstract) simplex *σ on S is simply a non-empty subset $\sigma \subseteq S$. The dimension of σ is $dim(\sigma) := |\sigma| - 1$. A* face *of σ is a subset $\tau \subseteq \sigma$. A* facet *of σ is a face of co-dimension 1. A* proper face *of σ is a face of co-dimension one or higher.*

A common example of simplices are those which have a geometric realization:

Definition 2.1.7 (geometric simplex). *A (non-degenerate)* geometric k-simplex *is the convex hull of a set of $(k+1)$-points $\{p_0, \ldots, p_k\} \subset \mathbb{R}^d$ such that the vectors $(p_j - p_0)$ are affinely independent, for $1 \leq j \leq k \leq d$.*

From the definition, it is clear that a 0-simplex is a vertex, a 1-simplex is an edge between two vertices, a 2-simplex is a triangle, and so on. From the definition, we see that a k-simplex has $(2^{k+1} - 2)$ proper faces. We next look at a special collection of simplices.

Definition 2.1.8 (simplicial complex). *A simplicial complex K on a finite set of elements S is a collection of simplices on S such that:*

- *For each simplex $\sigma \in K$ and each face $\tau \subset \sigma$, it holds that $\tau \in K$.*
- *For any two simplices $\sigma, \delta \in K$, $\sigma \cap \delta$ is either empty or a common face of both σ and δ.*

A sub-complex *K' (of K) is a simplicial complex that satisfies $K' \subseteq K$.*

The dimension of a simplicial complex K, $dim(K)$ is the maximum dimension of any simplex in K. It is known that any abstract simplicial complex of dimension k has a geometric realization in $\mathbb{R}^{2k+1}$ [EH10].

Definition 2.1.9 (skeleton). *The* i-skeleton *of a simplicial complex* K *is the maximal subcomplex* $K^i \subseteq K$ *such that each simplex of* K^i *has dimension at most* i*, for* $0 \leq i \leq dim(K)$*.*

Naturally, K^0 consists of just the vertices of K. Then, K^1 is a graph on the vertices of K, and its edges are the 1-simplices of K.

Definition 2.1.10 (flag complex). *A simplicial complex* K *is a* flag complex*, if for each subset of vertices* $\{p_0, \ldots, p_k\} \subseteq K^0$ *which has the property that each 1-simplex* (p_i, p_j) *is in* K^1*, it holds that the* k*-simplex* $(p_0, \ldots, p_k)$ *is in* K*.*

As a consequence, a flag complex is completely determined by its 1-skeleton. Also, a flag complex is the maximal simplicial complex among the set of complexes having the same 1-skeleton.

Nerve Theorem

Let $U := \{U_1, \ldots, U_m\}$ denote a finite collection of sets.

Definition 2.1.11 (nerve). *The nerve of* U *is an abstract simplicial complex* nerve(U) *consisting of simplices corresponding to non-empty intersections of elements of* U*, that is,*

$$\text{nerve}(U) := \{V \subseteq U \mid \cap_{U_i \in V} U_i \neq \varnothing\}.$$

In other words, for every $(k+1)$-wise intersection among the sets, there is a k-simplex in the nerve. While U can be of arbitrary form, nerve(U) is a discrete object. Under certain conditions, some topological properties of a space can be captured from its nerve.

Theorem 2.1.12 (nerve theorem). *Let* $U := \{U_1, \ldots, U_m\}$ *be a collection of closed sets in Euclidean space, such that all intersections of the form*

$$\{V \subseteq U \mid \cap_{U_i \in V} U_i \neq \varnothing\}$$

are contractible. Then, $\text{nerve}(U) \overset{h}{\simeq} \cup_{i=1}^{m} U_i$ *[Bjö95, Bjö03, Bor48, Wal81].*

In particular, if U is a collection of convex sets, then all non-empty common intersections are contractible, so that the nerve theorem applies to U. We now discuss about two well-known simplicial complexes.

Definition 2.1.13 (Čech complex). *Let* $P \subset \mathbb{R}^d$ *be a finite set of points. Let* $B_\alpha(x)$ *denote the Euclidean ball of radius* $\alpha \geq 0$ *centered at a point* $x \in \mathbb{R}^d$*. The* Čech complex *at scale* α *is the nerve of the collection of balls* $(B_\alpha(p))_{p \in P}$*, that is,*

$$\mathcal{C}_\alpha = \{\sigma \subseteq P \mid \cap_{p \in \sigma} B_\alpha(p) \neq \varnothing\}.$$

See Figure 2.1 for an example. Since Euclidean balls are convex objects, it follows from Theorem 2.1.12 that $\mathcal{C}_\alpha \overset{h}{\simeq} \cup_{p \in P} B_\alpha(p)$, that is, the Čech complex on P at scale α is homotopy equivalent to the union of balls of radius α centered at the points of P.

Let meb(Q) denote the minimum enclosing ball of any finite set of points $Q \subset \mathbb{R}^d$. Let rad(Q) denote the radius of meb(Q). An alternative and equivalent definition of the Čech complex can then be obtained as

$$\mathcal{C}_\alpha = \{\sigma \subseteq P \mid \text{rad}(\sigma) \leq \alpha\}.$$

Let $diam(Q)$ denote the diameter of Q. We define a related complex to the Čech complex:

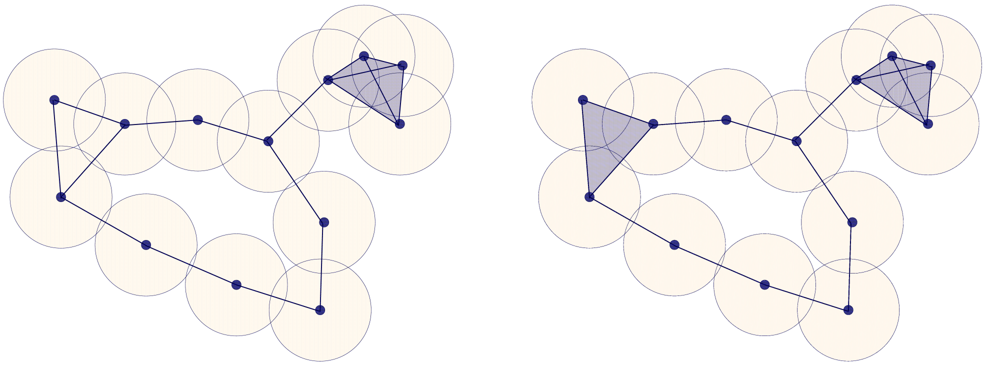


Figure 2.1: Left: Čech complex. Right: Rips complex, at the same scale (the same figure was also used in Chapter 1).

Definition 2.1.14 (Vietoris-Rips complex). *Let* $P \subset \mathbb{R}^d$ *be a finite set of points. The* (Vietoris-)Rips complex *on* P *at a scale* $\alpha \geq 0$ *is defined as*

$$\mathcal{R}_\alpha = \{\sigma \subseteq P \mid diam(\sigma) \leq 2\alpha\}.$$

An example is shown in Figure 2.1. By definition, the Rips complex is a flag complex, so it is completely determined by its 1-skeleton. In fact, the 1-skeleton of $\mathcal{R}_\alpha$ is the intersection graph of the balls $B_\alpha(p)$ defining the Čech complex, meaning that there is an edge between two points of P if their distance is at most 2α. Therefore, the 1-skeletons of $\mathcal{C}_\alpha$ and $\mathcal{R}_\alpha$ are the same. The Rips complex can be obtained from the Čech complex by a simple method: by adding simplices to $\mathcal{C}_\alpha$ so that it becomes a flag complex, that is $\mathcal{R}_\alpha$.

Lemma 2.1.15. *For any* $\alpha \geq 0$, $\mathcal{C}_\alpha \subseteq \mathcal{R}_\alpha \subseteq \mathcal{C}_{\sqrt{2}\alpha}$.

Proof. For the first relation, let $\sigma \in \mathcal{C}_\alpha$ be any simplex. This means that $\mathrm{rad}(\sigma) \leq \alpha$. Since $diam(S) \leq 2\mathrm{rad}(S)$ for any compact set $S \subset \mathbb{R}^d$, it follows that $diam(\sigma) \leq 2\alpha$. This shows that $\sigma \in \mathcal{R}_\alpha$.

For the second relation, let $\sigma \in \mathcal{R}_\alpha$ be any simplex so that $diam(\sigma) \leq 2\alpha$. Jung's theorem [Jun01] states that

$$\mathrm{rad}(S) \leq diam(S)\sqrt{\frac{d}{2(d+1)}} \leq diam(S)/\sqrt{2}$$

for any compact set $S \subset \mathbb{R}^d$. So,

$$\mathrm{rad}(\sigma) \leq diam(\sigma)/\sqrt{2} \leq (2\alpha)/\sqrt{2} \leq \sqrt{2}\alpha$$

which implies that $\sigma \in \mathcal{C}_{\sqrt{2}\alpha}$. ■

We remark that the definition of Rips complexes does not require P to be a geometric point set: the Rips complex can be defined for any metric space. In particular, for the metric space induced by the L_∞-norm in $\mathbb{R}^d$, we denote the Rips complex at scale α by

$$\mathcal{R}^\infty_\alpha = \{\sigma \subseteq P \mid diam_\infty(\sigma) \leq 2\alpha\},$$

where $diam_\infty$ denotes the diameter in the L_∞-metric.

2.1.3 Homology of simplicial complexes

We discuss the homology carried by a simplicial complex K. First, we discuss a few preliminaries.

Definition 2.1.16 (chains). *A* p-chain *c of K is a formal sum of p-simplices of K. Specifically, c is of the form $c = \sum a_i\sigma_i$, where $\sigma_i \in K^p \setminus K^{p-1}, \forall i$, the coefficients a_i come from some field $\mathbb{F}$ and $0 \leq p \leq dim(K)$ is an integer.*

In this thesis, we use the field $\mathbb{Z}_2$ to define the coefficients. We add two p-chains $c_1 = \sum a_i\sigma_i$ and $c_2 = \sum b_i\sigma_i$ linearly : $c_1 + c_2 = \sum(a_i + b_i)\sigma_i$.

Definition 2.1.17 (chain group). *The set of p-chains of K along with the addition operator is an abelian group $C_p(K)$, called the* p-th chain group *of K.*

It is easy to verify that $C_p(K)$ is a group. The 0-element of $C_p(K)$ is the empty chain 0. The inverse of any chain c is itself, since $c + c = 0$ due of addition over $\mathbb{Z}_2$. Also, the definition of addition over $\mathbb{Z}_2$ ensures that $C_p(K)$ is associative as well as abelian. To shorten the notation, we write C_p instead of $C_p(K)$ when K is clear from the context.

For each $0 \leq p \leq dim(K)$, we have a chain group C_p. These chain groups can be related by homomorphisms, which we discuss next.

Definition 2.1.18 (boundary map). *The* boundary map *$\partial_p : C_p \to C_{p-1}$ is a map from the group of p-chains to the group of $(p-1)$-chains. For a p-simplex $\sigma = (v_0, \ldots, v_p)$, the boundary is defined as the $(p-1)$-chain*

$$\partial\sigma = \sum_{i=0}^{p}(-1)^i(v_0, \ldots, \hat{v}_i, \ldots, v_p),$$

where $\hat{v}_i$ denotes the absence of the vertex v_i. The boundary of a p-chain $c = \sum a_i\sigma_i$ is defined as

$$\partial_p c = \partial_p \sum a_i\sigma_i = \sum a_i\partial_p\sigma_i,$$

which is a $(p-1)$-chain.

Since we use the field $\mathbb{Z}_2$, we can simplify the definition of the boundary of simplex σ to $\partial\sigma = \sum_{i=0}^{p}(v_0, \ldots, \hat{v}_i, \ldots, v_p)$. Informally, the boundary of a simplex is an oriented sum of its facets. In particular, the boundary of a vertex is 0. ∂_p commutes with addition, since $\partial_p(c_1 + c_2) = \partial_p c_1 + \partial_p c_2$. This means that ∂_p is a homomorphism from C_p to C_{p-1} and we call it a *boundary homomorphism.* Collecting the boundary homomorphisms for K, we get:

Definition 2.1.19 (chain complex). *The sequence*

$$\ldots \xrightarrow{\partial_{p+1}} C_p(K) \xrightarrow{\partial_p} C_{p-1}(K) \xrightarrow{\partial_{p-1}} C_{p-2}(K) \xrightarrow{\partial_{p-2}} \ldots$$

is called the chain complex *of K and is denoted by $C_*(K)$.*

Lemma 2.1.20. *For any $0 \leq p \leq dim(K)$ and any p-chain c, $\partial_{p-1} \circ \partial_p c = 0$.*

Proof. Let $c = \sum a_i\sigma_i$. We show that for any simplex σ in the combination, $\partial_{p-1} \circ \partial_p\sigma = 0$, which proves the claim.

$\partial_{p-1} \circ \partial_p\sigma$ consists of a sum of $(p-2)$-faces of σ. Each $(p-2)$-face of σ is incident to two $(p-1)$-faces of σ. $\partial_p(\sigma)$ consists of all $(p-1)$-faces of σ. As such, $\partial_{p-1} \circ \partial_p\sigma$ contains each $(p-2)$-face twice, which cancels out because of addition over $\mathbb{Z}_2$. The claim follows. ■

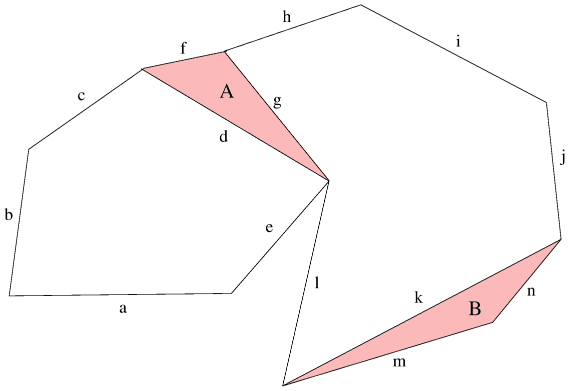


Figure 2.2: The figure contains a simplicial complex K which has two triangles, A and B in addition to the edges labeled from a to l. The following are the 2-chains of K: $0, A, B, A+B$. The 1-boundaries are the boundaries of 2-chains, so they are $0, d+f+g, k+m+n, \ldots$. The 1-chains $x := a+b+c+d+e$, $y := g+h+i+j+k+l$ and $z := g+h+i+j+n+m+l$ are some of the 1-cycles of K. We see that $y-z = n+k+m$ is a 1-boundary, so y and z are homologous cycles and it is also apparent from the figure that they are homotopic. On the other hand, $x-y$ is not a 1-boundary, so x and y are not homologous. $H_1(K)$ has two generators, because of the 1-cycles x and y.

Definition 2.1.21 (p-cycle). *An element c of C_p is called a* p-cycle*, if $\partial_p c = 0$.*

Let Z_p denote the collection of p-cycles of C_p. It is easy to see that Z_p is closed under addition and taking inverses, so Z_p is a sub-group of C_p. Moreover, by definition, Z_p is the kernel of the map ∂_p, $Z_p = \ker \partial_p$.

Definition 2.1.22 (p-boundary). *An element c of C_p is called a* p-boundary*, if there exists a $(p+1)$-chain τ of C_{p+1} such that $\partial_{p+1}\tau = c$.*

Let B_p denote the collection of p-boundaries of C_p. It is easy to see that B_p is a sub-group of C_p. Due to Lemma 2.1.20, it also follows that $B_p \subseteq Z_p$, so B_p is a sub-group of Z_p. Moreover, by definition B_p is the image of all $(p+1)$-chains of C_{p+1} under the map ∂_{p+1}, that is, $B_p = \text{im}\partial_{p+1}$.

Since B_p is a sub-group of the abelian group C_p, it is a normal group. Therefore, we can take the quotient of the two groups.

Definition 2.1.23 (homology group). *The* p-th Homology group *of K, $H_p(K)$ is defined as the quotient group $H_p(K) = Z_p(K)/B_p(K) = \ker \partial_p / \text{im}\partial_{p+1}$.*

Each element of H_p is obtained by taking any cycle $z \in Z_p$ and computing $z + B_p$, which is the coset of B_p in Z_p. If there is a cycle $z' = z + b$, $b \in B_p$, then by definition $z' + B_p = z + B_p$. We say that $z + B_p$ is a *homology class* and z, z' are *homologous* cycles of $z + B_p$, that is, they differ by a boundary.

Each element h of H_p represents a collections of homologous cycles, which informally means that all elements of h are homotopic. For an illustration, see Figure 2.2.

Definition 2.1.24 (Betti number). *The* p-th Betti number *of K is the rank of the group H_p,*

$$\beta_p = \text{rank}(H_p) = \text{rank}(Z_p) - \text{rank}(B_p).$$

Intuitively, β_p represents the number of p-dimensional holes in K for $p \geq 1$. β_0 represents the number of connected components of K.

2.2 Persistent homology

In the previous section we have seen the homology carried by a single simplicial complex. In this section we study maps between simplicial complexes and the effect on the homology groups.

2.2.1 Simplicial maps and induced homomorphisms

Definition 2.2.1 (vertex and simplicial maps). *Let K, L be simplicial complexes. A* vertex map *$\hat{f} : K^0 \to L^0$ is a map which assigns to each vertex $v \in K$, some vertex $\hat{f}(v) \in L$.*

If for each simplex $\sigma = (v_0, \ldots, v_k) \in K$, the set $\{\hat{f}(v_0), \ldots, \hat{f}(v_k)\}$ spans a simplex of L, then the linear extension of $\hat{f}$ to simplices of K, $f : K \to L$, is called the simplicial map *induced by $\hat{f}$. Formally, f is defined using barycentric co-ordinates as: if $x = \sum \lambda_i v_i$ is a point in σ, then $f(x) = \sum \lambda_i \hat{f}(v_i)$, $0 \leq \lambda_i \leq 1, \forall i$.*

Simplicial maps are an example of continuous maps between geometric simplicial complexes, but they are also applicable to abstract simplicial complexes, where the linear extension of the vertex map is skipped. An elementary observation is that the composition of any number of simplicial maps is a simplicial map. Simplicial maps do more than just relate simplicial complexes, which we explain next.

Definition 2.2.2 (chain map). *Let K, L be simplicial complexes. A* chain map *is a family of homomorphisms $f_\# : C_p(K) \to C_p(L)$ between the chain complexes of K and L, that satisfy $f_\# \circ \partial_{p,K} = \partial_{p,L} \circ f_\#$, for each p. Here, $\partial_{p,K}$ and $\partial_{p,L}$ are the p-th boundary maps of K and L, respectively.*

A simplicial map $f : K \to L$ gives rise to a chain map $f_\# : C_p(K) \to C_p(L)$ for each p. For each p-simplex $\sigma \in K$, we define $f_\#(\sigma) = f(\sigma)$ if $dim(f(\sigma)) = p$, and $f_\#(\sigma) = 0$ otherwise. For any chain $c = \sum a_i \sigma_i$, $f_\#(c)$ is then defined as $f_\#(c) = \sum a_i f_\#(\sigma_i)$. For $\sigma = (v_0, \ldots, v_p)$, we note that

$$\begin{aligned}
\partial_p f(\sigma) &= \partial_p[f(v_0), \ldots, f(v_p)] = \sum_{i=0}^{p} (-1)^i [f(v_0), \ldots, \hat{f(v_i)}, \ldots, f(v_p)] \\
&= \sum_{i=0}^{p} (-1)^i f(v_0, \ldots, \hat{v}_i, \ldots, v_p) = f\left(\sum_{i=0}^{p} (-1)^i (v_0, \ldots, \hat{v}_i, \ldots, v_p)\right) \\
&= f(\partial_p(\sigma)),
\end{aligned}$$

which implies that $\partial_p \circ f_\# = f_\# \circ \partial_p$. We say that $f_\#$ is the chain map induced by f.

We mention two important properties of chain maps:

- If $id : K \to K$ is the identity function on K, then the chain map induced by id, $id_\# : C_p(K) \to C_p(K)$ is identity on $C_p(K)$, for all p.
- If $f_1 : K_1 \to K_2, f_2 : K_2 \to K_3$ are simplicial maps for simplicial complexes K_1, K_2, K_3, then $(f_2 \circ f_1)_\# = (f_2)_\# \circ (f_1)_\#$.

The two properties exhibit the functorial nature of chain maps induced by simplicial maps: C_p sends simplices to chain complexes and # sends simplicial maps between two complexes to chain maps between the respective chain complexes. So $(C_p, \#)$ is a functor from the category of simplicial complexes and simplicial maps to the category of chain complexes and chain maps.

In this thesis, we mostly concern ourselves with chains maps induced by simplicial maps, but remark that chain maps are a more general concept, which should be clear from Definition 2.2.2.

Lemma 2.2.3. *The chain map* $f_{\#} : C_p(K) \to C_p(L)$ *induces homomorphisms of the form* $f_* : H_p(K) \to H_p(L)$ *between the respective homology groups, for each* p*. Since* $H_p(K)$ *is a vector space, we say that* f_* *is the* linear map *induced by* f*.*

Proof. For each cycle $c \in Z_p(K)$, it holds that $\partial_p f_{\#}(c) = f_{\#}(\partial_p c) = 0$, so $f_{\#}(c)$ is a cycle in $Z_p(L)$. This implies that $f_{\#}(Z_p(K)) \subseteq Z_p(L)$. For each boundary $b \in B_p(K)$, there exists a chain $e \in C_{p+1}(K)$ such that $\partial e = b$. Then, $f_{\#}(b) = f_{\#}(\partial e) = \partial f_{\#}(e)$ so that $f_{\#}(b)$ is a boundary in $B_p(L)$. This implies that $f_{\#}(B_p(K)) \subseteq B_p(L)$. Since $f_{\#}$ takes boundaries to boundaries and cycles to cycles, it induces a homomorphism $f_* : H_p(K) \to H_p(L)$ on the quotient groups. Also, f_* is induced by $f_{\#}$ which in turn is induced by f. Therefore, we say that f_* is induced by f. ■

Just like the extension of simplicial maps to chain maps, the extension of chain maps to linear maps is functorial in nature which means that it respects identity and composition. Taking into account both these functors, we can say that:

Definition 2.2.4 (homology functor). $(H, *)$ *is the* homology functor *from the category of simplicial complexes and simplicial maps to the category of abelian groups and homomorphisms.*

2.2.2 Towers and Persistence Modules

Definition 2.2.5 (inclusion). *A map* $inc : X \to Y$ *between topological spaces* X *and* Y *is called an* inclusion map*, if* $X \subseteq Y$ *and* $inc(x) = x, \forall x \in X$*. We denote this map as* $X \hookrightarrow Y$ *.*

Definition 2.2.6 (filtration). *Let* $I \subseteq \mathbb{R}$ *be a set of real values. A* filtration $(X)_{\alpha \in I}$ *is a collection of topological spaces* X_α *such that for each pair* $\{\alpha_1 \leq \alpha_2\} \in I$*, there is an inclusion* $X_{\alpha_1} \hookrightarrow X_{\alpha_2}$ *. The elements of the set* I *are called the* scales *of the filtration.*

We can obtain filtrations of simplicial complexes using concepts from Definition 2.1.14 and Definition 2.1.13.

- For any two scales $0 \leq \alpha \leq \beta$, the Rips complexes at those scales satisfy $\mathcal{R}_\alpha \subseteq \mathcal{R}_\beta$, so there is a natural inclusion from $\mathcal{R}_\alpha$ to $\mathcal{R}_\beta$. This gives a *Rips filtration* $(\mathcal{R}_\alpha)_{\alpha \geq 0}$ over the range of scales $[0, \infty)$.
- Similarly, the Čech complexes at those scales satisfy $\mathcal{C}_\alpha \subseteq \mathcal{C}_\beta$ which gives the *Čech filtration* $(\mathcal{C}_\alpha)_{\alpha \geq 0}$ over the range of scales $[0, \infty)$.

The above filtrations can also be defined for any subset of the range of scales $[0, \infty)$ with the same conditions. We next look at a more general collection of simplicial complexes:

Definition 2.2.7 (tower). *Let* $J \subset \mathbb{R}$ *be a discrete index set. A* (simplicial) tower *over* J *is a sequence of simplicial complexes* $(K_\alpha)_{\alpha \in J}$ *indexed over* J*, along with simplicial maps* $f_\alpha : K_\alpha \to K_{\alpha'}$ *for every pair of consecutive scales* $\alpha \leq \alpha' \in J$*.*

For any collection of consecutive scales $\{\alpha_1 \leq \ldots \leq \alpha_m\} \in J$, we can compose simplicial maps to get $f_{\alpha_1,\alpha_m} : K_{\alpha_1} \to K_{\alpha_m}$ by defining $f_{\alpha_1,\alpha_m} = f_{\alpha_{m-1}} \circ \ldots \circ f_{\alpha_1}$. This turns the tower into a category, where the objects are the simplicial complexes at different scales and the morphisms are simplicial maps between them. The simplest example of a tower is a filtration such as the Rips filtration restricted to a discrete number of scales, where the simplicial maps are simply inclusions.

Each tower can be written in the form

$$K_1 \xrightarrow{f_1} K_2 \xrightarrow{f_2} \ldots \xrightarrow{f_{m-1}} K_m$$

where each K_i differs from K_{i-1} in an elementary fashion:

- either K_i contains precisely one simplex more than K_{i-1}, in which case we call the simplicial map $f_{i-1} : K_{i-1} \to K_i$ an *elementary inclusion* and say that the simplex $\sigma = K_i \setminus K_{i-1}$ is *included* in the tower at scale i.
- or, there is exactly one pair of vertices $(u, v) \in K_{i-1}$ such that $f_{i-1}(u) = f_{i-1}(v)$. In this case, we say that f_{i-1} is an *elementary contraction* at scale i.

The size of a tower is the number of simplices which are included over all scales. For the special case of filtrations, the size of the tower is simply the largest complex in the sequence. However, this is not true for in general for simplicial towers, since simplices can collapse in the tower, and the size of the complex at a given scale may not take into account the collapsed simplices which were included at earlier scales in the tower.

Definition 2.2.8 (persistence module). *A* persistence module $\mathbb{V}$ *over an index set* $I \subseteq \mathbb{R}$ *is a collection of vector spaces* $(V_\alpha)_{\alpha\in I}$ *connected with homomorphisms of the form* $\phi^{\alpha,\beta} : V_\alpha \to V_\beta$ *for* $\alpha \leq \beta \in I$*, that satisfy:*

- $\phi^{\alpha,\alpha}$ *is the identity map on* V_α *for each* $\alpha \in I$*.*
- $\phi^{\alpha,\gamma} = \phi^{\beta,\gamma} \circ \phi^{\alpha,\beta}$ *for all* $\alpha \leq \beta \leq \gamma \in I$*.*

We can construct a persistence module from the concepts defined earlier. Let $(K_\alpha)_{\alpha\in I}$ be a tower connected with maps $\{f_\alpha\}_{\alpha\in I}$. As mentioned earlier, $(K_\alpha)_{\alpha\in I}$ along with the composition of simplicial maps $\{f_{\alpha,\beta}\}_{\alpha\leq\beta\in I}$ forms a category. Applying the homology functor from Definition 2.2.4 gives a collection of vector spaces $(H_p(K_\alpha))_{\alpha\in I}$ connected with linear maps $f_*^{\alpha,\beta} : H_p(K_\alpha) \to H_p(K_\beta)$ which satisfy the conditions for being a persistence module, for all p [BS14, Mun84]. Therefore, $(H_p(K_\alpha))_{\alpha\in I}$ is the p-th persistence module of $(K_\alpha)_{\alpha\in I}$. We denote by $H(K_\alpha)$ the direct sum of $H_p(K_\alpha)$ for all p, and in short say that $(H(K_\alpha))_{\alpha\in I}$ is the persistence module of the tower.

2.2.3 Persistence Barcodes

Next, we discuss a compact representation of persistence modules.

Definition 2.2.9 (interval module). *Given an index set* $J \subseteq \mathbb{R}$ *and a pair* $\{b < d\} \in J$*, an* interval module $\mathbb{I}_{b,d}$ *over a field* $\mathbb{F}$ *is a persistence module that consists of the following collection of vector spaces* V_i *and linear maps:*

$$V_i = \begin{cases} 0 & \text{for } i < b \subseteq J \\ \mathbb{F} & \text{for } i \in [b, d] \subseteq J \\ 0 & \text{for } i > d \subseteq J \end{cases} \qquad \text{and} \qquad \phi^i = \begin{cases} 0 & \text{for } i < b \subseteq J \\ id & \text{for } i \in [b, d-1] \subseteq J \\ 0 & \text{for } i \geq d \subseteq J \end{cases},$$

where ϕ^i is the linear map from V_i to the next space, and id is the identity function. Pictorially, this can be represented as:

$$\underbrace{0 \xrightarrow{0} \dots \xrightarrow{0} 0}_{i<b} \xrightarrow{0} \underbrace{\mathbb{F} \xrightarrow{id} \dots \xrightarrow{id} \mathbb{F}}_{i\in[b,d]\subseteq J} \xrightarrow{0} \underbrace{0 \xrightarrow{0} \dots \xrightarrow{0} 0}_{i>d}.$$

We consider interval modules over $\mathbb{Z}_2$. Let $\mathbb{V} := (V_\alpha)_{\alpha\in J}$ be a persistence module over an index set J. If each V_i is finite-dimensional, then $\mathbb{V}$ is isomorphic to a direct sum of interval modules [CdS10]:

$$\mathbb{V} \simeq \bigoplus \mathbb{I}_{b_i,d_i}.$$

Definition 2.2.10 (persistence barcode). *Let $\mathbb{V}$ be a persistence module which can be decomposed in the form $\mathbb{V} \simeq \mathbb{I}_{b_1,d_1} \bigoplus \dots \bigoplus \mathbb{I}_{b_m,d_m}$. The collection of intervals*

$$\{(b_1, d_1), \dots, (b_m, d_m)\}$$

is called the persistence barcode *of $\mathbb{V}$.*

It is evident that the algebraic structure of $\mathbb{V}$ is completely determined by its decomposition into interval modules. Therefore, the barcode completely characterizes $\mathbb{V}$ up to isomorphism.

We concern ourselves with persistence modules generated by towers of the form $(K_\alpha)_{\alpha\in J}$. Since $(H(K_\alpha))_{\alpha\in J}$ is a persistence module, it allows a decomposition into intervals. The barcode of such a module has a simple interpretation: each interval $[b, d]$ represents a homology class which existed in the sequence of groups $H(K_b), \dots, H(K_{d-1})$.

A more intuitive explanation is possible for the special case of filtrations. Let $(L_\alpha)_{\alpha\in J}$ be a filtration. Applying the homology functor, we get a persistence module $(H(L_\alpha))_{\alpha\in J}$ where the vector spaces are $H_p(L_\alpha)$ with linear maps $\phi^{\alpha,\beta} : H_p(L_\alpha) \to H_p(L_\beta)$ for all p and $\alpha \leq \beta \in J$. Let there be a homology class $\gamma \in H_p(L_\alpha)$ such that $\gamma \notin \mathrm{im}\phi^{\alpha-\varepsilon,\alpha}$ for any $\varepsilon > 0$. Then, we say that the class γ was *born* at scale α. Let β be the smallest scale such that $\phi^{\alpha,\beta}(\gamma) \in \mathrm{im}\phi^{\alpha-\varepsilon,\beta}$ for some $\varepsilon > 0$. We say that γ *dies* at scale β. In other words, the homology class γ *persists* for the range of scale $[\alpha, \beta]$. This range $[\alpha, \beta]$ corresponds to an interval in the barcode of $(H(L_\alpha))_{\alpha\in J}$.

Definition 2.2.11 (persistent homology groups and Betti numbers). *The* p-th persistent homology groups *of the module $(H(L_\alpha))_{\alpha\in J}$ are the images of the linear maps $H_p^{a,b} = \mathrm{im}\phi^{a,b}$. The* p-th persistent Betti numbers *are the ranks of the p-th homology groups, $\beta^{a,b} = \mathrm{rank}(H_p^{a,b})$.*

In other words, the persistent Betti numbers determine the barcode of the persistence module. A persistence module $\mathbb{V}$ with linear maps $\phi^{\cdot,\cdot}$ is said to be *tame*, if all the linear maps $\phi^{\alpha,\beta}$ have finite rank. In this thesis we only consider persistence modules which are tame. The condition for tameness allows for a different, but equivalent representation of the barcode [CdSGO16].

Definition 2.2.12 (persistence diagram). *Let $\mathbb{V}$ be a persistence module with barcode $\{(b_1, d_1), \dots, (b_m, d_m)\}$. The* persistence diagram (dgm) *of $\mathbb{V}$ is the multi-set of points*

$$\mathrm{dgm}(\mathbb{V}) = \{(b_i, d_i)\}_{1\leq i\leq m}$$

in the extended Euclidean plane $[-\infty, \infty]^2$. We say that the value $(d_i - b_i)$ is the persistence *of the interval (b_i, d_i).*

Since $d_i \geq b_i$ for each interval $[b_i, d_i]$ in the barcode, all the points in the persistence diagram lie on or above the diagonal $y = x$. See Figure 2.3 for an example.

2.2.4 Computing the barcode

We discuss an algorithm [EH10, ELZ02] to compute the persistence barcode from the persistence module induced by a filtration $\mathbb{K} := (K_\alpha)_{\alpha \geq 0}$.

Let $\{\sigma_1, \ldots, \sigma_m\}$ denote the simplices of $\mathbb{K}$, in the order of inclusion. Let M denote a $m \times m$ matrix whose rows and columns represent the simplices $\{\sigma_1, \ldots, \sigma_m\}$ in order and $M(i, j) = 1$ if and only if σ_i is a facet of σ_j. M is called the *boundary matrix* of $\mathbb{K}$.

From the definition, it is clear that M is an upper triangular matrix. For any $1 \leq j \leq m$, let $col(j)$ and $row(j)$ denote the j-th columns and rows of M, respectively. For any $col(j)$, let $anchor(j) = \{\max i \mid M(i, j) = 1\}$ denote the index of the lowest row which contains a 1 in $col(j)$. If $col(j)$ has only zero entries, then we let $anchor(j) = 0$. We call the matrix M as *reduced*, if for each $row(i)$ of M, there is a unique column $col(j)$ such that $anchor(j) = i$. In general, the boundary matrix M is not reduced. With a simple procedure, we transform M in to a reduced form.

Algorithm 1 Matrix Reduction

```
$R = M$
for $j = 1$ to $m$ do
  while there exists $k < j$ such that $anchor(k) = anchor(j)$ do
    Set $col(j) \leftarrow col(j) + col(k)$ (in $R$).
  end while
end for
```

Algorithm 1 transforms M in to R, a reduced matrix. The matrix R has the following properties:

- Each zero column corresponds to a *positive* simplex, whose addition creates a homology class. The number of zero columns corresponding to the p-simplices of $\mathbb{K}$ is the rank of Z_p.
- Each non-zero column corresponds to a *negative* simplex, whose addition destroys a homology class. The number of non-zero columns corresponding to the p-simplices of $\mathbb{K}$ is the rank of B_p.
- Consider a zero column $col(i)$ and the corresponding row, $row(i)$. There are two possibilities:
  - $row(i)$ contains an anchor, that is, $anchor(j) = i$ for some j. Then σ_j is said to be paired with σ_i. This means that σ_j destroys a class which was added by σ_i, and therefore $[\alpha_i, \alpha_j]$ is an interval in the barcode of $H_p(\mathbb{K})$, where $dim(\sigma_i) = p$.
  - $row(i)$ does not contain an anchor. Then, σ_i remains un-paired, so σ_i corresponds to a class which was born but not destroyed and gives an interval $[\alpha_i, \infty)$ in the barcode of $H_p(\mathbb{K})$, where $dim(\sigma_i) = p$.

While the reduced matrix R is not unique, the above properties hold for any matrix reduced from M. Therefore, collecting the column-anchor pairs as above gives the barcode of the filtration.

From the description of the algorithm, it is apparent that the runtime complexity is $O(m^3)$, accounting for the two loops and the addition of two columns. For practical data sets, however, it has been observed that the runtime is usually near-linear in m, because M is usually sparse.

Recent developments While we have briefly mentioned the simplest algorithm for computing persistence barcodes, there have been several advancements in this area. Algorithms to compute the barcode of filtrations have been improved to exploit the sparsity and the structure of the boundary matrix [BKR14, CK11]. Techniques have also been employed in a distributed setting [BKRW17]. Another way to get the barcode is to compute persistence *cohomology*, which is a dual to homology [BDM15, dSMVJ11a, dSMVJ11b]. A generalization of filtrations are *zig-zag* sequences, where inclusions can occur in both directions [CdS10] in the sequence of simplicial complexes, unlike filtrations where inclusions are monotonic. Algorithms have been developed to compute zig-zag persistence [CdSM09, MMv11]. Until now, we have only discussed about computing persistence for filtrations. However, our main concern are general simplicial towers. Fortunately, there have been recent developments in computing barcodes for general simplicial towers efficiently [DFW14, KS17].

2.3 Stability of persistence modules

In this section we discuss the notion of closeness of persistence modules. First, we look at a way to compare two barcodes.

2.3.1 Distance between barcodes

Let $\mathbb{V}, \mathbb{W}$ be two persistence modules. We discuss a way to compare the persistence diagrams $\mathrm{dgm}(\mathbb{V})$ and $\mathrm{dgm}(\mathbb{W})$ (Definition 2.2.12).

Definition 2.3.1 (bottleneck distance). *A bottleneck matching* $M \subseteq \mathrm{dgm}(\mathbb{V}) \times \mathrm{dgm}(\mathbb{W})$ *between points of* $\mathrm{dgm}(\mathbb{V})$ *and* $\mathrm{dgm}(\mathbb{W})$ *with cost* μ *is a matching which satisfies:*

- *each point of* $\mathrm{dgm}(\mathbb{V})$ *is matched to at most one point of* $\mathrm{dgm}(\mathbb{W})$ *and vice-versa, with* $\|v - w\|_\infty \leq \mu$ *for each* $(v, w) \in M$.
- *for each unmatched point of* $u \in \mathrm{dgm}(\mathbb{V}) \cup \mathrm{dgm}(\mathbb{W})$*, the* L_∞*-distance of* u *from the diagonal* $y = x$ *is at most* μ.

The bottleneck distance *between* $\mathrm{dgm}(\mathbb{V})$ *and* $\mathrm{dgm}(\mathbb{W})$ *is defined as*

$$\mathrm{d_B}(\mathrm{dgm}(\mathbb{V}), \mathrm{dgm}(\mathbb{W})) = \inf_{M \subseteq \mathrm{dgm}(\mathbb{V}) \times \mathrm{dgm}(\mathbb{W})} \mathit{cost}(M),$$

where $\mathit{cost}(M)$ *is the cost of the matching* M.

For simplicity of notation, we write $\mathrm{d_B}(\mathbb{V}, \mathbb{W})$ instead of $\mathrm{d_B}(\mathrm{dgm}(\mathbb{V}), \mathrm{dgm}(\mathbb{W}))$ when it is clear from context. Informally, features of low persistence are represented by points which are close to the diagonal and do not contribute to the bottleneck distance when comparing diagrams. The points with high persistence are matched to determine the bottleneck distance. See Figure 2.3 for an illustration.

Definition 2.3.2 (log-scale bottleneck distance). *Let* $\mathbb{V}, \mathbb{W}$ *be persistence modules and let* $\log \mathrm{dgm}(V)$ *denote the transformation of each point* $(x, y) \in \mathrm{dgm}(\mathbb{V})$ *to* $(\log x, \log y)$*. The multi-set of points* $\log \mathrm{dgm}(V)$ *is called the* log-scale persistence diagram *of* $\mathbb{V}$*. The* log-scale bottleneck distance *between the persistence diagrams of* $\mathbb{V}$ *and* $\mathbb{W}$ *is defined as* $\mathrm{d_B}(\log \mathrm{dgm}(V), \log \mathrm{dgm}(W))$.

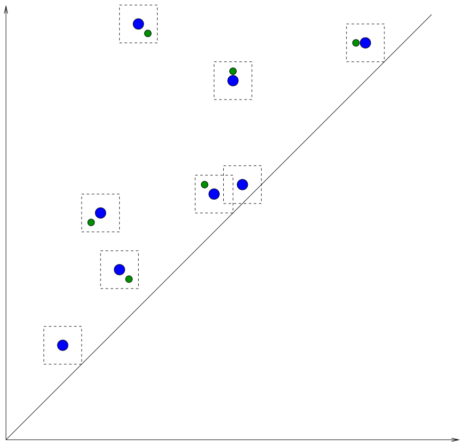


Figure 2.3: The collection of blue points (large dots) represents the persistence diagram of some persistence module. Similarly, the collection of green points (small dots) is the persistence diagram of some other persistence module. We have drawn each point with multiplicity one.
The figure also shows a matching between the two diagrams. When a green point and a blue point are in the same box, they are a matched pair in the bottleneck matching. Blue points in empty boxes are matched to the diagonal. The side-length of the box determines the cost of the matching.

2.3.2 Interleavings

Now we elaborate the conditions which can be used to compare persistence modules.

Definition 2.3.3 (weak interleaving [CCSG$^+$09]). *Let* $\mathbb{V}, \mathbb{W}$ *be persistence modules over an index set* $I \subseteq R$ *with linear maps* ϕ, ψ*, respectively.* $\mathbb{V}, \mathbb{W}$ *are said to be* (multiplicatively) weakly ε-interleaved *if there exist a pair of families of linear maps* $\gamma_\alpha : V_\alpha \to W_{\varepsilon\alpha}$ *and* $\delta_\alpha : W_\alpha \to V_{\varepsilon\alpha}$ *for all* $\alpha \in I$ *and* $\varepsilon > 0$*, such that the diagram*

$$\begin{tikzcd}
 & \dots \arrow[r] & V_{\alpha\varepsilon} \arrow[rr, "\phi"] \arrow[dr, "\gamma"] & & V_{\alpha\varepsilon^3} \arrow[r] & \dots \\
\dots \arrow[r] & W_\alpha \arrow[rr, "\psi"] \arrow[ur, "\delta"] & & W_{\alpha\varepsilon^2} \arrow[r] \arrow[ur, "\delta"] & \dots &
\end{tikzcd} \tag{2.1}$$

commutes for all $\alpha \in I$ *(the subscripts of the maps have been removed for readability).*

Theorem 2.3.4 (weak stability theorem [CCSG$^+$09]). *If two persistence modules* $\mathbb{V}, \mathbb{W}$ *are weakly* ϵ*-interleaved, then* $\mathrm{d_B}(\log \mathrm{dgm}(\mathbb{V}), \log \mathrm{dgm}(\mathbb{W})) \leq 3\varepsilon$*. We say that* $\mathbb{V}, \mathbb{W}$ *are* 3ε*-approximations of each other.*

Under more stringent conditions, the approximation factor can be improved:

Definition 2.3.5 (strong interleaving [CCSG$^+$09]). *Persistence modules* $\mathbb{V}, \mathbb{W}$ *indexed over* $\mathbb{R}$ *with respective linear maps* ϕ, ψ *are said to be* (multiplicatively) strongly ε-interleaved *if there exist a pair of families of linear maps* $\gamma_\alpha : V_\alpha \to W_{\varepsilon\alpha}$ *and* $\delta_\alpha : W_\alpha \to V_{\varepsilon\alpha}$

for $\varepsilon > 0$, such that the diagrams

$$\begin{tikzcd}
V_{\frac{\alpha}{\varepsilon}} \arrow[rrr, "\phi"] \arrow[dr, "\gamma"] & & & V_{\varepsilon\alpha'} \\
& W_{\alpha} \arrow[r, "\psi"] & W_{\alpha'} \arrow[ur, "\delta"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& V_{\varepsilon\alpha} \arrow[r, "\phi"] & V_{\varepsilon\alpha'} \\
W_{\alpha} \arrow[ur, "\delta"] \arrow[r, "\psi"] & W_{\alpha'} \arrow[ur, "\delta"] &
\end{tikzcd}$$

(2.2)

$$\begin{tikzcd}
& V_{\alpha} \arrow[r, "\phi"] & V_{\alpha'} \arrow[dr, "\gamma"] & \\
W_{\frac{\alpha}{\varepsilon}} \arrow[ur, "\delta"] \arrow[rrr, "\psi"] & & & W_{\varepsilon\alpha'}
\end{tikzcd}
\qquad
\begin{tikzcd}
V_{\alpha} \arrow[r, "\phi"] \arrow[dr, "\gamma"] & V_{\alpha'} \arrow[dr, "\gamma"] & \\
& W_{\varepsilon\alpha} \arrow[r, "\psi"] & W_{\varepsilon\alpha'}
\end{tikzcd}$$

commute for all $\alpha \leq \alpha' \in \mathbb{R}$.

Theorem 2.3.6 (strong stability theorem [CCSG$^+$09]). *If two persistence modules $\mathbb{V}, \mathbb{W}$ are strongly ϵ-interleaved, then $\mathrm{d_B}(\log \mathrm{dgm}(\mathbb{V}), \log \mathrm{dgm}(\mathbb{W})) \leq \varepsilon$. We say that $\mathbb{V}, \mathbb{W}$ are ε-approximations of each other.*

We remark that interleavings satisfy the triangle inequality. This result is folklore; see [BS14, Theorem 3.3] or [CdSGO16, Proposition 5.3] for a proof in a generalized context.

Lemma 2.3.7 (transitivity of interleavings). *Let $\mathbb{V}_1$, $\mathbb{V}_2$, and $\mathbb{V}_3$ be persistence modules. If $\mathbb{V}_1$ is a ε-approximation of $\mathbb{V}_2$ and $\mathbb{V}_2$ is a η-approximation of $\mathbb{V}_3$, then $\mathbb{V}_1$ is a $(\varepsilon\eta)$-approximation of $\mathbb{V}_3$.*

We end this part by discussing an important result which relates to the equivalence of persistence modules.

Theorem 2.3.8 (persistence equivalence theorem [CZ05, GOT17]). *Two persistence modules $\mathbb{V} = (V_\alpha)_{\alpha \in I}$ and $\mathbb{W} = (W_\alpha)_{\alpha \in I}$ on linear maps ϕ, ψ respectively are isomorphic, if there exists an isomorphism $f : V_\alpha \to W_\alpha$ for each $\alpha \in I$ such that the diagram*

$$\begin{tikzcd}
\ldots \arrow[r] & V_{\alpha} \arrow[r, "\phi"] \arrow[d, "f"] & V_{\beta} \arrow[r] \arrow[d, "f"] & \ldots \\
\ldots \arrow[r] & W_{\alpha} \arrow[r, "\psi"] & W_{\beta} \arrow[r] & \ldots
\end{tikzcd}$$

(2.3)

commutes for all $\alpha \leq \beta \in I$. Isomorphic persistence modules have identical persistence diagrams, that is, $\mathrm{dgm}(\mathbb{V}) = \mathrm{dgm}(\mathbb{W})$.

2.3.3 Scale balancing

Scaling a tower Let $\mathbb{A} := (A_\alpha)_{\alpha \in I}$ be a tower connected with simplicial maps ϕ over an index set I. We denote by $A'_\alpha := A_{\alpha\varepsilon}$ a complex at a scale factored by some $\epsilon > 0$ such that $\alpha, \alpha\epsilon \in I$ and there exist simplicial maps $\phi : A'_\alpha \to A'_\beta$ for all $\alpha, \beta \in I$. Then, we can build a tower $\mathbb{A}' := (A'_\alpha)_{\alpha \in I}$ using ϕ. Essentially, $\mathbb{A}'$ is a shifted version of $\mathbb{A}$ with respect to the scales involved, so the persistence barcodes of the two towers are related.

Let (x, y) be an interval in the barcode of the persistence module of $\mathbb{A}$. This means that there is a homology class which is born at scale x in the complex A_x and dies at scale y in the complex A_y. The complex A_x corresponds to $A'_{x/\varepsilon}$ and A_y corresponds to $A'_{y/\varepsilon}$. Therefore, the homology class corresponding to the interval $[x, y]$ in $\mathbb{A}$ appears in the interval $[x/\varepsilon, y/\varepsilon]$ in $\mathbb{A}'$. This leads to a injective map from the barcode of $\mathbb{A}$ to the barcode of $\mathbb{A}'$. The same argument also works in the opposite direction, from $\mathbb{A}'$ to $\mathbb{A}$. Therefore, we have a bijection between the two barcodes, which sends any interval $[b, d]$ of $\mathbb{A}$ to $[b/\varepsilon, d/\varepsilon]$ of $\mathbb{A}'$. The two barcodes are scaled versions of each other. Computing either barcode is equivalent, and in this thesis we treat the barcode of any tower and any shifted version as equivalent, without being explicit.

Scaling and interleaving Let $\mathbb{V} = (V_\alpha)_{\alpha\in I}$ and $\mathbb{W} = (W_\alpha)_{\alpha\in I}$ be two persistence modules over linear maps f_v, f_w, respectively. Let there be linear maps $\phi : V_{\alpha/\varepsilon_1} \to W_\alpha$ and $\psi : W_\alpha \to V_{\alpha\varepsilon_2}$ for $1 \le \varepsilon_1, \varepsilon_2$ such that all $\alpha, \alpha/\varepsilon_1, \alpha\varepsilon_2 \in I$. Suppose that the following diagram commutes, for all $\alpha \in I$.

$$\begin{tikzcd}
 & \cdots \arrow[r] & W_\alpha \arrow[rr, "f_w"] \arrow[dr, "\psi"] & & W_{\alpha\varepsilon_1\varepsilon_2} \arrow[r] & \cdots \\
\cdots \arrow[r] & V_{\alpha/\varepsilon_1} \arrow[ur, "\phi"] \arrow[rr, "f_v"] & & V_{\alpha\varepsilon_2} \arrow[ur, "\psi"] \arrow[r] & \cdots &
\end{tikzcd}$$
(2.4)

Let $\varepsilon := max(\varepsilon_1, \varepsilon_2)$. Then, by replacing $\varepsilon_1, \varepsilon_2$ by ε in Diagram (2.4), the diagram still commutes, so $\mathbb{V}$ is a 3ε-approximation of $\mathbb{W}$, by Theorem 2.3.4.

We improve the approximation factor by interpreting the scale a little differently. We define a new vector space $V'_{c\alpha} := V_\alpha$, where $c = \sqrt{\frac{\varepsilon_1}{\varepsilon_2}}$ and $c\alpha \in I$. This gives rise to a new persistence module, $\mathbb{V}' = (V_{c\alpha})_{\alpha\in I}$. The maps ϕ and ψ can then be interpreted as $\phi : V'_{\alpha/\sqrt{\varepsilon_1\varepsilon_2}} \to W_\alpha$, or $\phi : V'_\alpha \to W_{\alpha\sqrt{\varepsilon_1\varepsilon_2}}$ and $\psi : W_\alpha \to V'_{\alpha\sqrt{\varepsilon_1\varepsilon_2}}$. Then, Diagram (2.4) can be re-interpreted as

$$\begin{tikzcd}
 & \cdots \arrow[r] & W_{\alpha\sqrt{\varepsilon_1\varepsilon_2}} \arrow[rr, "f_w"] \arrow[dr, "\psi"] & & W_{\alpha(\varepsilon_1\varepsilon_2)^{3/2}} \arrow[r] & \cdots \\
\cdots \arrow[r] & V_{\alpha'} \arrow[ur, "\phi"] \arrow[rr, "f_v"] & & V'_{\alpha\varepsilon_1\varepsilon_2} \arrow[ur, "\psi"] \arrow[r] & \cdots &
\end{tikzcd}$$
(2.5)

which still commutes. Therefore, $\mathbb{V}'$ is a $3\sqrt{\varepsilon_1\varepsilon_2}$-approximation of $\mathbb{W}$, which is an improvement over $\mathbb{V}$, since $\sqrt{\varepsilon_1\varepsilon_2} \le max(\varepsilon_1, \varepsilon_2)$. As discussed earlier, $\mathbb{V}$ and $\mathbb{V}'$ have the same barcode up to scaling. Therefore, by re-interpreting the scales differently, we have improved the approximation factor.

This scaling trick also works when $\mathbb{V}$ and $\mathbb{W}$ are strongly interleaved. If we have the diagrams which commute (where we have skipped the maps for readability):

$$\begin{tikzcd}
W_\alpha \arrow[rrr] \arrow[dr] & & & W_{\alpha'\varepsilon_1\varepsilon_2} \\
 & V_{\alpha\varepsilon_2} \arrow[r] & V_{\alpha'\varepsilon_2} \arrow[ur] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & W_{\alpha\varepsilon_1} \arrow[r] & W_{\alpha'\varepsilon_1} \\
V_\alpha \arrow[ur] \arrow[r] & V_{\alpha'} \arrow[ur] &
\end{tikzcd}$$
(2.6)

$$\begin{tikzcd}
 & W_{\alpha\varepsilon_1} \arrow[r] & W_{\alpha'\varepsilon_1} \arrow[dr] & \\
V_\alpha \arrow[ur] \arrow[rrr] & & & V_{\alpha'\varepsilon_1\varepsilon_2}
\end{tikzcd}
\qquad
\begin{tikzcd}
W_\alpha \arrow[r] \arrow[dr] & W_{\alpha'} \arrow[dr] & \\
 & V_{\alpha\varepsilon_2} \arrow[r] & V_{\alpha'\varepsilon_2}
\end{tikzcd}$$

then $\mathbb{V}$ and $\mathbb{W}$ are $max(\varepsilon_1, \varepsilon_2)$-approximations of each other. By defining $\mathbb{V}'$ as before, the following diagrams

$$\begin{tikzcd}
W_\alpha \arrow[rrr] \arrow[dr] & & & W_{\alpha c^2} \\
& V'_{\alpha c} \arrow[r] & V'_{\alpha' c} \arrow[ur] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& W_{\alpha c} \arrow[r] & W_{\alpha' c} \\
V'_\alpha \arrow[ur] \arrow[r] & V'_{\alpha'} \arrow[ur] &
\end{tikzcd} \tag{2.7}$$

$$\begin{tikzcd}
& W_{\alpha c} \arrow[r] & W_{\alpha' c} \arrow[dr] & \\
V'_\alpha \arrow[ur] \arrow[rrr] & & & V'_{\alpha' c^2}
\end{tikzcd}
\qquad
\begin{tikzcd}
W_\alpha \arrow[r] \arrow[dr] & W_{\alpha'} \arrow[dr] & \\
& V'_{\alpha c} \arrow[r] & V'_{\alpha' c}
\end{tikzcd}$$

commute for $c = \sqrt{\varepsilon_1 \varepsilon_2}$, so we can improve a $\max(\varepsilon_1 \varepsilon_2)$-approximation to an $\sqrt{\varepsilon_1 \varepsilon_2}$-approximation.

We end the section by discussing an important relation between Čech and Rips filtrations.

Lemma 2.3.9. *The Čech persistence module $(H(\mathcal{C}_\alpha))_{\alpha \geq 0}$ and the Rips persistence module $(H(\mathcal{R}_\alpha))_{\alpha \geq 0}$ are $\sqrt[4]{2}$-approximations of each other.*

Proof. Recall from Lemma 2.1.15 that

$$\mathcal{C}_\alpha \subseteq \mathcal{R}_\alpha \subseteq \mathcal{C}_{\sqrt{2}\alpha}.$$

Using this result, it is straightforward to see that the following diagrams

$$\begin{tikzcd}
\mathcal{C}_\alpha \arrow[rrr, hook] \arrow[d, hook] & & & \mathcal{C}_{\varepsilon\alpha'} \\
\mathcal{R}_\alpha \arrow[rr, hook] & & \mathcal{R}_{\alpha'} \arrow[ur, hook] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathcal{C}_{\varepsilon\alpha} \arrow[r, hook] & \mathcal{C}_{\varepsilon\alpha'} \\
\mathcal{R}_\alpha \arrow[ur, hook] \arrow[r, hook] & \mathcal{R}_{\alpha'} \arrow[ur, hook] &
\end{tikzcd}$$

$$\begin{tikzcd}
& \mathcal{C}_\alpha \arrow[r, hook] & \mathcal{C}_{\alpha'} \arrow[d, hook] \\
\mathcal{R}_{\frac{\alpha}{\varepsilon}} \arrow[ur, hook] \arrow[rr, hook] & & \mathcal{R}_{\alpha'}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{C}_\alpha \arrow[r, hook] \arrow[d, hook] & \mathcal{C}_{\alpha'} \arrow[d, hook] \\
\mathcal{R}_\alpha \arrow[r, hook] & \mathcal{R}_{\alpha'}
\end{tikzcd}$$

commute for $\varepsilon = \sqrt{2}$ and all $\alpha' \geq \alpha \geq 0$. The claim follows from Theorem 2.3.6 and from applying the scale balancing technique for strongly interleaved modules on $\mathcal{C}$ and $\mathcal{R}$. ■

2.4 Additional concepts

We discuss two additional concepts, which is useful for our results.

2.4.1 Acyclic carriers

Definition 2.4.1 (acyclicity). *We call a simplicial complex K* acyclic*, if K is connected and all homology groups $H_p(K)$ are trivial.*

In particular, if K is contractible, then it is also acyclic.

Definition 2.4.2 (acyclic carrier). *For simplicial complexes* K *and* L*, an* acyclic carrier Φ *is a map that assigns to each simplex* σ *in* K*, a non-empty subcomplex* $\Phi(\sigma) \subseteq L$ *such that* $\Phi(\sigma)$ *is acyclic, and whenever* τ *is a face of* σ*, then* $\Phi(\tau) \subseteq \Phi(\sigma)$*.*

We say that a chain c in $\mathcal{C}_p(K)$ is *carried* by a subcomplex $K' \subseteq K$, if c takes value 0 except for p-simplices in K'. A chain map $\phi : \mathcal{C}_*(K) \to \mathcal{C}_*(L)$ is *carried by* Φ, if for each simplex $\sigma \in K$, $\phi(\sigma)$ is carried by $\Phi(\sigma)$. We make use of the *acyclic carrier theorem*:

Theorem 2.4.3 (acyclic carrier theorem [Mun84]). *Let* $\Phi : K \to L$ *be an acyclic carrier.*

- *There exists a chain map* $\phi : \mathcal{C}_*(K) \to \mathcal{C}_*(L)$ *such that* ϕ *is carried by* Φ*.*
- *If two chain maps* $\phi, \psi : \mathcal{C}_*(K) \to \mathcal{C}_*(L)$ *are both carried by* Φ*, then* $\phi \overset{h}{\simeq} \psi$ *and hence the induced linear maps* $\phi_* = \psi_*$*.*

Sometimes, we use an alternative form of the acyclic carrier theorem, which was stated in [Wal81], and guarantees that for an acyclic carrier $\Phi : K \to L$,

- There exists a continuous function $\psi : |K| \to |L|$ which is carried by Φ.
- If two continuous functions $f_1, f_2 : |K| \to |L|$ are both carried by Φ, then $f_1 \overset{h}{\simeq} f_2$, that is, they are homotopic and the induced linear maps are equal.

We mention a special case of acyclic carriers for simplicial maps:

Definition 2.4.4. *Let* K, L *be simplicial complexes and let* $f, g : K \to L$ *be simplicial maps.* f *and* g *are said to be* contiguous *maps, if for each simplex* $\sigma = (p_0, \ldots, p_k) \in K$*, the vertex set*

$$\{f(p_0), \ldots, f(p_k), g(p_0), \ldots, g(p_k)\}$$

forms a simplex in L*. In this case, the homomorphisms induced by* f *and* g *between* $H(K)$ *and* $H(L)$ *are equal [Mun84].*

2.4.2 Co-face distances

In the Čech filtration $(\mathcal{C}_\alpha)$ of a point set P, every simplex is associated with a *radius value*

$$\mathrm{rad}(\sigma) := \min\{\alpha \geq 0 \mid \sigma \in \mathcal{C}_\alpha\},$$

which is the radius of the minimal enclosing ball of its boundary vertices. If P is finite, the Čech filtration consists of a finite number of simplices, and we can define a *simplex-wise filtration*

$$\emptyset = \mathcal{C}^0 \subsetneq \mathcal{C}^2 \subsetneq \ldots \subsetneq \mathcal{C}^m,$$

where exactly one simplex is added from $\mathcal{C}^i$ to $\mathcal{C}^{i+1}$, and where τ is added before σ whenever $\mathrm{rad}(\tau) \leq \mathrm{rad}(\sigma)$. The filtration is not unique and ties can be broken arbitrarily.

In a simplex-wise filtration, passing from $\mathcal{C}^i$ to $\mathcal{C}^{i+1}$ means adding some k-simplex σ. The effect of this addition is that either a k-homology class comes into existence, or a $(k-1)$-homology class is destroyed. Depending on the case, we call σ *positive* or *negative*, respectively. In terms of the corresponding persistent barcode, there is exactly one interval *associated to* σ, either starting at i (if σ is positive) or ending at i (if σ is negative).

Definition 2.4.5 (co-face distance). *We define the* face (co-face) distance D_σ *(*D^*_σ*) of* σ *as the minimal distance between* σ *and its facets (co-facets),*

$$D_\sigma := \min_{\tau \text{ facet of } \sigma} (\operatorname{rad}(\sigma) - \operatorname{rad}(\tau)) \quad \textit{and} \quad D^*_\sigma := \min_{\tau \text{ co-facet of } \sigma} (\operatorname{rad}(\tau) - \operatorname{rad}(\sigma)).$$

Note that D_σ and D^*_σ can be zero. Nevertheless, we show that they constitute lower bounds for the persistence of the associated barcode intervals:

Lemma 2.4.6. *If* σ *is negative, the barcode interval associated to* σ *has persistence at least* D_σ*.*

Proof. σ kills a $(k-1)$-homology class by assumption, and this class is represented by the cycle $\partial\sigma$. However, this cycle came into existence when the last facet τ of σ was added. Therefore, the lifetime of the cycle destroyed by σ is at least $\operatorname{rad}(\sigma) - \operatorname{rad}(\tau)$. ■

Lemma 2.4.7. *If* σ *is positive, the homology class created by* σ *has persistence at least* D^*_σ*.*

Proof. σ creates a k-homology class; every representative cycle of this class is non-zero for σ. To turn such a cycle into a boundary, we have to add a $(k+1)$-simplex τ with σ in its boundary (otherwise, any $(k+1)$-chain formed will be zero for σ). Therefore, the cycle created at σ persists for at least $\operatorname{rad}(\tau) - \operatorname{rad}(\sigma)$. ■

Chapter 3

Geometric Concepts

In this chapter we discuss several geometric concepts that are crucial for our results. Some of the concepts are already well-known in literature, and we elaborate mostly on the parts relevant for our results. We direct the reader to the appropriate sources for more details. In addition, there are new results in most of the sections.

In Section 3.1 we explore the A^* lattice and its Voronoi polytope, which has many interesting combinatorial and geometric properties. The basic concepts can be found in textbooks [CSB87, Zie95]. Next we turn to the more familiar grid lattice in Section 3.2, where we discuss some properties of the cubical tessellation. In Section 3.3 we discuss a concept which captures the intrinsic dimension of any metric space [Ass83, Tal04]. Further, we discuss a data structure from [HPM06] for hierarchical clustering on metric spaces of low intrinsic dimension. We also revise the well-known concepts of pair decompositions, both in Euclidean space and general metric spaces [AHP12, CK95, PM13, Var98]. Section 3.4 presents a data structure [DIIM04, IM98] which can answer near-neighbor queries efficiently. We end the chapter by discussing techniques from [Bou85, JLS86, Mat90] to reduce the dimension of metric spaces in Section 3.5.

3.1 The A^* Lattice and the permutahedron

We begin the section by explaining a few basics of lattice geometry.

Definition 3.1.1 (lattice and basis). *A d-dimensional* lattice $L \subset \mathbb{R}^d$ *is the collection of all integer combinations of d independent vectors* $\{v_1, \ldots, v_d\}$ *in* $\mathbb{R}^d$*,*

$$L = \{x \in \mathbb{R}^d \mid x = \sum_{i=1}^{d} m_i v_i, \text{ where } (m_1, \ldots, m_d) \in \mathbb{Z}^d\}.$$

The collection of vectors $\{v_1, \ldots, v_d\}$ *is a* basis *of L.*

One of the simplest lattices is the integer lattice $\mathbb{Z}^d$, where the basis vectors are the standard basis of $\mathbb{R}^d$, $\{e_1, \ldots, e_d\}$. From the definition, we see that each lattice contains the origin $\mathbb{O}$.

Definition 3.1.2 (Voronoi cells of lattices). *For a lattice L and any point* $x \in L$*, the* Voronoi cell *of x in L, denoted by* $Vor_L(x) \subset \mathbb{R}^d$ *is the collection of points of* $\mathbb{R}^d$ *for which x is among the nearest lattice points,*

$$Vor_L(x) = \{y \in \mathbb{R}^d \mid \forall z \in L, \|y - x\| \leq \|y - z\|\}.$$

For simplicity, we call the Voronoi cell of the origin $\mathbb{O}$ *as the* Voronoi polytope *of L.*

The Voronoi polytope of a lattice is a convex polytope. The lattice is invariant under translation by any basis vector, so the Voronoi cell of each lattice point is simply a translation of the Voronoi polytope. For our elementary example of $\mathbb{Z}^d$, the Voronoi polytope is a d-cube of sidelength 1 centered at $\mathbb{O}$.

Definition 3.1.3 (neighbors and Voronoi vectors). *For a lattice L and a point $x \in L$, the* (lattice) neighbors *of x are the set of lattice points (including x) whose Voronoi cells intersect the Voronoi cell of x. We denote this set of points by*

$$\mathcal{NBR}(x) = \{y \in L \mid Vor_L(y) \cap Vor_L(x) \neq \varnothing\}.$$

The vectors $\{\vec{y} - \vec{x} \mid y \in \mathcal{NBR}(x), y \neq x\}$ are called the Voronoi vectors *of L.*

For the lattice $\mathbb{Z}^d$, it is easy to see that $\mathcal{NBR}(\mathbb{O})$ are the points $\{0, 1, -1\}^d$, and the Voronoi vectors are the corresponding vectors, excluding $(0, \ldots, 0)$. See Figure 3.1 for a different example.

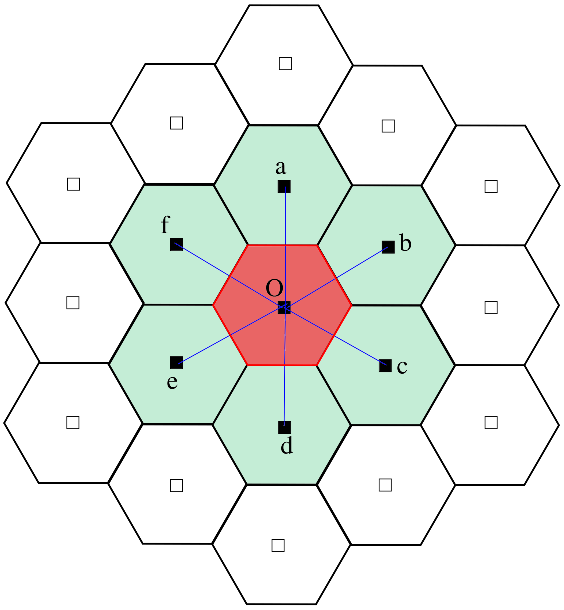


Figure 3.1: The figure shows a portion of the hexagonal lattice in the plane: the squares represent the lattice points. O represents the origin and the red (darkly shaded) hexagon is the Voronoi polytope of the lattice. The green (lightly shaded) hexagons intersect the Voronoi polytope. $\mathcal{NBR}(O)$ consists of the points $\{a, b, c, d, e, f, O\}$. The vectors $\{Oa, \ldots, Of\}$ are the Voronoi vectors.

Definition 3.1.4 (dual lattice). *Let L be a lattice. The* dual lattice *of L, denoted by L^* is the set of points*

$$L^* = \{y \in \mathbb{R}^d \mid \vec{y} \cdot \vec{x} \in \mathbb{Z}, \forall x \in L\}.$$

As an example, the integer lattice $\mathbb{Z}^d$ is self-dual. The lattices $2\mathbb{Z}^d$ and $\frac{1}{2}\mathbb{Z}^d$ are dual to each other. In three dimensions, the well-known lattices FCC (Face-centered cubic) and BCC (Body-centered cubic) are dual to each other[1].

Definition 3.1.5 (A_d lattice). *The d-dimensional A_d* lattice *consists of the set of points $(x_1, \ldots, x_{d+1}) \in \mathbb{Z}^{d+1}$ satisfying $\sum_{i=1}^{d+1} x_i = 0$. A basis for A_d consists of vectors of the form $(e_i, -1)$, $i = 1, \ldots, d$.*

---

[1] After suitable unitary transformations.

While the A_d lattice is defined in $\mathbb{R}^{d+1}$, all points lie on the hyperplane HP defined by $\sum_{i=1}^{d+1} y_i = 0$. After a suitable change of basis, we can express A_d by d vectors in $\mathbb{R}^d$, so it is indeed a d-dimensional lattice. In low dimensions, A_2 is more commonly known as the hexagonal lattice, and A_3 is a translated and rotated version of the FCC lattice that realizes the best sphere packing configuration in $\mathbb{R}^3$ [Hal05].

Definition 3.1.6 (A_d^* lattice). *The d-dimensional A_d^** lattice *is the dual lattice to A_d. The standard basis for A_d^* consists of the vectors*

$$\frac{1}{(d+1)}(\underbrace{t,\ldots,t}_{d+1-t}, \underbrace{t-(d+1),\ldots,t-(d+1)}_{t}), t = 1,\ldots,d.$$

From the definition, it follows that $A_d \subseteq A_d^*$, and that A_d^* lies in HP. A_2^* is the hexagonal lattice, same as A_2. A_3^* is a translated and rotated version of the BCC lattice that realizes the thinnest sphere covering configuration among lattices in $\mathbb{R}^3$ [Bam54]. We are mostly interested in the Voronoi polytope of A_d^*.

Definition 3.1.7 (permutahedron). *The d-dimensional* permutahedron*, denoted by Π_d, is the Voronoi polytope of the A_d^* lattice. Π_d has $(d+1)!$ vertices obtained by taking all permutations of the coordinates* [2] *of*

$$\frac{1}{2(d+1)}(d, d-2, d-4, \ldots, -d+2, -d).$$

See Figure 3.2 for an three-dimensional example. For brevity, we write $A^* := A_d^*$ and $\Pi := \Pi_d$ when d is clear from the context. Π is a d-dimensional convex polytope and lies in the hyperplane HP. The A^* lattice is sometimes also called the *permutahedral lattice*.

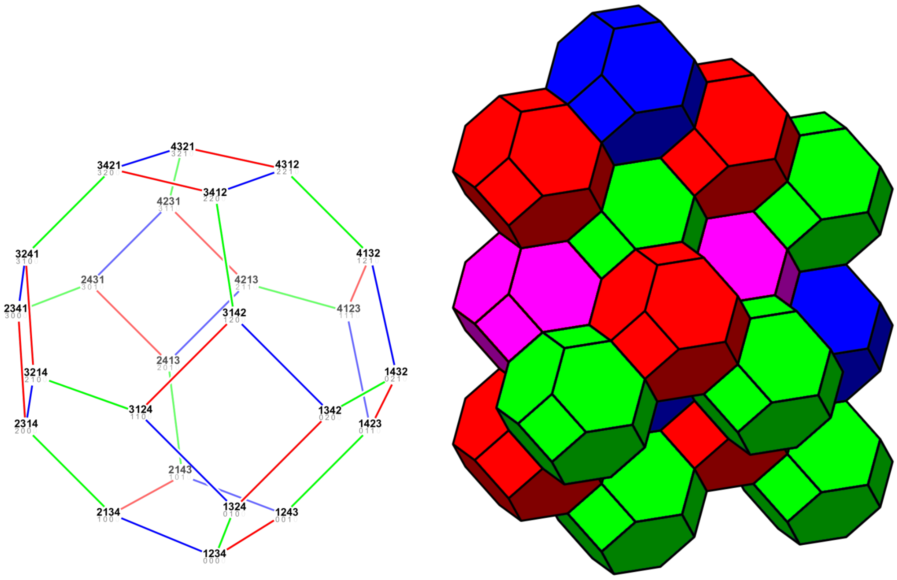


Figure 3.2: The left figure shows the 1-skeleton of a three dimensional permutahedron, which is a truncated octahedron. The right figure shows a tessellation of $\mathbb{R}^3$ using permutahedra. Both figures are taken from Wikipedia [Per] (authors: Tilman Piesk, Tom ruen, licensed under CC BY 3.0).

---

[2] Often a scaled, translated and rotated version is considered, in which the vertices of the permutahedron are all permutations of the point $(1,\ldots,d+1)$.

3.1.1 Combinatorial properties

The k-faces of Π are in correspondence to ordered partitions of the set of coordinate indices $[d+1] := \{1, \dots, d+1\}$ into $(d+1-k)$ non-empty subsets $S_1, \dots, S_{d+1-k}$. That means, each vertex v of any k-face is such that all coordinates of v at the indices of S_i are smaller than all coordinates of v at the indices of S_j for all $1 \le i < j \le d+1-k$. For example, with $d = 3$, the partition $(\{1,3\},\{2,4\})$ is the 2-face spanned by all points for which the two smallest coordinates appear at the first and the third position. This is an example of a facet of Π, for which we need to partition the indices in exactly 2 subsets. Equivalently, the facets of Π are in one-to-one correspondence to non-empty proper subsets of $[d+1]$, so Π has $2^{d+1} - 2$ facets.

The vertices of Π are the $(d+1)$-fold ordered partitions of $[d+1]$, which correspond to permutations of $[d+1]$, reinstating the fact that Π has $(d+1)!$ vertices. Moreover, any two faces σ, τ of Π with $\dim \sigma < \dim \tau$ are incident if the partition of σ is a refinement of the partition of τ. Continuing our example from before, the four 1-faces bounding the 2-face $(\{1,3\},\{2,4\})$ are $(\{1\},\{3\},\{2,4\})$,$(\{3\},\{1\},\{2,4\})$, $(\{1,3\},\{2\},\{4\})$, and $(\{1,3\},\{4\},\{2\})$. Vice-versa, we obtain co-faces of a face by combining consecutive partitions into one larger partition. For instance, the two co-facets of $(\{1,3\},\{4\},\{2\})$ are $(\{1,3\},\{2,4\})$ and $(\{1,3,4\},\{2\})$.

Lemma 3.1.8. *Let σ and τ be two facets of Π, defined using the two ordered partitions $(S_\sigma, [d+1] \setminus S_\sigma)$ and $(S_\tau, [d+1] \setminus S_\tau)$, respectively. Then σ and τ are adjacent in Π iff either $S_\sigma \subseteq S_\tau$ or $S_\tau \subseteq S_\sigma$.*

Proof. Two facets are adjacent if they share a common face. By the properties of the permutahedron, this means that the two facets are adjacent if and only if their partitions permit a common refinement, which is only possible if one set of defining indices is contained in the other. ■

We have already established that Π has "few" $(2^{d+1} - 2 = 2^{O(d)})$ $(d-1)$-faces and "many" $((d+1)! = 2^{O(d \log d)})$ 0-faces. We give an interpolating bound for all faces of intermediate dimensions.

Lemma 3.1.9. *The number of $(d-k)$-faces of Π is upper bounded by $2^{3(d+1)\log_2(k+1)}$.*

Proof. By our characterization of faces of Π, it suffices to count the number of ordered partitions of $[d+1]$ into $(k+1)$ subsets. That number equals $(k+1)!$ times the number of unordered partitions. The number of unordered partitions, in turn, is known as *Stirling number of the second kind* [RD69] and is bounded by $\frac{1}{2}\binom{d+1}{k+1}(k+1)^{d-k}$. To get an upper bound for the number of $(d-k)$-faces, we multiply the Stirling number with $(k+1)!$ and get

$$\begin{aligned}
&\frac{1}{2}\binom{d+1}{k+1}(k+1)^{d-k}(k+1)! \le (d+1)^{k+1}(k+1)^{d-k}(k+1)! \\
&\le (d+1)^{k+1}(k+1)^{d-k}(k+1)^{k+1} \le (d+1)^{k+1}(k+1)^{d+1} \\
&\le (k+1)^{3(d+1)} = 2^{3(d+1)\log_2(k+1)},
\end{aligned}$$

where we have used the fact that $(d+1)^{k+1} \le (k+1)^{2(d+1)}$ for $k \le d$. ■

3.1.2 Geometric properties

All vertices of Π are equidistant from $\mathbb{O}$, and a simple calculation shows that this distance is $\sqrt{\frac{d(d+2)}{12(d+1)}}$. Using the triangle inequality, we obtain:

Lemma 3.1.10. *The diameter of* Π *is at most* $\sqrt{d}$.

Definition 3.1.11 (tessellation and triangulation). *The union of permutahedra centered at all lattice points of* A^* *is the* Voronoi tessellation *of* A^*. *The nerve of this tessellation is the* Delaunay triangulation $\mathcal{D}$ *of* A^*.

Definition 3.1.12 (remainder point). *We call a point* $v \in \mathbb{R}^{d+1}$ *as a* remainder-k *point, if it is of the form*

$$v = \frac{1}{(d+1)}(a_0, \ldots, a_d)$$

such that $a_i \equiv k \bmod (d+1), \forall i$ *for some* $k \in \mathbb{Z}$.

An important property of A^* is that $\mathcal{D}$ is non-degenerate, unlike the Delaunay triangulation of the integer lattice.

Lemma 3.1.13. *Each vertex of* Π *has precisely* $(d+1)$ *permutahedral cells incident to it. In other words, the points of the* A_d^* *lattice are in general position. As a consequence, we can identify Delaunay simplices incident to* $\mathbb{O}$ *with faces of* Π.

Proof. The proof idea is to look at any vertex of Π and argue that it has precisely $(d+1)$ equidistant lattice points. See [BA09, Theorem 2.5] for a concise argument. Here, we rephrase the proof idea of [BA09, Theorem 2.5] in slightly simplified terms.

We show that all Delaunay cells of the A_d^* lattice are d-simplices, which proves our claim. Let $\vec{v}$ be a vertex of Π. Without loss of generality, we can assume that $\vec{v} = \frac{1}{2(d+1)}(d, d-2, \ldots, -d)$. The A_d^* lattice points closest to $\vec{v}$ define the Delaunay cell corresponding to $\vec{v}$.

Recall that the basis vectors of A_d^* are of the form

$$g_t = \frac{1}{(d+1)}(\underbrace{t, \ldots, t}_{d+1-t}, \underbrace{t-(d+1), \ldots, t-(d+1)}_{t})$$

for $1 \leq t \leq d$ [CSB87]. From Definition 3.1.12, we see that g_t is a *remainder-t* point. Since any lattice point $y \in A_d^*$ can be written as a integral combination of the vectors g_ts, y turns out to be a remainder-k point for some $k \in \mathbb{Z}$. So we can write $\vec{y} = \frac{1}{d+1}(\vec{m}(d+1) + k\vec{1})$, where $\vec{m} \in \mathbb{Z}^{d+1}$. y lies in HP, which means that $\vec{y} \cdot \vec{1} = 0$, and that translates to $\vec{m} \cdot \vec{1} = -k$.

We fix k and find the remainder-k point y, which is closest to v. To do so, we minimize the distance between v and y by choosing a suitable value for $\vec{m}$. In other words, we wish to find $\text{argmin}_{\vec{m}}||\vec{y}-\vec{v}||^2$. We see that

$$\begin{aligned}
\text{argmin}_{\vec{m}}||\vec{y}-\vec{v}||^2 &= \text{argmin}_{\vec{m}}\sum(m_i+\frac{k}{d+1}-v_i)^2 \\
&= \text{argmin}_{\vec{m}}\sum(m_i-v_i)^2+2(m_i-v_i)\frac{k}{d+1} \\
&= \text{argmin}_{\vec{m}}\sum(m_i-v_i)^2+\frac{2k}{d+1}\sum m_i \\
&= \text{argmin}_{\vec{m}}\sum(m_i-v_i)^2+\frac{2k}{d+1}\cdot(-k) \\
&= \text{argmin}_{\vec{m}}\sum(m_i-v_i)^2 \\
&= \text{argmin}_{\vec{m}}||\vec{m}-\vec{v}||^2 \\
&= \text{argmin}_{\vec{m}}||\vec{m}-\frac{1}{2(d+1)}(d,\ldots,-d)||^2
\end{aligned}$$

Using $\vec{m}\cdot\vec{1}=-k$ and an elementary calculation, we see that $||\vec{y}-\vec{v}||^2$ is minimized for

$$\vec{m}=(\underbrace{0,\ldots,0}_{d+1-k},\underbrace{-1,\ldots,-1}_{k}),$$

given any fixed k. This means that there is a unique remainder-k nearest lattice point to $\vec{v}$, for $k\in\{0,\ldots,d\}$. Moreover, the corresponding lattice points $\vec{y}$ are Delaunay neighbors of the origin, and are equidistant from $\vec{v}$. The Delaunay cell corresponding to $\vec{v}$ contains precisely $(d+1)$ points, one for each value of k, which proves the claim for the vertex $\vec{v}$.

Recall that any other vertex $\vec{u}$ of Π can be written as some permutation π of $\vec{v}$, that is, $\vec{u}=\pi(\vec{v})$. Following the above derivation, the nearest lattice points for $\vec{u}$ can be found by simply applying the permutation π on the nearest lattice points for $\vec{v}$. As a result, the vertex $\vec{u}$ also has $(d+1)$ nearest lattice points, and the corresponding d-simplices are congruent for all $\vec{u}$. This proves the claim. ■

Proposition 3.1.14. *The $(k-1)$-simplices in $\mathcal{D}$ that are incident to $\mathbb{O}$ are in one-to-one correspondence to the $(d-k+1)$-faces of Π and, hence, in one-to-one correspondence to the ordered k-partitions of $[d+1]$.*

This means that if $\sigma,\tau\in\mathcal{D}$ are simplices such that σ is a face of τ, then the partition corresponding to τ is a refinement of the partition corresponding to σ.

Let $V:=\mathcal{NBR}(\mathbb{O})$ denote the set of lattice points that share a Delaunay edge with the origin. The following statement shows that the point set V is in convex position, and the convex hull encloses Π with some "safety margin". The proof is a mere calculation, deriving an explicit equation for each hyperplane supporting the convex hull and applying it to all vertices of V and of Π.

Lemma 3.1.15. *Let σ be any d-simplex incident to $\mathbb{O}$. The facet $\tau\subset\sigma$ which is opposite to $\mathbb{O}$ lies on a hyperplane that is at least a distance $\frac{1}{\sqrt{2(d+1)}}$ to Π and all points of V are either on the hyperplane or on the same side as $\mathbb{O}$.*

Proof. Consider the d-simplex σ incident to $\mathbb{O}$ that is dual to the vertex of Π with coordinates

$$v=\frac{1}{d+1}\Big(d/2,d/2-1,\ldots,d/2-(d-1),d/2-d\Big).$$

The $(d-1)$-facet τ of σ opposite to $\mathbb{O}$ is spanned by lattice points of the form

$$\ell_k = \frac{1}{(d+1)}(\underbrace{k,\ldots,k}_{d+1-k},\underbrace{k-(d+1),\ldots,k-(d+1)}_{k}), 1 \leq k \leq d,$$

(see the proof of Lemma 3.1.13 above). All points in V can be obtained by permuting the coordinates of ℓ_k.

We can verify at once that all points of τ lie on the hyperplane $-x_1 + x_{d+1} + 1 = 0$, so this plane supports τ. The origin lies on the positive side of the plane. All points in V either lie on the plane or are on the positive side as well. For the vertices of Π, observe that the value $x_1 - x_{d+1}$ is minimized for the point v above, for which $x_1 - x_{d+1} + 1 = 1/(d+1)$ is obtained. It follows that v as well as any vertex of V is at least in distance $\frac{1}{\sqrt{2}(d+1)}$ from H (the $\sqrt{2}$ comes from the length of the normal vector). This proves the claim for the simplex dual to v.

Any other choice of σ is dual to a permuted version of v. Let π denote the permutation on v that yields the dual vertex. The vertices of τ are obtained by applying the same permutation on the points ℓ_k from above. Consequently, the plane equation changes to $-x_{\pi(1)} + x_{\pi(d+1)} + 1 = 0$. The same reasoning as above applies, proving the statement in general. ■

Lemma 3.1.16. *If two lattice points are not adjacent in $\mathcal{D}$, then the corresponding Voronoi polytopes have a distance of at least $\frac{\sqrt{2}}{d+1}$.*

Proof. Lemma 3.1.15 shows that Π is contained in a convex polytope C, which is the convex hull of V. Also, the distance of Π to the boundary of C is at least $\frac{1}{\sqrt{2}(d+1)}$. Let o' be a lattice point not in $\mathcal{NBR}(\mathbb{O})$. Let Π' be the Voronoi cell of o', and C' be convex hull of $\mathcal{NBR}(o')$. C' is interior-disjoint from C. To see that, note that the simplices in $\mathcal{D}$ incident to the origin triangulate the interior of C, and likewise for o'. Any interior intersection would be covered by a simplex incident to $\mathbb{O}$ and one incident to o', and since they are not connected, the simplices are distinct, contradicting the fact that $\mathcal{D}$ is a triangulation. Having established that C and C' are interior-disjoint, the distance between Π and Π' is at least $\frac{2}{\sqrt{2}(d+1)}$, as required. ■

Recall the definition of a flag complex as the maximal simplicial complex that one can form from a given graph (Definition 2.1.10). We next show that $\mathcal{D}$ is of this form. Our proof exploits certain properties of the A^* lattice, but we could not exclude the possibility that the Delaunay triangulation of any lattice is a flag complex and leave it as an open question. For the proof, we need the following fact, which is straightforward to prove:

Lemma 3.1.17. *For any natural number $N \geq 2$, consider any two vectors in $\mathbb{R}^N$, $U = (u_1, \ldots, u_N)$ and $W = (w_1, \ldots, w_N)$ with $u_1 \leq \ldots \leq u_N$ and $w_1 \leq \ldots \leq w_N$. Let π be a permutation over $[N]$, and let $\pi(W)$ be the vector with the corresponding permuted coordinates of W. Then, $\max_\pi\{U \cdot \pi(W)\} = U \cdot W$.*

An implication of the following result is that $\mathcal{D}$ is a flag complex. We use this to prove the main result.

Lemma 3.1.18. *Consider any two facets f_1 and f_2 of Π that are disjoint, that is, they do not share a vertex. In the tessellation, there are permutahedra $\Pi_1 \neq \Pi$ attached to f_1 and $\Pi_2 \neq \Pi$ attached to f_2, respectively. Then, Π_1 and Π_2 are disjoint.*

Proof. We prove the claim by explicitly constructing a hyperplane which strictly separates Π_1 and Π_2.

Let $(S_1, [d+1] \setminus S_1)$, $(S_2, [d+1] \setminus S_2)$ be the partitions defining facets f_1 and f_2, respectively. Since f_1 and f_2 are disjoint, we have that $S_1 \not\subset S_2$ and $S_2 \not\subset S_1$ by Lemma 3.1.8. Let us define the sets $T_1 = S_1 \setminus S_2$, $T_2 = S_2 \setminus S_1$, $T_3 = S_1 \cap S_2$ and $T_4 = [d+1] \setminus S_1 \cup S_2$. Also, let $|T_1| = a$, $|T_2| = b$ and $|T_3| = c$ with $a, b, c \geq 1$. Then, $|T_4| = d + 1 - (a + b + c)$, let $|S_1| = k := a + c$ and $|S_2| = p := b + c$.

Let ℓ_1, ℓ_2 denote the lattice points at the centers of the permutahedra Π_1, Π_2 that are attached to Π on the faces f_1 and f_2, respectively. We can derive the coordinates of ℓ_1 and ℓ_2 easily: an elementary calculation shows that barycenter of the face f_1 has coordinates $\frac{k-(d+1)}{2(d+1)} = \frac{k}{2(d+1)} - \frac{1}{2}$ at indices in S_1 and $\frac{k}{2(d+1)}$ at the rest of the positions. Similarly, the barycenter of f_2 has coordinates $\frac{p-(d+1)}{2(d+1)} = \frac{p}{2(d+1)} - \frac{1}{2}$ at indices in S_2 and $\frac{p}{2(d+1)}$ otherwise. Since Π is centered at the origin, the coordinates of ℓ_1 and ℓ_2 are obtained by multiplying the coordinates of the barycenters with 2. See Table 3.1 for details.

Let B denote the bisector hyperplane between ℓ_1 and ℓ_2. We show that B is a separating hyperplane for Π_1 and Π_2 with no point of either on the hyperplane, which proves the claim. The vector $n = (n_1, \ldots, n_{d+1}) := \ell_2 - \ell_1$ is a normal vector to B. Then, we define B by $n \cdot (x - m) = 0$ with $m = (\ell_1 + \ell_2)/2$ being the midpoint of ℓ_1 and ℓ_2. See Table 3.1 for a description of n and m.

indices	ℓ_2	ℓ_1	$n = \ell_2 - \ell_1$	$m = (\ell_2 + \ell_1)/2$	count
T_1	$\frac{p}{d+1}$	$\frac{k}{d+1} - 1$	$\frac{(p-k)}{d+1} + 1 = \alpha + 1$	$\frac{(p+k)}{2(d+1)} - \frac{1}{2} = \beta - 1/2$	a
T_2	$\frac{p}{d+1} - 1$	$\frac{k}{d+1}$	$\frac{(p-k)}{d+1} - 1 = \alpha - 1$	$\frac{(p+k)}{2(d+1)} - \frac{1}{2} = \beta - 1/2$	b
T_3	$\frac{p}{d+1} - 1$	$\frac{k}{d+1} - 1$	$\frac{p-k}{d+1} = \alpha$	$\frac{(p+k)}{2(d+1)} - 1 = \beta - 1$	c
T_4	$\frac{p}{d+1}$	$\frac{k}{d+1}$	$\frac{p-k}{d+1} = \alpha$	$\frac{p+k}{2(d+1)} = \beta$	e

Table 3.1: ℓ_1, ℓ_2, n, m. Here, $e = d + 1 - (a + b + c)$.

Since permutahedra tile space by translation, the vertices of Π_1 are of the form $x_1 = \ell_1 + \pi$ where π is any permutation of $y = \frac{1}{d+1}(\frac{d}{2}, \frac{d}{2} - 1, \ldots, \frac{-d}{2})$. Writing $B(x_1) := n \cdot (x_1 - m)$ for the function whose sign determines the halfspace of x_1 with respect to B, we can write $B(x_1) = B(\ell_1 + \pi) = n \cdot (\ell_1 + \pi - m) = n \cdot \ell_1 - n \cdot m + n \cdot \pi$. Similarly, for any vertex $x_2 = \ell_2 + \pi$ of Π_2, $B(x_2) = n \cdot \ell_2 - n \cdot m + n \cdot \pi$. We show that $B(x_1) < 0$ and $B(x_2) > 0$ for all permutations π. First, we calculate $n \cdot \ell_1$, $n \cdot \ell_2$ and $n \cdot m$ using Table 3.1:

$$\begin{aligned} n \cdot \ell_1 = &(\alpha + 1)(\frac{k}{d+1} - 1)a + (\alpha - 1)\frac{k}{d+1}b \\ &+ \alpha(\frac{k}{d+1} - 1)c + \alpha(\frac{k}{d+1})\{d + 1 - (a + b + c)\} \end{aligned}$$

Upon simplification, this reduces to $n \cdot \ell_1 = -a + \frac{k}{d+1}(a - b)$. Similarly, we calculate that $n \cdot \ell_2 = b + \frac{p}{d+1}(a - b)$. Next,

$$n \cdot m = (\alpha + 1)(\beta - 1/2)a + (\alpha - 1)(\beta - 1/2)b + \alpha(\beta - 1)c + \alpha\beta[d + 1 - (a + b + c)]$$

This simplifies to $n \cdot m = -(a - b)\frac{(d+1)-(p+k)}{2(d+1)}$. Subtracting, we get

$$n \cdot \ell_1 - n \cdot m = -a + \frac{k}{d+1}(a - b) + (a - b)\frac{(d+1) - (p+k)}{2(d+1)}$$

which reduces to $n \cdot \ell_1 - n \cdot m = -\frac{a+b}{2} + \frac{(b-a)^2}{2(d+1)}$.

Since $\ell_1 - m = -(\ell_2 - m)$, hence $n \cdot (\ell_2 - m) = -n \cdot (\ell_1 - m)$. Also,

$$\begin{aligned} n \cdot \ell_1 - n \cdot m &= -\frac{a+b}{2} + \frac{(b-a)^2}{2(d+1)} < -\frac{a+b}{2} + \frac{(b+a)^2}{2(d+1)} \\ &< -\frac{a+b}{2}(1 - \frac{a+b}{d+1}) < 0. \end{aligned}$$

Hence, $n \cdot \ell_1 - n \cdot m$ is negative and $n \cdot \ell_2 - n \cdot m$ is positive. Substituting these values in $B(x_1)$ and $B(x_2)$, we get

$$B(x_1) = -\frac{a+b}{2} + \frac{(b-a)^2}{2(d+1)} + n \cdot \pi, B(x_2) = \frac{a+b}{2} - \frac{(b-a)^2}{2(d+1)} + n \cdot \pi$$

We now calculate the maximum absolute value of $n \cdot \pi$ and use it to show that $B(x_1)$ is always negative and $B(x_2)$ is always positive.

The dot product $n \cdot \pi$ is obtained by first multiplying each component y_i of the vector $y = \frac{1}{d+1}(\frac{d}{2}, \frac{d}{2} - 1, \ldots, \frac{-d}{2})$ with a component of n, which has one of 3 values: $\alpha + 1$ for indices in T_1, α for $T_3 \cup T_4$, $\alpha - 1$ for T_2 (refer Table 3.1); the intermediate products are then added up. The permutation of y maximizing $n \cdot \pi$ is found using Lemma 3.1.17.

Let us denote the sum of the q smallest components of y by N_q and the sum of the q largest components of y by M_q. It is easy to verify that $M_q + N_q = 0$, $N_q = N_{d+1-q}$ and $M_q = M_{d+1-q}$. Then,

$$\begin{aligned} max(|n \cdot \pi|) &= (\alpha+1)M_a + (\alpha-1)N_b + \alpha(N_{d+1-a} - N_b) \\ &= \alpha(M_a + N_{d+1-a}) + M_a - N_b \\ &= 0 - N_a - N_b = -\Big[\frac{a\{a-(d+1)\}}{2(d+1)} + \frac{b\{b-(d+1)\}}{2(d+1)}\Big] \\ &= \frac{a+b}{2} - \frac{a^2+b^2}{2(d+1)} < \frac{a+b}{2} - \frac{(b-a)^2}{2(d+1)} \end{aligned}$$

The last inequality implies that

$$B(x_1) = -\frac{a+b}{2} + \frac{(b-a)^2}{2(d+1)} + n \cdot \pi < 0,$$

and similarly, $B(x_2) > 0$. The claim follows. ■

Lemma 3.1.19. *$\mathcal{D}$ is a flag complex.*

Proof. We first claim that if any k facets of Π are pairwise intersecting, they also have a common intersection. Assume that the k facets $\{f_1, \ldots, f_k\}$ of Π are pairwise intersecting. For any facet f_i, there is a partition $(S_i, [d+1] \setminus S_i)$ associated to it. By Lemma 3.1.8, we have that either $S_i \subset S_j$ or $S_j \subset S_i$ for each $i \neq j$. This means that the S_i are totally ordered, that means, there exists an ordering π of $\{1, \ldots, k\}$ such that $S_{\pi(1)} \subset S_{\pi(2)} \subset \ldots \subset S_{\pi(k)}$. Now, the partition

$$\Big(S_{\pi(1)}, S_{\pi(2)} \setminus S_{\pi(1)}, S_{\pi(3)} \setminus S_{\pi(2)}, \ldots, S_{\pi(k)} \setminus S_{\pi(k-1)}, [d+1] \setminus S_{\pi(k)}\Big)$$

is a common refinement of all the partitions, which implies that the corresponding face is incident to all k facets. This shows that the k facets have a common intersection.

The lemma follows directly with this claim and Lemma 3.1.18: consider $k+1$ vertices of $\mathcal{D}$, for which each pair has an edge in $\mathcal{D}$. We can assume that one point is the origin, and the other k points are the centers of permutahedra that intersect Π in a facet, without loss of generality. By the contrapositive of Lemma 3.1.18, all these facets have to intersect pairwisely, because all vertices have pairwise Delaunay edges. By the auxiliary claim, there is some common vertex of Π to all these facets, and its dual Delaunay simplex contains the k-simplex spanned by the vertices. ■

Lemma 3.1.20. *The shortest Voronoi vector of the A_d^* lattice has length $\sqrt{\frac{d}{d+1}}$.*

Proof. Recall that the Voronoi vectors of the A_d^* lattice are permutations of the vectors

$$v_t = \frac{1}{(d+1)}(\underbrace{t,\ldots,t}_{d+1-t},\underbrace{t-(d+1),\ldots,t-(d+1)}_{t}), t = 1\ldots d.$$

The lengths of the vectors are of the form

$$|v_t| = \frac{1}{d+1}\sqrt{t^2(d+t-1)+(d+1-t)^2 t} = \sqrt{\frac{t(d+1-t)}{d+1}},$$

which is minimum for $t=1$ and $t=d$, so $|v_1| = |v_d| = \sqrt{\frac{d}{d+1}}$ is the shortest length of any Voronoi vector. ■

For any $\beta > 0$, by scaling the lattice vectors of the A_d^* lattice by β, we get a scaled A_d^* lattice. The Voronoi cells of this scaled lattice are permutahedra which scaled by β. We show an additional property for scaled permutahedra:

Lemma 3.1.21. *Let π and π' denote the permutahedral cells at the origin at scales β and β', respectively where $\beta > \beta' > 0$. Then,*

- *$\pi' \subset \pi$, and*
- *the minimum distance between any facet of π' and any facet of π is at least $\frac{(\beta-\beta')}{2}\sqrt{\frac{d}{d+1}}$.*

In particular, this implies that the Minkowski sum[3] of π' with a ball of radius $\frac{(\beta-\beta')}{2}\sqrt{\frac{d}{d+1}}$ (with the ball's center being the reference point) lies within π.

Proof. The first claim, $\pi' \subset \pi$, follows since both permutahedra are scalings of a convex object centered at the origin.

For the second claim, consider any Voronoi vector v of the standard A_d^* lattice. The corresponding vectors at scales β and β' are $v\beta$ and $v\beta'$, respectively. Let f and f' be facets of π and π', corresponding to $v\beta$ and $v\beta'$, respectively. Then f and f' lie in parallel hyperplanes, which are separated by distance $|(v\beta - v\beta')/2| = |v|(\beta-\beta')/2$. From Lemma 3.1.20, we know that the shortest Voronoi vector has length $\sqrt{\frac{d}{d+1}}$ for the standard A_d^* lattice. This quantity scales linearly for any scaling of the lattice. This means that the minimal distance between facets of the form f, f' is $\delta := \frac{(\beta-\beta')}{2}\sqrt{\frac{d}{d+1}}$.

Let f' be any facet of π' and, g be any facet of π. Then there is a facet g' of π' which is a scaled version of g. Let H be the supporting hyperplane of g'. Since π' is convex, f' lies in the half-space of H (on H if $f' = g'$) containing the origin. On the other hand, g lies in other half-space. Moreover, g is at a distance at least δ from g'. Therefore, f' is separated from g by distance at least δ. This is true for any choice of f' or g, so the second claim follows. ■

---

[3] The Minkowski sum of two sets $X, Y \subset \mathbb{R}^d$ is the set $X \oplus Y = \{x+y \mid x \in X, y \in Y\}$.

3.1.3 Closest point in A^*

Definition 3.1.22 (closest vector problem). *The* closest vector problem (CVP) *asks for a given lattice $L \subset \mathbb{R}^d$ and a point $x \in \mathbb{R}^d$, what is (are) the lattice point(s) $y \in L$ that minimizes $\|x - y\|$?*

In other words, CVP asks for the point(s) $y \in L$ such that $x \in Vor_L(y)$. While it is known that it is NP-hard to solve this problem in general [DKRS03], there are efficient algorithms for some lattices such as the A_d, A_d^* lattices. We first detail an algorithm which solves CVP for the A_d lattice [CSB87]. Then we make use of the algorithm to solve CVP for the A_d^* lattice [CSB87].

Let $z(u) : \mathbb{R} \to \mathbb{Z}$ denote a function, which takes $u \in \mathbb{R}$ to the closest integer. In case of ties, $z(u)$ is the integer with the smaller absolute value. We also define another function $\delta : \mathbb{R} \to [-1/2, 1/2]$ as $\delta(u) := z(u) - x$.

Algorithm 2 $CVP_{A_d}(x)$

```
For $x = (x_0, \ldots, x_d) \in HP$, compute the point $z(x) = (z(x_0), \ldots, z(x_d))$ and let $y = \sum_{i=0}^{d} z(x_i)$.
if $y = 0$ then
  return $z(x)$.
else
  Sort the vector $x$ in ascending order of the values of $\delta(x_i)$, to get an arrangement
      $$-\frac{1}{2} \leq \delta(x_{i_0}) \leq \ldots \leq \delta(x_{i_d}) \leq \frac{1}{2}$$
  where $(i_0, \ldots, i_d)$ is a permutation of $(0, \ldots, d)$.
  if $y > 0$ then
    Subtract one from the co-ordinates $z(x_{i_0}), \ldots, z(x_{i_{y-1}})$ of $z(x)$ to get a point $z'(x) \in HP$.
    return $z'(x)$.
  else
    Add one to the co-ordinates $z(x_{i_{d-y+1}}), \ldots, z(x_{i_d})$ of $z(x)$ to get a point $z'(x) \in HP$.

    return $z'(x)$.
  end if
end if
```

In Algorithm 2, x is a point on the hyperplane HP and $z(x)$ is the closest integer point of $\mathbb{Z}^{d+1}$ to x. If $y = \sum_{i=0}^{d} z(x_i) = 0$, then $z(x)$ lies in HP, so $z(x)$ is the closest point of A_d to x. If $y > 0$, then $z(x)$ does not lie in HP. The steps in the algorithm find an integer point on HP by subtracting 1 from the $|y|$ smallest components of $z(x)$. The steps make the least changes to the norm of $\|z(x) - x\|$ to make y vanish, so $\|z'(x) - x\|$ is indeed the second closest integer point to x, and the closest integer point of HP. A similar argument applies for the case $y < 0$, to show that the algorithm works correctly. The algorithm can be implemented to run in $O(d)$ time [CSB87, Chapter 20].

Let Φ be an algorithm that returns the closest point of a lattice L. Let v be a vector and $L + v$ denote the set of points

$$L + v := \{l + v \mid l \in L\}.$$

With a simple transformation, Φ can be used to find the closest point for $L + v$. Given a query point x,

$$\Phi(x - v) + v$$

is the closest point of $L + v$ to x [CSB87].

Let $\{v_1, \ldots, v_d\}$ be the standard basis of A_d^* and let v_0 be the vector at the origin. In [CSB87], it was shown that

$$A_d^* = \{\cup_{i=0}^{d} A_d + v_i\}.$$

This immediately gives an algorithm to solve CVP for A^*: let x be the query point. Calculate the set of points $\{p_0, \ldots, p_d\}$ such that

$$p_i = CVP_{A_d}(x - v_i) + v_i, \forall i.$$

The closest point of A^* to x is the point p_i, that minimizes $\|p_i - x\|$. Overall, the algorithm runs in $O(d^2)$ time.

3.2 Shifted grids and cubes

In this section, we take a look at simple modifications of the integer lattice.

We denote by $I := \{\alpha_s := \lambda 2^s \mid s \in \mathbb{Z}\}$ with $\lambda > 0$, a discrete set of scales. For each scale in I, we define grids which are scaled and translated (shifted) versions of the integer lattice inductively:

Definition 3.2.1 (scaled and shifted grids). *For each scale $\alpha_s \in I$, we define the* scaled and shifted grid G_{α_s} *(or simply G_s) as:*

- *For $s = 0$, G_s is simply the scaled integer grid $\lambda\mathbb{Z}^d$, where each basis vector has been scaled by λ.*
- *For $s \geq 0$, we choose an arbitrary $O_s \in G_s$ and define*

$$G_{s+1} = 2(G_s - O_s) + O_s + \frac{\alpha_s}{2}(\pm 1, \ldots, \pm 1), \tag{3.1}$$

  where the signs of the components of the last vector are chosen independently and uniformly at random (and the choice is independent for each s).
- *For $s \leq 0$, we define*

$$G_{s-1} = \frac{1}{2}(G_s - O_s) + O_s + \frac{\alpha_{s-1}}{2}(\pm 1, \ldots, \pm 1), \tag{3.2}$$

  where the last vector is chosen as in the case of $s \geq 0$.

It is then easy to check that Equation (3.1) and Equation (3.2) are consistent at $s = 0$. A simple example of the above construction is the sequence of grids with $G_s := \alpha_s \mathbb{Z}^d$ for even s, and $G_s := \alpha_s \mathbb{Z}^d + \frac{\alpha_{s-1}}{2}(1, \ldots, 1)$ for odd s.

Next, we motivate the shifting of the grids. Using Definition 3.1.2, we see that for any point $x \in G_s$, $Vor_{G_s}(x)$ is a cube of side length α_s centered at x. For shorter notation, we write $Vor_s(x)$ instead of $Vor_{G_s}(x)$. The shifting of the grids ensures that each $x \in G_s$ lies in the Voronoi region of a unique $y \in G_{s+1}$. Using an elementary calculation, we show a stronger statement:

Lemma 3.2.2. *Let $x \in G_s, y \in G_{s+1}$ be such that $x \in Vor_{s+1}(y)$. Then,*

$$Vor_s(x) \subset Vor_{s+1}(y).$$

Proof. Without loss of generality, we can assume that $\alpha_s = 2$ and x is the origin, using an appropriate translation and scaling. Also, we assume for the sake of simplicity that $G_{s+1} = 2G_s + (1, \ldots, 1)$; the proof is analogous for any other translation vector. In that case, it is clear that $y = (1, \ldots, 1)$. Since $G_s = 2\mathbb{Z}^d$, the Voronoi region of x is the set $[-1, 1]^d$. Since G_{s+1} is a translated version of $4\mathbb{Z}^d$, the Voronoi region of y is the cube $[-1, 3]^d$, which covers $[-1, 1]^d$. The claim follows. ■

3.2.1 Cubical complexes

The integer grid $\mathbb{Z}^d$ naturally defines a cubical complex, where each element is an axis-aligned, k-dimensional cube with $0 \leq k \leq d$. To define it formally,

Definition 3.2.3 (faces of $\mathbb{Z}^d$). *Let $\square$ denote the set of all integer translates of faces of the unit cube $[0, 1]^d$, considered as a convex polytope in $\mathbb{R}^d$. We call the elements of $\square$* faces of $\mathbb{Z}^d$*.*

Each face has a dimension k; the 0-faces, or vertices are exactly the points in $\mathbb{Z}^d$. The facets of a k-face E are the $(k-1)$-faces contained in E. We call a pair of facets of E as *opposite facets*, if they are disjoint. Naturally, these concepts carry over to scaled and shifted versions of $\mathbb{Z}^d$, so we define $\square_s$ as the cubical complex defined by G_s.

We define a map $g_s : \square_s \to \square_{s+1}$ as follows: for vertices of $\square_s$, we assign to $x \in G_s$ the (unique) vertex $y \in G_{s+1}$ such that $x \in Vor_{s+1}(y)$ (see Lemma 3.2.2). For a k-face f of $\square_s$ with vertices $(p_1, \ldots, p_{2^k})$ in G_s, we set $g_s(f)$ to be the convex hull of $\{g_s(p_1), \ldots, g_s(p_{2^k})\}$; the next lemma shows that this is a well-defined map. In this thesis, we sometimes call g_s a *cubical map*, since it is a counterpart of simplicial maps for cubical complexes.

Lemma 3.2.4. *$\{g_s(p_1), \ldots, g_s(p_{2^k})\}$ are the vertices of a face e of G_{s+1}. Moreover, if e_1, e_2 are any two opposite facets of e, then there exists a pair of opposite facets f_1, f_2 of f such that $g_s(f_1) = e_1$ and $g_s(f_2) = e_2$.*

Proof. **First claim:** We prove the first claim by induction on the dimension of faces of G_s. Base case: for vertices, the claim is trivial using Lemma 3.2.2. Induction case: let the claim hold true for all $(k-1)$-faces of G_s. We show that the claim holds true for all k-faces of G_s.

Let f be a k-face of G_s. Let f_1 and f_2 be opposite facets of f, along the m-th coordinate. Let the vertices of f_1 be $(p_1, \ldots, p_{2^{k-1}})$ and f_2 be $(p_{2^{k-1}+1}, \ldots, p_{2^k})$ taken in the same order, that is, p_j and $p_{2^{k-1}+j}$ differ in only the m-th coordinate for all $1 \leq j \leq 2^{k-1}$. By definition, all vertices of f_1 share the m-th coordinate, and we denote coordinate of these vertices by z. Then, the m-th coordinate of all vertices of f_2 equals $z + \alpha_s$. By induction hypothesis, $e_1 = g_s(f_1)$ and $e_2 = g_s(f_2)$ are two faces of G_{s+1}. We show that the vertices of $e_1 \cup e_2$ are vertices of a face e of G_{s+1}.

The map g_s acts on each coordinate direction independently. Therefore, $g_s(p_j)$ and $g_s(p_{2^{k-1}+j})$ have the same coordinates, except possibly the m-th coordinate. This further implies that e_2 is a translate of e_1 along the m-th coordinate.

There are two cases: if e_1 and e_2 share the m-th coordinate, then $e_1 = e_2$ and therefore $g_s(f) = e_1 = e_2 = e$, so the claim follows. On the other hand, if e_1 and e_2 do not share the m-th coordinate: e_1's m-th coordinate is $g_s(z)$, while for e_2 it is $g_s(z + \alpha_s)$. From the structure of g_s, we see that $g_s(z)$ and $g_s(z + \alpha_s)$ differ by α_{s+1}. It follows that e_1 and e_2

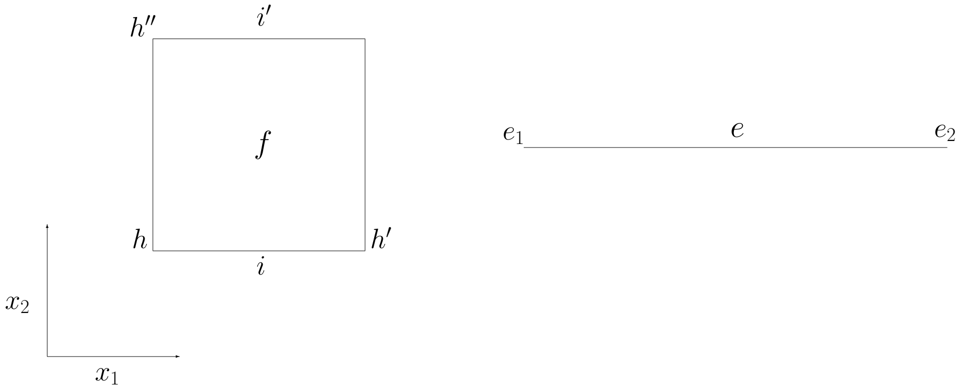


Figure 3.3: The face f is a square, for which $g(f) = e$ is a line segment. The horizontal and vertical directions are x_1 and x_2 respectively. The rest of the labels are self-explanatory in relation to the proof of Lemma 3.2.4.

are two faces of $\Box_{s+1}$ which differ in only one coordinate by α_{s+1}. So they are opposite facets of a codimension-1 face e of G_{s+1}. Using induction, the claim follows.

Second claim: Without loss of generality, assume that x_1 is the direction in which e_2 is a translate of e_1. Let h denote the maximal face of f such that $g_s(h) = e_1$. Clearly, $h \neq f$, since that would imply $g_s(f) = e_1 = e$, which is a contradiction. See Figure 3.3 for a simple illustration.

Suppose h has dimension less than $k - 1$. Let h' be the face obtained by translating h along x_1. As in the first claim, it is easy to see that $g_s(h') = e_2$, from the structure of g_s. This means that there is a facet i of f containing h and h' such that $g_s(i) = e$. Let i' be the opposite facet of i in f and let x_2 be the direction which separates i from i'. Then, $g_s(i') = e$ otherwise $g_s(f) = e$ does not hold. Let h'' be the face of f, obtained by translating h along x_2. Then, from the structure of g_s, $g_s(h'') = e_1$ holds. The facet of f containing h and h'' also maps to e_1 under g_s. This is a contradiction to our assumption that h is the highest dimensional face of f such that $g_s(h) = e_1$.

Therefore, the only possibility is that h is a facet f_1 of f such that $g_s(f_1) = e_1$. Let f_2 be the opposite facet of f_1. From the structure of g_s, it is easy to see that $g_s(f_2) = e_2$. The claim follows. ■

3.2.2 Barycentric subdivision

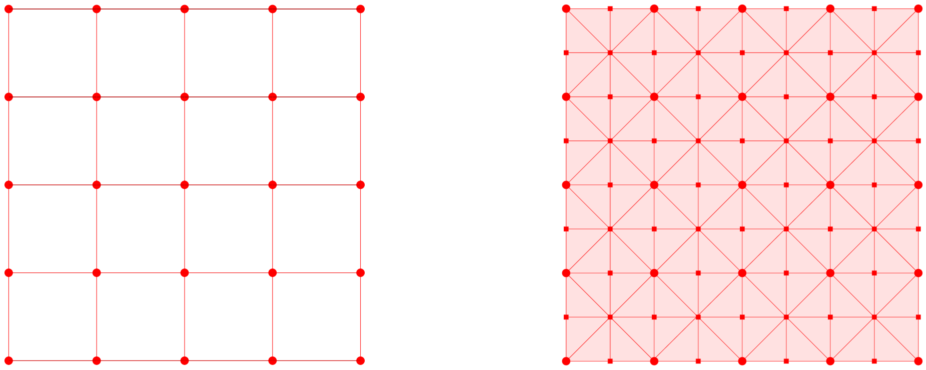
We discuss a special triangulation of $\Box_s$.

Definition 3.2.5 (flags). *A* flag *in* $\Box_s$ *is a set of faces* $\{f_0, \dots, f_k\}$ *of* $\Box_s$ *such that*

$$f_0 \subseteq \dots \subseteq f_k.$$

Definition 3.2.6 (barycentric subdivision). *The* barycentric subdivision *of* $\Box_s$*, denoted by* sd_s*, is the (infinite) simplicial complex whose simplices are the flags of* $\Box_s$*.*

In particular, the 0-simplices of sd_s are the faces of $\Box_s$. An equivalent geometric description of sd_s can be obtained by defining the 0-simplices as the barycenters of the faces in sd_s, and introducing an k-simplex between $(k+1)$ barycenters if the corresponding faces form a flag. For a simple example, see Figure 3.4. It is easy to see that sd_s is a flag complex. Given a face f in $\Box_s$, we write $sd(f)$ for the subcomplex of sd_s consisting of all flags that are formed only by faces contained in f.



(a) A portion of the grid in two dimensions. The dots are the grid points which form the 0-faces of the cubical complex.

(b) The barycentric subdivision of the grid. The tiny squares are barycenters of the 1-faces and 2-faces of the cubical complex.

Figure 3.4: Barycentric subdivision in two dimensions.

3.3 Intrinsic dimension

We now turn to a notion of intrinsic dimension of metric spaces.

3.3.1 Doubling dimension

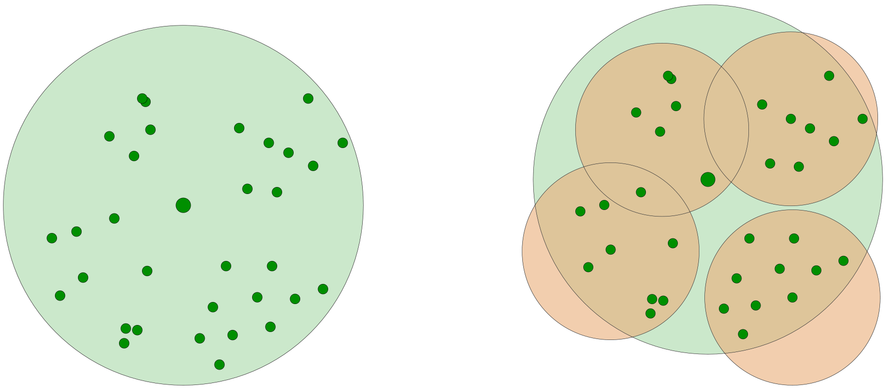
Often in practice, a metric space may lie in an ambient space of high dimension, but it may have a low-dimensional structure. A common example of such a space is a low-dimensional manifold embedded in high-dimensional Euclidean space. In such cases, the ambient space does not capture the real dimension of the manifold. We discuss a notion which correctly captures the intrinsic dimension.

Definition 3.3.1 (discrete ball). *For a metric space X, a* discrete ball *of radius r centered at a point $x \in X$ is the collection of points $Y \subseteq X$ that satisfy $\|x - y\| \leq r$ for all $y \in Y$.*

Definition 3.3.2 (doubling dimension). *For a metric space X, the* doubling constant *λ_X is the smallest integer such that for all $x \in X$ and all $r > 0$, the discrete ball of radius r centered at x is covered by at most λ_X discrete balls of radius $r/2$.*

The doubling dimension *of X, denoted by $\Upsilon(X)$ is $\lceil \log_2 \lambda_X \rceil$. A family of metrics is called* doubling *if its doubling dimension is bounded.*

A simple example of covering by balls of half the radius is shown in Figure 3.5. For $\mathbb{R}^d$ space with the ℓ_p norm, the doubling dimension is $\Theta(d)$ [GKL03]. For any finite metric space X, it is easy to see that $\Upsilon(X) = O(\log |X|)$. The doubling dimension of a metric depends on its structure, and can be upper bounded using the dimension of the space in which it is embedded. For example, a point set that is sampled from a k-dimensional subspace of $\mathbb{R}^d$ has a doubling dimension of $\Theta(k)$, and since $k \leq d$, it is upper bounded by $\Theta(d)$, the doubling dimension of $\mathbb{R}^d$. In contrast, the d boundary points of the standard $(d-1)$-simplex form a doubling space of dimension $\lceil \log_2 d \rceil$. Even worse, we can construct a subset of doubling dimension $\Theta(d)$ by placing $2^{\Theta(d)}$ points inside the unit ball in $\mathbb{R}^d$ such that any two points have a distance of at least $3/2$ (the existence of such a point set follows from a simple volume argument).



(a) The green dots are the input points. The shaded disk is centered at the (enlarged) dot.

(b) The points in the green ball are covered by 4 balls (shaded orange) of half the radius.

Figure 3.5: An illustration of covering balls.

3.3.2 Nets and Net-trees

We discuss a data structure which represents a metric space in terms of hierarchical clusters, while capturing the doubling dimension of the space. This makes it convenient to design algorithms which utilize the structure of the clusters, and ensures that the algorithms have complexities which are dominated by the doubling dimension, instead of the ambient dimension.

Definition 3.3.3 (spread). *Let X be any finite metric space. We denote by $diam(X)$ the diameter of X and by $CP(X)$ the closest-pair distance in X. The* spread *of X is the ratio*

$$\Delta_X = \frac{diam(X)}{CP(X)}.$$

Definition 3.3.4 (nets). *For any metric space X, we call a subset $\mathcal{N}_{\alpha,\beta} \subseteq X$ an (α, β)-*net*, if all points of X are in distance at most α from some point in $\mathcal{N}_{\alpha,\beta}$ and the minimum distance between any pair of points in $\mathcal{N}_{\alpha,\beta}$ is at least β.*

Usually, α and β are coupled, that is, $\beta = \Theta(\alpha)$, in which case we simply call $\mathcal{N}_{\alpha,\beta}$ as a net at scale α for simplicity. Informally, nets are subsets which capture the geometry of the metric space at a given resolution.

Definition 3.3.5 (net-tree). *Given a finite metric space X, we represent a nested sequence of nets on X for increasing scales using a rooted tree T, called the* net-tree *[HPM06]. $\mathcal{T}$ has n leaves, each of which represents a point of X, and each internal node of T has at least two children. Every tree-node v*

- *represents the subset of points given by the sub-tree rooted at v and we denote this set by P_v.*
- *has a representative,* $\mathrm{rep}_v \in P_v$ *that equals the representative of one of its children if v is not a leaf.*
- *is associated with an integer $\ell(v)$ called the* level *of v which satisfies $\ell(v) < \ell(\mathrm{par}(v))$, where* $\mathrm{par}(v)$ *is the parent of v in the tree.*

We denote by $\mathrm{scale}_v := \frac{2\tau^{\ell(v)+1}}{\tau-1}$*, where* $\tau = 11$ *is a constant. Finally, each node satisfies the following properties:*

- covering *:*
$$P_v \subseteq \mathbb{B}(\mathrm{rep}_v, \frac{2\tau}{\tau-1}\tau^{\ell(v)}) = \mathbb{B}(\mathrm{rep}_v, \mathrm{scale}_v).$$
- packing *:*
$$P_v \supseteq P \bigcap \mathbb{B}(\mathrm{rep}_v, \frac{\tau-5}{2\tau(\tau-1)}\tau^{\ell(\mathrm{par}(v))}) = \mathbb{B}(\mathrm{rep}_v, \frac{\tau-5}{4\tau^2}\mathrm{scale}_{(\mathrm{par}(v))}).$$

where $\mathbb{B}(p, r)$ *denotes the discrete ball centered at* p *with radius* rt*.*

The covering and packing properties ensure that each node v has at most $\lambda_X^{O(1)}$ children. Moreover, for any $\alpha \geq 0$, a net at a desired scale can be accessed from the net-tree simply by collecting the representatives of the nodes of a suitable scale: collecting the representatives

$$\mathcal{N}(\ell) = \{\mathrm{rep}_v \mid \ell(v) < \ell \leq \ell(\mathrm{par}(v)\}$$

gives a $(4\tau^{\ell}, \frac{\tau^{\ell-1}}{4})$-net.

There are algorithms to construct a net-tree on X in $2^{O(\Upsilon)} n \log n \log \Delta_X$ time deterministically, or $2^{O(\Upsilon)} n \log n$ time in expectation [HPM06], where $|X| = n$. The net-tree construction is oblivious to knowing the value of Υ, and in fact the constructed net-tree can be used to get a constant approximation of Υ [HPM06].

The net-tree can be *augmented* to maintain for each node $u \in T$, a list of close-by nodes with similar diameter. Specifically, for each node $u \in T$ the data structure maintains the set

$$\begin{aligned}\mathrm{Rel}(u) := \{v \in T \mid \ell(v) \leq \ell(u) < \ell(\mathrm{par}(v)) \text{ and} \\ \|\mathrm{rep}_u - \mathrm{rep}_v\| \leq 14\tau^{\ell(u)}\}.\end{aligned}$$

Informally, net-trees can be interpreted as a generalization of the well-known quad-trees to general metric spaces. Several algorithms which are applicable to quad-trees have been adapted to work with net-trees.

3.3.3 Pair decompositions

An n-point metric space X has $O(n^2)$ pairwise distances. By allowing for an arbitrarily small multiplicative error in the pairwise distances, the collection of distances of X can be represented using just linear space in n. We discuss a well-known data structure which achieves this.

Definition 3.3.6 (well-separation). *Let* X *be a finite metric space and let* $\varepsilon > 0$ *any value. Let* $A, B \subset X$ *be disjoint subsets of* X*. The pair* (A, B) *is said to be* ε-well-separated*, if*

$$max(diam(A), diam(B)) \leq \varepsilon d(A, B),$$

where $d(A, B)$ *is the minimum separation between points of* A *and points of* B*.*

Definition 3.3.7 (well-separated pair decomposition). *Let* X *be a finite metric space and* $\varepsilon > 0$*. An* ε-Well-Separated Pair Decomposition (WSPD) *[CK95] of* X *consists of pairs of the form* $(A_i, B_i) \subset X$ *such that*

- *each (A_i, B_i) is ε-well-separated, and*
- *for each pair of points $p, q \in X$, there exists a pair (A_j, B_j) in the WSPD such that either $(p \in A_j, q \in B_j)$ or $(p \in B_j, q \in A_j)$. That means, a WSPD covers each pair of points of P.*

Let W be an ε-WSPD on X. For each pair $(A, B) \in W$, we denote by $P_A \subset X$ the set of points of A and by $P_B \subset X$ denote the set of points of B. We select a representative point for A, which we call $rep(A)$, by taking an arbitrary point $rep(A) \in P_A$. Similarly, we select a representative $rep(B) \in P_B$ for B. For the pair (A, B), we denote the distance between the representatives by $\hat{d}(A, B) := \|rep(A) - rep(B)\|$. With this, it follows that $d(A, B) \leq \hat{d}(A, B) \leq d(A, B) + diam(A) + diam(B)$, which can be simplified to

$$d(A,B) \leq \hat{d}(A,B) \leq d(A,B)(1+2\varepsilon), \text{ or } \quad \frac{\hat{d}(A,B)}{1+2\varepsilon} \leq d(A,B) \leq \hat{d}(A,B).$$

For Euclidean spaces, an ε-WSPD of size at most $n(1/\varepsilon)^{O(d)}$ can be computed in time $n \log n 2^{O(d)} + n(1/\varepsilon)^{O(d)}$ using quad-trees (see, for instance [CK95, HP11, Smi07]). In metric spaces with doubling dimension Υ, a net-tree can be used to compute an ε-WSPD of size at most $n(1/\varepsilon)^{O(\Upsilon)}$ in $n \log n 2^{O(\Upsilon)} + n(1/\varepsilon)^{O(\Upsilon)}$ time [HPM06].

While WSPDs are useful in many applications, they can pose a drawback. For a WSPD W on an n-point metric space, the quantity $\sum_{(A_i,B_i)\in W}(|A_i| + |B_i|)$, which is the total weight of the pairs involved, can be as large as $\Omega(n^2)$ [AHP12]. To overcome this, a related concept is used sometimes:

Definition 3.3.8 (semi-separation). *Let X be a finite metric space and let $\varepsilon > 0$. Let $A, B \subset X$ be disjoint subsets of X. The pair (A, B) is said to be ε-*semi-separated*, if*

$$min(diam(A), diam(B)) \leq \varepsilon d(A, B).$$

This is a weaker notion than well-separation, because it only requires that the *smaller* diameter of the participating sets be small compared to the distance between them.

Definition 3.3.9 (semi-separated pair decomposition). *For a finite metric space X and any $\varepsilon > 0$, an ε-*Semi-Separated Pair Decomposition (SSPD) *[AHP12, Var98] of X consists of pairs of the form $(A_i, B_i) \subset X$ such that*

- *each (A_i, B_i) is ε-semi-separated, and*
- *for each pair of points $p, q \in X$, there exists a pair (A_j, B_j) in the SSPD such that either $(p \in A_j, q \in B_j)$ or $(p \in B_j, q \in A_j)$.*

The main advantage of using a SSPD over a WSPD is reduced weight: an ε-SSPD of expected weight $\varepsilon^{-O(\Upsilon)} n \log n$ can be calculated in $\varepsilon^{-O(\Upsilon)} n \log n$ expected time [AHP12], for a n-point metric space with doubling dimension Υ.

3.4 Locality-sensitive hashing

In this section, we discuss a technique which can answer near-neighbor queries for points in Euclidean spaces efficiently. First, we define the problem formally:

Definition 3.4.1 ((r, c)-nearest neighbor problem). *Let $S \subset \mathbb{R}^d$ be a set of n points. For a given query point $q \in \mathbb{R}^d$, the (r, c)-*nearest neighbor query *returns any point of S in distance at most cr from q, if there exists a point in distance at most r from q, where $r > 0, c \geq 1$.*

To tackle the problem, we use the notion of *locality-sensitive hashing* (LSH), which was introduced by [IM98] for the Hamming metric, and was extended to Euclidean spaces in [DIIM04]. LSH is a popular approach to find approximate near-neighbors in high dimensions because of its near-linear complexity in n and d. Informally, LSH collects points of S in a data structure U such that close-by points are stored together with a high probability. This makes answering near-neighbor queries efficient. The scheme relies on the use of hash functions, which we describe next.

Definition 3.4.2 ((r_1, r_2, p_1, p_2)-sensitive hash functions). *A family of hash functions $\mathcal{H} = \{h : S \to U\}$ from S to a collection of buckets U is called* (r_1, r_2, p_1, p_2)-sensitive *if for all $a, b \in S$, the following holds:*

- $p_1 \geq p_2$ *and* $r_1 \leq r_2$,
- *if* $\|a - b\| \leq r_1, P_{r_1} := P[h(a) = h(b)] \geq p_1$, *and*
- *if* $\|a - b\| \geq r_2, P_{r_2} := P[h(a) = h(b)] \leq p_2$.

We amplify the ratio between P_{r_1} and P_{r_2} by concatenating k hash functions from a family of (r_1, r_2, p_1, p_2)-sensitive hash functions. This creates a new family of hash functions $\mathcal{G} = \{g : S \to U^k\}$ such that $g(x) = (h_1(x), \ldots, h_k(x))$ with $\{h_1, \ldots, h_k\} \in \mathcal{H}$ being independently chosen hash functions. For any $g \in \mathcal{G}$ in this family, we have the modified properties:

- if $\|a - b\| \leq r_1, P[g(a) = g(b)] \geq p_1^k$, and
- if $\|a - b\| \geq r_2, P[g(a) = g(b)] \leq p_2^k$.

We describe the LSH scheme next. The input is a n-point set $S \subset \mathbb{R}^d$ and a distance parameter $r > 0$.

- *Pre-processing*: We choose l hash functions $\{g_1, \ldots, g_l\}$ uniformly at random from $\mathcal{G}$ [DIIM04, Section 3] and hash each $p \in S$ to the buckets $g_i(p), \forall i \in [1, l]$.
- *Querying*: Given a query point $q \in \mathbb{R}^d$, the set of buckets $\{g_1(q), \ldots, g_l(q)\}$ is inspected, and for each point of S encountered, we check whether the distance to q is at most cr. As soon as we encounter such a point, we output it as the c-approximate near-neighbor of q.

To complete the description, we need to specify the parameters of LSH. The parameters should be such that

- with high probability, the output contains a point in distance cr from q.
- the buckets should have small size so that the query does not have to filter out too many false positives.

We choose the parameters p_1, p_2, r_1 and r_2 of the hashing scheme such that $\rho := \frac{\log p_1}{\log p_2} = \frac{r_1}{r_2}$ (see [DIIM04] for more details on the choice). The length of the hash functions is $k := \lceil -\log_{p_2} n \rceil$ and the number of buckets is $l := \lceil n^\rho \rceil$. With these parameters, the algorithm reports a near-neighbor correctly with probability greater than 1/2 [IM98, Theorem 5] in $O(kl)$ evaluations of the hash functions from $\mathcal{H}$.

In this thesis, we require a slight modification of the above problem:

Definition 3.4.3 (all near neighbors problem). *Given a point set $S \subset \mathbb{R}^d$ and a distance parameter $r > 0$, the r-*near neighbors *query returns all points of S at distance at most r from any query point q.*

We only consider the case when the query point $q \in S$. To achieve this, we modify the algorithm and the construction slightly. We use $l := \lceil 2n^\rho \ln \frac{n}{\sqrt{\delta}} \rceil$ where $\delta < 1$ is an arbitrarily chosen constant. Also, in the query for q, we inspect all the buckets $\{g_1(q), \ldots, g_l(q)\}$ and for each point encountered, we report it if the distance to q is at most r. The complexity of our method is summarized in the following lemma:

Lemma 3.4.4. *Let $r_1 := r$ and $r_2 := r/\rho$, $k := \lceil -\log_{p_2} n \rceil$ and $l := \lceil 2n^\rho \ln \frac{n}{\sqrt{\delta}} \rceil$ with an arbitrarily chosen constant $\delta < 1$. The near-neighbor primitive has the following properties:*

(i) With probability at least $1 - \delta$, all points in distance at most r are reported for all query points of S.

(ii) For any query point q, the expected aggregate size of the buckets $\{g_1(q), \ldots, g_l(q)\}$ is at most $l(\widetilde{C} + 1)$, where $\widetilde{C}$ is the number of points in S with distance at most r_2 to q.

(iii) The pre-processing runtime is $O(dnkl)$ and the expected query runtime for a point is $O(dl(k + \widetilde{C}))$, where $\widetilde{C}$ is defined as in (ii).

Proof. First we bound the expected aggregate size of the buckets. A bucket contains "close" points which are in distance at most r_2 from q and "far" points which are further away. However, since the probability of a far point falling in the same bucket as q is at most p_2^k, the expected size of a single bucket is at most $\widetilde{C} + np_2^k \leq \widetilde{C} + 1$ by our choice of k. Since there are l buckets, (ii) is satisfied.

For (i), fix two points $q_1, q_2 \in S$ such that $\|q_1 - q_2\| \leq r_1$. We have to ensure that $g_j(q_1) = g_j(q_2)$ for some $j \in \{1, \ldots, l\}$; this implies that q_1 will be reported for query point q_2, and vice versa for at least one of the l buckets. The probability for $g_j(q_1) = g_j(q_2)$ for a fixed j is at least p_1^k, which is $p_1^{-\log_{p_2} n} = n^{-\rho}$. Hence the probability that $g_j(q_1) \neq g_j(q_2)$ holds for all $j \in \{1, \ldots, l\}$ is at most $(1 - n^{-\rho})^l$ because we choose the hash functions uniformly at random. There are less than n^2 pairs of points within distance at most r_1. By union bound, the probability that at least one such pair maps into different buckets is at most $n^2(1 - n^{-\rho})^l$. Now we can bound

$$\begin{aligned} n^2(1 - n^{-\rho})^l &= n^2(1 - n^{-\rho})^{2n^\rho \ln \frac{n}{\sqrt{\delta}}} \\ &= n^2(1 - \frac{1}{n^\rho})^{n^\rho \ln \frac{n^2}{\delta}} \\ &\leq n^2 e^{-\ln \frac{n^2}{\delta}} = \delta, \end{aligned}$$

where we used the fact that $(1 - 1/x)^x \leq 1/e$ for all $x \geq 1$. It follows that the probability that all pairs of points in distance at most r_1 fall in at least one common bucket is at least $1 - \delta$. This implies (i).

It remains to show (iii): in the pre-processing step, we have to compute kl hash functions for n points. Computing the hash value for a point p, $h_i(p)$ takes $O(d)$ time [DIIM04, Section 3.2]. For a query, we have to identify the buckets to consider in $O(dkl)$ time and then iterate through the (expected) $l(\widetilde{C} + 1)$ candidates (using (ii)), spending $O(d)$ for each. ■

3.5 Dimension reduction

In this section, we discuss about dimension reduction techniques for Euclidean spaces. Given a finite point set in $\mathbb{R}^d$, a natural question is whether it can be embedded in a lower dimensional space while preserving some geometric properties. We begin by defining such a notion.

Definition 3.5.1 (distortion). *Let* $P \subset \mathbb{R}^d$ *be a point set. Let* $f : P \to \mathbb{R}^m$ *be an injective map, such that there exist numbers* $\xi_1, \xi_2 \geq 0$ *satisfying*

$$\xi_1 \|x - y\| \leq \|f(x) - f(y)\| \leq \xi_2 \|x - y\|$$

for all $x, y \in P$*. Then, the* distortion *of the map* f *is* ξ_2/ξ_1*.*

Informally, the distortion measures the changes in the pairwise distances of P, when embedded into $\mathbb{R}^m$. In general, it is not possible to have an isometric embedding of P into a lower-dimensional space, that is, a map with distortion equal to unity is not possible in general. However, allowing for a small distortion makes it possible to reduce the dimension significantly. We mention the most celebrated result in this area.

Theorem 3.5.2 (Johnson-Lindenstrauss lemma [JLS86]). *Let* $P \subset \mathbb{R}^d$ *be any set of* n *points. For any* $0 < \varepsilon < 1$*, there is a one-to-one map* $f : P \to \mathbb{R}^k$ *such that*

$$(1 - \varepsilon)\|x - y\| \leq \|f(x) - f(y)\| \leq (1 + \varepsilon)\|x - y\|$$

for all $x, y \in P$ *and* $k := \lambda \log n/\varepsilon^2$*, where* λ *is an absolute constant.*

In fact, a random orthogonal projection of P into a subspace of dimension k yields such a map with high probability. The result shows that for point clouds in arbitrary dimensions, a random projection can greatly reduce the dimension while still preserving pairwise distances to a high precision. A more general result is due to Matoušek [Mat90]:

Theorem 3.5.3. *Let* P *be an* n*-point set in* $\mathbb{R}^d$*. Then, a random orthogonal projection into* $\mathbb{R}^k$ *for* $3 \leq k \leq C \log n$ *distorts pairwise distances in* P *by at most* $O(n^{2/k}\sqrt{\log n/k})$*. The constants in the bound depend only on* C*.*

Random projections also preserve the radius of minimum enclosing balls:

Lemma 3.5.4 (distortion of minimum enclosing balls [KR15]). *Let* $P \subseteq \mathbb{R}^d$ *be a set of* n *points and* $f : P \to \mathbb{R}^m$ *with* $m = \Theta(\log n/\varepsilon^3), \varepsilon \leq 1/2$ *be a random projection. For each subset* $S \subseteq P$*,*

$$(1 - 2\varepsilon)\mathrm{rad}(S) \leq \mathrm{rad}(f(S)) \leq (1 + 2\varepsilon)\mathrm{rad}(S).$$

3.5.1 Dimension reduction for towers

The following statement is a simple application of interleaving distances from Section 2.3.

Lemma 3.5.5 (Dimension reduction for Rips filtration). *Let* $f : P \to \mathbb{R}^m$ *be an injective map such that*

$$\xi_1 \|p - q\| \leq \|f(p) - f(q)\| \leq \xi_2 \|p - q\|$$

for some constants $\xi_1 \leq 1 \leq \xi_2$*. Let* $\overline{\mathcal{R}}_\alpha$ *denote the Rips complex of the point set* $f(P)$*. Then, the persistence module* $(H(\overline{\mathcal{R}}_\alpha))_{\alpha \geq 0}$ *is an* $\frac{\xi_2}{\xi_1}$*-approximation of* $(H(\mathcal{R}_\alpha))_{\alpha \geq 0}$*.*

Proof. The map f is a bijection between P and $f(P)$. The properties of f ensure that the vertex maps f^{-1} and f, composed with appropriate inclusion maps, induce simplicial maps

$$\overline{\mathcal{R}}_{\frac{\alpha}{\xi_2/\xi_1}} \overset{\phi}{\hookrightarrow} \mathcal{R}_\alpha \overset{\psi}{\hookrightarrow} \overline{\mathcal{R}}_{\alpha\xi_2/\xi_1}.$$

It is straightforward to show that the following diagrams commute on a simplicial level,

$$\begin{tikzcd}
\mathcal{R}_{\frac{\alpha}{\beta}} \arrow[rrr, "inc"] \arrow[dr, "\psi"'] & & & \mathcal{R}_{\beta\alpha'} \\
& \overline{\mathcal{R}}_\alpha \arrow[r, "inc"] & \overline{\mathcal{R}}_{\alpha'} \arrow[ur, "\phi"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
& \mathcal{R}_{\beta\alpha} \arrow[r, "inc"] & \mathcal{R}_{\beta\alpha'} \\
\overline{\mathcal{R}}_\alpha \arrow[r, "inc"] \arrow[ur, "\phi"] & \overline{\mathcal{R}}_{\alpha'} \arrow[ur, "\phi"] &
\end{tikzcd} \tag{3.3}$$

$$\begin{tikzcd}
& \mathcal{R}_\alpha \arrow[r, "inc"] \arrow[dl, phantom] & \mathcal{R}_{\alpha'} \arrow[dr, "\psi"] & \\
\overline{\mathcal{R}}_{\frac{\alpha}{\beta}} \arrow[rrr, "inc"] \arrow[ur, "\phi"] & & & \overline{\mathcal{R}}_{\beta\alpha'}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{R}_\alpha \arrow[r, "inc"] \arrow[dr, "\psi"] & \mathcal{R}_{\alpha'} \arrow[dr, "\psi"] & \\
& \overline{\mathcal{R}}_{\beta\alpha} \arrow[r, "inc"] & \overline{\mathcal{R}}_{\beta\alpha'}
\end{tikzcd}$$

for all $0 \leq \alpha \leq \alpha'$, where $\beta = \xi_2/\xi_1$ and *inc* is the inclusion map. Hence, the strong interleaving result from Theorem 2.3.6 implies that both persistence modules are $\frac{\xi_2}{\xi_1}$-approximations of each other. ■

Suppose that for any set of n points $P \subset \mathbb{R}^d$, there exists an algorithm to construct an approximate persistence module $\mathbb{V}$, that is a $\phi(d)$-approximation of the Rips filtration of P, $\phi(d)$ being a function only dependent on d. When the ambient dimension d is large, such an approximation scheme can play nice with dimension reduction techniques, potentially improving the approximation quality and the size of the approximate tower.

To use dimension reduction in this context, there are two steps:

1. Use a dimension reduction technique to embed $P \subset \mathbb{R}^d$ into $P' \subset \mathbb{R}^m$ with some distortion γ.
2. Apply the approximation scheme on P'.

Using Lemma 3.5.5, we see that the Rips filtration of P' is a γ-approximation of the Rips filtration of P. Applying the approximation scheme on P' and using transitivity of interleavings of persistence modules, we get that $\mathbb{V}$ is a $\gamma\phi(m)$-approximation of the Rips filtration of P. Moreover, the size of the approximation depends on m rather than d. Therefore, we can potentially reduce the size of the approximation and improve the quality to a great extent. With the help of Lemma 3.5.4, this pipeline can also be extended to the case of Čech filtrations.

We mention another relevant result in this area, which concerns with embedding for general metric spaces.

Theorem 3.5.6 (Bourgain's embedding [Bou85]). *Every n-point metric space has an embedding into $\mathbb{R}^k$ with distortion $O(\log n)$, where $k = \lambda \log^2 n$ and the constant λ is independent of n.*

To use dimension reduction techniques for a general metric space, we first use Bourgain's technique to embed it into Euclidean space. Then use the pipeline as in the Euclidean case. This yields an approximation ratio of $O(\gamma\phi(m)\log n)$, where $m, \gamma, \phi(m)$ are as defined earlier.

Part I

Techniques for Euclidean Spaces

Chapter 4

Approximation using the Permutahedron

In this chapter we present the first approximation result of this thesis, which concerns with point clouds in Euclidean spaces. We show two main results in this chapter. The first is an approximation scheme for Rips filtrations, which is discussed in Sections 4.1 and 4.2. The second result is a lower bound on the size of approximations of Čech filtrations, and is detailed in Section 4.3. For both results, we make intensive use of the A^* lattice from Section 3.1 of Chapter 3.

4.1 Approximation scheme

We detail our approximation complex for any set of n points $P \subset \mathbb{R}^d$. We denote our approximation complex by X_β for any fixed scale $\beta > 0$.

Let L_β denote a scaled version of the A_d^* lattice in $\mathbb{R}^d$, where each lattice vector has been scaled by β. Recall from Section 3.1 that the Voronoi cells of these lattice points are scaled permutahedra which tile $\mathbb{R}^d$. The bounds for most quantities, including the diameter as well as for the distance between non-intersecting Voronoi cells remain valid when multiplying them with the scaling factor. Hence, any cell of L_β has diameter at most $\beta\sqrt{d}$, using Lemma 3.1.10. Moreover, any two non-adjacent cells have a distance at least $\frac{\beta\sqrt{2}}{d+1}$, from Lemma 3.1.16.

We call a permutahedron *full*, if it contains at least one point of P, and *empty* otherwise. Here, we have assumed for simplicity that each point of P lies in the interior of some permutahedron, which can be ensured with well-known methods to remove degeneracies [EM90]. Clearly, at any given scale, there are at most n full permutahedra for a given P. We define X_β as the nerve (Definition 2.1.11) of the full permutahedra defined by L_β. An equivalent formulation is that X_β is a subset of the Delaunay triangulation $\mathcal{D}$ of A^*, that is induced by the lattice points of full permutahedra. This implies that X_β is a flag complex, just like $\mathcal{D}$ (using Lemma 3.1.19). We usually identify a permutahedron at scale β by its center in L_β and interpret the vertices of X_β as a subset of L_β. See Figure 4.1 for a simple example in two dimensions.

4.1.1 Interleaving

We construct a tower using X_β at different scales. To do so, we first define simplicial maps connecting the complexes on related scales.

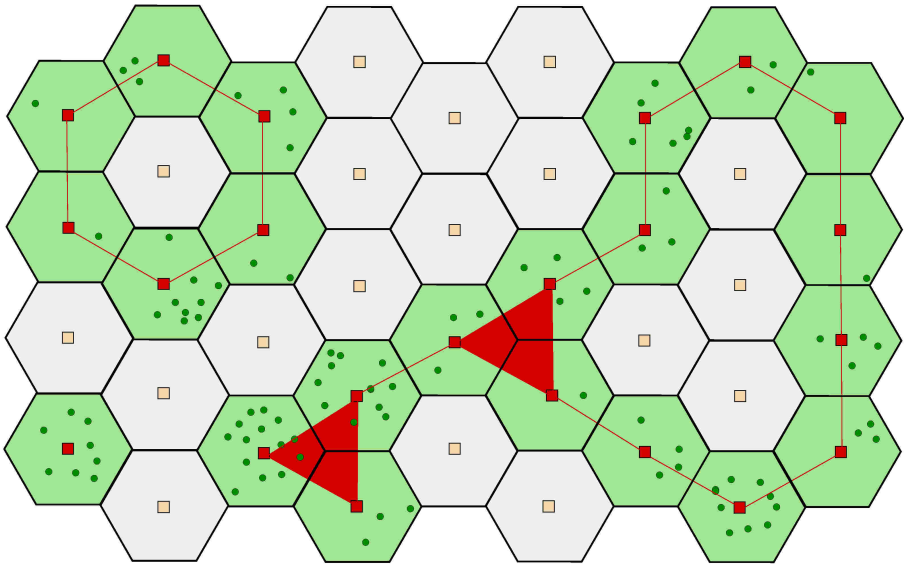


Figure 4.1: An example of X_β: the darkly shaded hexagons are the *full* permutahedra, which contain the input points marked as small, dark disks. Each dark square corresponds to the center of a full permutahedron and represents a vertex of X_β. If two full permutahedra are adjacent, there is an edge between the corresponding vertices. The clique completion on the edge graph constitutes the complex X_β.

Let V_β denote the subset of L_β corresponding to full permutahedra. We define two maps

- $v_\beta : P \to V_\beta$, which maps each point $p \in P$ to its closest lattice point, corresponding to a full permutahedron.
- $w_\beta : V_\beta \to P$ maps a vertex in V_β to the closest point of P. We assume for simplicity that this closest point is unique.

Our assumption of general position ensures that the two maps are well-defined. Note that $v_\beta \circ w_\beta$ is the identity map for V_β, while $w_\beta \circ v_\beta$ is not in general. We show that these maps establish a connection between the approximation complexes and Rips complexes on P.

Lemma 4.1.1. *The map v_β induces a simplicial map*

$$\phi_\beta : \mathcal{R}_{\frac{\beta}{\sqrt{2(d+1)}}} \to X_\beta.$$

Proof. Since X_β is a flag complex, to show that ϕ_β is simplicial, it suffices to show that for any edge $(p, q) \in \mathcal{R}_{\frac{\beta}{\sqrt{2(d+1)}}}$, $(v_\beta(p), v_\beta(q))$ is an edge of X_β. This follows at once from the contra-positive of Lemma 3.1.16. ■

Lemma 4.1.2. *The map w_β induces a simplicial map*

$$\psi_\beta : X_\beta \to \mathcal{R}_{\beta 2\sqrt{d}}.$$

Proof. It is enough to show that for any edge (p, q) in X_β, $(w_\beta(p), w_\beta(q))$ is an edge of $\mathcal{R}_{\beta 2\sqrt{d}}$, since the Rips complex is a flag complex. Note that $w_\beta(p)$ lies in the permutahedron of p and similarly, $w_\beta(q)$ lies in the permutahedron of q, so $\|w_\beta(p) - w_\beta(q)\|$ is bounded by twice the diameter of the permutahedron. Using Lemma 3.1.10, this distance is upper bounded by $\beta 2\sqrt{d}$. The claim follows. ■

Since $\beta 2\sqrt{d} < \beta 2(d+1)$, there is a natural inclusion map $inc : \mathcal{R}_{\beta 2\sqrt{d}} \to \mathcal{R}_{\beta 2(d+1)}$. We compose the map ψ_β from Lemma 4.1.2 with *inc* to get a simplicial map (which we denote by ψ_β as well for simplicity), $\psi_\beta : X_\beta \to \mathcal{R}_{\beta 2(d+1)}$.

Similarly, since $\frac{1}{2(d+1)} < \frac{1}{\sqrt{2}(d+1)}$, there is an inclusion $inc : \mathcal{R}_{\frac{1}{2(d+1)}} \to \mathcal{R}_{\frac{1}{\sqrt{2}(d+1)}}$, which we compose with ϕ_β (Lemma 4.1.1) to get a simplicial map (again, with the same name) $\phi_\beta : \mathcal{R}_{\frac{1}{2(d+1)}} \to X_\beta$.

Composing the simplicial maps ψ and ϕ, we obtain simplicial maps of the form

$$\theta_\beta : X_\beta \to X_{\beta(2(d+1))^2}$$

for any $\beta > 0$. This gives rise to the tower

$$\left(X_{\beta(2(d+1))^{2k}}\right)_{k\in\mathbb{Z}}.$$

Collecting the simplicial complexes and the simplicial maps between them, we get the following diagram (we have omitted the scale indices of the maps for readability):

$$\begin{tikzcd}
 & \cdots \arrow[r] & \mathcal{R}_{\beta 2(d+1)} \arrow[rr, "inc"] \arrow[dr, "\phi"] & & \mathcal{R}_{\beta 8(d+1)^3} \arrow[r] & \cdots, \\
\cdots \arrow[r] & X_\beta \arrow[ur, "\psi"] \arrow[rr, "\theta"] & & X_{\beta 4(d+1)^2} \arrow[ur, "\psi"] \arrow[rr] & & \cdots
\end{tikzcd} \qquad (4.1)$$

where, *inc* is the inclusion map between the corresponding Rips complexes. Applying the homology functor from Definition 2.2.4 yields a sequence of vector spaces and linear maps between them, which we represent in the following diagram:

$$\begin{tikzcd}
 & \cdots \arrow[r] & H(\mathcal{R}_{\beta 2(d+1)}) \arrow[rr, "inc_*"] \arrow[dr, "\phi_*"] & & H(\mathcal{R}_{\beta 8(d+1)^3}) \arrow[r] & \cdots, \\
\cdots \arrow[r] & H(X_\beta) \arrow[ur, "\psi_*"] \arrow[rr, "\theta_*"] & & H(X_{\beta 4(d+1)^2}) \arrow[ur, "\psi_*"] \arrow[rr] & & \cdots
\end{tikzcd}$$

(4.2)

where the asterisk subscripts denote the linear maps corresponding to the respective simplicial maps.

Lemma 4.1.3. *Diagram* (4.2) *commutes for any* $\beta > 0$*, that is,*

$$\theta_* = \phi_* \circ \psi_* \quad \textit{and} \quad inc_* = \psi_* \circ \phi_*.$$

Proof. We make use of Diagram (4.1) to prove the claim. For the first statement, since θ is defined as $\theta := \phi \circ \psi$, so the maps commute at the simplicial level. Since the construction of the persistence module is functorial, the corresponding linear maps commute as well.

The second identity is not true on a simplicial level in general. Instead, we show that the maps *inc* and $h := \psi \circ \phi$ are contiguous (see Definition 2.4.4), which means that for every simplex $(x_0, \ldots, x_k) \in \mathcal{R}_{\beta 2(d+1)}$, the vertices

$$(inc(x_0), \ldots, inc(x_k), h(x_0), \ldots, h(x_k))$$

form a simplex in $\mathcal{R}_{\beta 8(d+1)^3}$. Contiguity implies that the induced maps on the homology level, inc_* and $h_* = \psi_* \circ \phi_*$ are equal [Mun84].

For the second statement, it suffices to prove that any pair of vertices among

$$\{inc(x_0), \ldots, inc(x_k), h(x_0), \ldots, h(x_k)\}$$

is at most $\beta 16(d+1)^3$ apart. This is immediately clear for any pair $(inc(x_i), inc(x_j))$ and $(h(x_i), h(x_j))$, so we can restrict to pairs of the form $(g(x_i), h(x_j))$. Note that $inc(x_i) = x_i$ since inc is the inclusion map. Moreover, $h(x_j) = \psi(\phi(x_j))$,and $\ell := \phi(x_j)$ is the closest lattice point to x_j in $X_{\beta 4(d+1)^2}$. Since $\psi(\ell)$ is the closest point in P to ℓ, it follows that $\|x_j - h(x_j)\| \leq 2\|x_j - \ell\|$. With Lemma 3.1.10, we know that $\|x_j - \ell\| \leq \beta 4(d+1)^2\sqrt{d}$, which is the diameter of the permutahedron cell . Using triangle inequality, we obtain

$$\begin{aligned}\|inc(x_i) - h(x_j)\| &\leq \|x_i - x_j\| + \|x_j - h(x_j)\| \\ &\leq \beta 4(d+1) + \beta 8(d+1)^2\sqrt{d} \\ &< \beta 16(d+1)^3.\end{aligned}$$

The claim follows. ■

Theorem 4.1.4. *The persistence module* $\left(H(X_{\beta(2(d+1))^{2k}})\right)_{k\in\mathbb{Z}}$ *approximates the persistence module* $(H(\mathcal{R}_\beta))_{\beta\geq 0}$ *by a factor of* $6(d+1)$.

Proof. Lemma 4.1.3 proves that on the logarithmic scale, the two persistence modules are weakly $2(d+1)$-interleaved, in the sense of Definition 2.3.3. Then, Theorem 2.3.4 asserts that the two persistence modules are $6(d+1)$-approximations of each other. ■

4.2 Computational aspects

We now discuss the computational aspects of the approximation scheme. We discuss arguments to bound the size of the approximate tower in Sub-section 4.2.1. Then, we present two algorithms to compute the tower efficiently in Sub-section 4.2.2.

We utilize the non-degenerate configuration of the permutahedral tessellation (refer to Lemma 3.1.13) to prove that X_β is not too large, for any scale $\beta > 0$. In the rest of the chapter, we make no distinction between a vertex of X_β and the corresponding permutahedron, when it is clear from the context.

Theorem 4.2.1. *For any scale* $\beta > 0$, *each vertex of* X_β *has at most* $2^{O(d\log k)}$ *incident* k*-simplices. This means that the* k*-skeleton of* X_β *has at most* $n2^{O(d\log k)}$ *simplices.*

Proof. We fix $1 \leq k \leq d$ and any vertex v of X_β. v represents a permutahedron, which we denote by $\Pi(v)$. By definition, any k-simplex containing v corresponds to a common intersection of a set of $(k+1)$ permutahedra, involving $\Pi(v)$. By Proposition 3.1.14, such an intersection corresponds to a $(d-k)$-face of $\Pi(v)$. Therefore, the number of k-simplices incident to v is upper bounded by the number of $(d-k)$-faces of the permutahedron, which is $2^{O(d\log k)}$, using Lemma 3.1.9. The first claim follows.

Using the first statement, the k-skeleton incident to v has size at most

$$\sum_{i=1}^{k} 2^{O(d\log i)} = 2^{O(d\log k)}.$$

The second bound follows from the fact that at any scale, X_β has at most n vertices. ■

Range of scales

Let $CP(P)$ denote the closest-pair distance of P and $diam(P)$ the diameter of P. Then, $\Delta = \frac{diam(P)}{CP(P)}$ is the spread of the point set P (see Definition 3.3.3). At the scale $\beta_0 := \frac{CP(P)}{3d}$ and lower, no two points of P lie in adjacent cells, since the minimal separation between any two points is more than twice the diameter of the cells. Therefore, the complex at such scales consists of n isolated vertices.

At scale $\beta_m := diam(P)(d+1)$ and higher, all points of P lie in a collection of adjacent cells, by the contra-positive of Lemma 3.1.16. Therefore, the nerve of the cells at scales β_m and higher is a contractible simplicial complex (Definition 2.1.5), and hence has trivial homology in all dimensions. As a result, the persistence barcodes for scales lower than β_0 and greater than β_m are known explicitly. We restrict our attention only to the range of scales $[\beta_0, \beta_m]$ to construct the tower.

In our tower, the scales jump by a factor of $c := (2(d+1))^2$ from one scale to the next. The total number of scales to be inspected is at most

$$\begin{aligned}\lceil \log_c \beta_m/\beta_0 \rceil &= \lceil \log_c \frac{diam(P)}{CP(P)} 3d(d+1) \rceil = \lceil \log_c \Delta + \log_c 3d(d+1) \rceil \\ &\leq \lceil \log_c \Delta + 1 \rceil = O(\log \Delta).\end{aligned}$$

The scales of the tower can be written as $\beta_i = \beta_0 c^i$, for $i = 0, \ldots, m$.

In Theorem 4.2.1, we showed that the size of the k-skeleton at each scale is upper bounded by $n2^{O(d \log k)}$. Accounting for the $O(\log \Delta)$ scales, a simple upper bound on the size of the k-skeleton of the tower is then $n2^{O(d \log k)} \log \Delta$. The spread of a point set can be arbitrarily large, independent of the number of points or the ambient dimension, and this makes the upper bound unattractive.

To mitigate this undesirable dependence, we introduce a slight modification in the construction: at each scale β of the tower, we apply a random translation to the A_d^* lattice. More specifically, let π be the permutahedron at the origin at scale β. We translate the origin uniformly at random inside π, so that the lattice and the cells translate by the same amount. With this randomization, we show that the expected size of the tower is independent of the spread. More specifically, we use the random translations to bound the expected number of vertex inclusions in the tower, which leads to the main result. The expectation is taken over the random translation of the origin, and does not depend on the choice of the input. We emphasize that the selection of the origin is the only randomized part of our construction. Also, Lemma 4.1.3 and Theorem 4.2.1 hold for any unitary transformation (and in particular, any random translation) of the lattice, so the approximation results of Section 4.1 still hold true.

Critical scales

First, we discuss a concept that is helpful in bounding the number of vertex inclusions of the tower. Let W be an ε-WSPD (Definition 3.3.7) on P with $\varepsilon = \frac{1}{6d^2}$. Recall that for each subset A appearing as some pair in $(A, B) \in W$, we have a representative point $rep(A) \in P_A$, where $P_A \subset P$ is the set of points of A (similarly for B). Also, for any pair $(A, B) \in W$, the distance between the representatives is $\hat{d}(A, B) := \|rep(A) - rep(B)\|$ and satisfies

$$d(A,B) \leq \hat{d}(A,B) \leq d(A,B)(1+2\varepsilon) \quad \text{or} \quad \frac{\hat{d}(A,B)}{1+2\varepsilon} \leq d(A,B) \leq \hat{d}(A,B),$$

where $d(A, B)$ is the distance between A and B.

Definition 4.2.2 (critical scales). *For any pair* $(A, B) \in W$*, let* i *be the largest integer such that* $\hat{d}(A, B) > (1 + 2\varepsilon)\beta_i 2\sqrt{d}$*. We say that the scales* $\{\beta_{i+1}, \beta_{i+2}\}$ *are* critical *for* (A, B)*. All higher scales are* non-critical *for* (A, B)*.*

For any permutahedron π, we denote by $\mathrm{NBR}(\pi)$ the union of π and its neighboring cells, that is, those cells which are adjacent to π in the tessellation.

Lemma 4.2.3. *Let* $(A, B) \in W$ *be any WSPD pair. Let* $\{\beta < \delta\}$ *denote the critical scales for* (A, B)*.*

- *At scale* β*, let* π, π' *denote the permutahedra containing* $rep(A), rep(B)$*, respectively. Then* P_A *lies in* $\mathrm{NBR}(\pi)$*. Similarly,* P_B *lies in* $\mathrm{NBR}(\pi')$*.*
- *At scale* δ*, let* Π *denote the permutahedron that contains* $rep(A)$*. Then,* $P_A \cup P_B$ *lies in* $\mathrm{NBR}(\Pi)$*.*

Proof. For the first claim, we have $\frac{\hat{d}(A,B)}{1+2\varepsilon} \leq \beta 2\sqrt{d}$ by definition, which implies that

$$diam(A) \leq \frac{d(A, B)}{6d^2} \leq \frac{\hat{d}(A, B)}{6d^2} \leq \frac{(1 + 2\varepsilon)\beta 2\sqrt{d}}{6d^2} < \frac{\beta\sqrt{2}}{d + 1}.$$

Using Lemma 3.1.16, we get that P_A lies in $\mathrm{NBR}(\pi)$. The argument for P_B follows similarly. For the second claim, we have

$$\begin{aligned} diam(P_A \cup P_B) &\leq diam(A) + d(A, B) + diam(B) \leq (1 + 2\varepsilon)d(A, B) \\ &\leq (1 + 2\varepsilon)\hat{d}(A, B) \leq (1 + 2\varepsilon)^2\beta 2\sqrt{d} \leq \frac{(1 + 2\varepsilon)^2\delta 2\sqrt{d}}{c}. \end{aligned}$$

Substituting $\varepsilon = \frac{1}{6d^2}$ and $c = (2(d + 1))^2$, we get that $diam(P_A \cup P_B) < \frac{\delta\sqrt{2}}{d+1}$. Using Lemma 3.1.16, it follows that $P_A \cup P_B$ lies in $\mathrm{NBR}(\Pi)$. ■

Lemma 4.2.4. *Let* $(A, B) \in W$ *be any WSPD pair. Let* $\{\beta < \delta\}$ *denote the critical scales for* (A, B)*. Consider any arbitrary pair of points* $(a \in P_A, b \in P_B)$*. Let* $\alpha' < \alpha$ *be the pair of consecutive scales such that*

- *at scale* α'*,* a *and* b *lie in distinct non-adjacent permutahedra,*
- *but at scale* α*, they lie in adjacent (or the same) permutahedra.*

Then α *is a critical scale for* (A, B)*, that is, either* $\alpha = \beta$ *or* $\alpha = \delta$*.*

Proof. We prove the claim by contradiction. There are two cases:

- $\alpha < \beta$: From the definition of critical scales, we have that

  $$d(A, B) \geq \frac{\hat{d}(A, B)}{1 + 2\varepsilon} > \alpha(2\sqrt{d}),$$

  that is, the minimum distance between points of P_A and P_B is more than twice the diameter of the cells at scale α. This means that for all pairs of the form $(a \in P_A, b \in P_B)$, the cells containing a and b are not adjacent. This contradicts our assumption that at scale α, there exists a pair of points $(a \in P_A, b \in P_B)$, such that they lie in adjacent (or the same) cells.

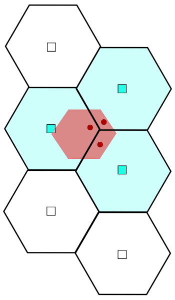


Figure 4.2: The red permutahedron (small) contains three input points, denoted as red (dark) dots. The other permutahedra are at a larger scale. The three points which lie in a single small permutahedron lie in three different blue (shaded) permutahedra, leading to splits.

- $\alpha > \delta$: In such a case, we have $\alpha' \geq \delta$. From Lemma 4.2.3, we know that if $rep(A)$ lies in a cell π at scale δ or higher, then $P_A \cup P_B$ lies in $\mathrm{NBR}(\pi)$. This contradicts our assumption that at scale α', there exists a pair of points $(a \in P_A, b \in P_B)$ which lies in distinct non-adjacent cells.

The claim follows. ■

4.2.1 Size of the tower

Splits

In a cubical tessellation, with a suitable translation and scaling it is easy to ensure that each cell at a given scale is contained entirely in a cell at a larger scale (for instance, in a quad-tree like decomposition). However, in the permutahedral tessellation, each cell at a given scale may not be entirely contained within a single cell at larger scales, irrespective of the choice of translation and scaling. This can lead to cases where the input points contained in a single cell map to several distinct cells at a higher scale. We define such events formally:

Definition 4.2.5 (splits). *At a given scale $\beta > 0$, let π be a non-empty permutahedral cell and denote by $P_\pi \subset P$ the set of input points contained in π. At the next scale β', let*

$$\{\Pi_0, \ldots, \Pi_m\}$$

be the collection of cells to which P_π maps, with π mapping to Π_0.

We call each pair (Π_0, Π_i) for $1 \leq i \leq m$ a split *at scale β'. For each split (Π_0, Π_i), there exists at least one pair of points $\{a, b\} \subset P$ such that*

$$\{a, b\} \in \pi, a \in \Pi_0, b \in \Pi_i.$$

We call (a, b) a split inducing pair *(SIP).*

For an elementary example of splits, see Figure 4.2. Each split is induced by at least one SIP. This also means that several SIPs may induce the same split. In any case, the number of SIPs is an upper bound for the number of splits.

Let W be a $\frac{1}{6d^2}$-WSPD as before and let $(A, B) \in W$ be any well-separated pair. We upper bound the number of splits induced by SIPs of the form (a, b) where $(a \in P_A, b \in P_B)$ over all scales. Counting this for each element of W gives an upper bound on the number of splits for all SIPs, since each pair of points of P is covered by some element of W (see Definition 3.3.7).

Lemma 4.2.6. *Let $(A, B) \in W$ be any well-separated pair. The expected number of splits for SIPs of the form $(a \in P_A, b \in P_B)$ is upper bounded by $2^{O(d)}$.*

Proof. First, we count the expected number of scales at which splits may be induced by pairs of points of the form $(a \in P_A, b \in P_B)$.

We see that at scales below the critical scales for (A, B), points of A and B never lie in the same cell. So, there are no splits induced by such SIPs, and we ignore those scales. There are two relevant cases:

1. *Critical scales:* let $\{\beta < \delta\}$ denote the two critical scales for (A, B). Suppose that there is a split at scale β. Then there exists some SIP $(a \in P_A, b \in P_B)$ which was in a single cell at the scale immediately lower than β, but is in different cells at scale β. From the definition of the critical scales (Definition 4.2.2), we see that a and b must lie in non-adjacent permutahedra at all scales lower than β, so the assumption above is not possible. Therefore, there are no splits at scale β.

   Splits may occur at the next critical scale δ. If $rep(A)$ lies in cell π, then the points of $P_A \cup P_B$ lie in NBR(π), using Lemma 4.2.3. Therefore, an upper bound on the number of full cells occupied by the points of $P_A \cup P_B$ at scale δ is the number of cells in NBR(π), which is $2^{O(d)}$.

2. *Non-critical scales:* we denote these scales by $\mu_i = c^i \delta$, $i \geq 1$. Let π denote the permutahedron at scale μ_i that contains $rep(A)$. Using the proof of Lemma 4.2.3, it holds that

   $$\begin{aligned} diam(P_A \cup P_B) &\leq \frac{(1+2\varepsilon)^2 \delta 2\sqrt{d}}{c} \\ &\leq \frac{(1+2\varepsilon)^2 \mu_i 2\sqrt{d}}{c^{i+1}} < \frac{\mu_i}{d^i}. \end{aligned}$$

   Therefore, points of $P_A \cup P_B$ lie in NBR(π), using Lemma 3.1.16, so that an upper bound on the number of cells occupied by $P_A \cup P_B$ is $2^{O(d)}$. We show that with a high probability, points of $P_A \cup P_B$ lie in π, so that it is unlikely that they occupy many cells. We give an upper bound for the expected number of scales where $P_A \cup P_B$ does not lie in a single cell.

   If $rep(A)$ has distance greater than $diam(P_A \cup P_B)$ from all facets of π, then all points of $P_A \cup P_B$ lie in π. Without loss of generality, assume that π is centered at the origin. Set $x := \mu_i - 3diam(P_A \cup P_B)$ and let π' denote the permutahedron at the origin at scale x. Using Lemma 3.1.21, the Minkowski sum of π' with a ball of radius $\frac{\mu_i - x}{2}\sqrt{\frac{d}{d+1}}$ lies inside π. It follows that

   $$\begin{aligned} \frac{\mu_i - x}{2}\sqrt{\frac{d}{d+1}} &\geq \frac{3diam(P_A \cup P_B)}{2}\sqrt{\frac{d}{d+1}} \\ &> diam(P_A \cup P_B). \end{aligned}$$

   Because of the random translation of the lattice at each scale, the location of $rep(A)$ inside π is uniformly distributed. Let Q_i be the probability that $rep(A)$ lies in π', and $Q_i' = 1 - Q_i$ its complement. As before, $diam(P_A \cup P_B) < \frac{\mu_i}{d^i}$, so $\frac{diam(P_A \cup P_B)}{\mu_i} < \frac{1}{d^i}$.

Using this fact, we see that

$$\begin{aligned} Q_i = \frac{Vol(\pi')}{Vol(\pi)} &= \Big(\frac{x}{\mu_i}\Big)^d \\ &= \Big(1 - \frac{3diam(P_A \cup P_B)}{\mu_i}\Big)^d \implies Q_i > \Big(1 - \frac{3}{d^i}\Big)^d. \end{aligned}$$

Using Bernoulli's inequality [Ber], $Q_i > 1 - \frac{3d}{d^i}$, so $Q_i' < \frac{3}{d^{i-1}}$. Let T_i denote the probability that at scale μ_i, $P_A \cup P_B$ lies in π, with $T_i' = 1 - T_i$ denoting the complement. Since $T_i \geq Q_i$, we have that $T_i' \leq Q_i' < \frac{3}{d^{i-1}}$. The expected number of scales where $P_A \cup P_B$ does not lie in π, implying that splits can occur, is upper bounded by

$$\sum_{i=1}^{\infty} T_i' < \sum_{i=1}^{\infty} \frac{3}{d^{i-1}} < 6.$$

The total number of scales where splits can occur for (A, B) is seven in expectation, one being the critical scale δ and six being non-critical scales. At each such scale, points of $P_A \cup P_B$ lie in $2^{O(d)}$ cells, so a simple upper bound for the number of splits is the number of combinations of two cells from the full cells, which is $\binom{2^{O(d)}}{2}$. This is again $2^{O(d)}$, so the claim follows. ■

We now bound the size of our approximation tower.

Lemma 4.2.7. *The expected number of vertex inclusions in the tower is upper bounded by* $n2^{O(d \log d)}$.

Proof. At scale β_0, there are n vertex inclusions in the tower due to n full permutahedra. To prove the claim, we first show that each vertex inclusion at higher scales is caused by a split.

Let $\alpha' < \alpha$ be any two consecutive scales in the tower, with the set of full vertices being V' and V, respectively and let θ be the simplicial map from the complex at α' to the complex at α. Let $v \in V \setminus \theta(V')$ denote a vertex inclusion. There is an input point $p \in \Pi(v)$, which is the full cell corresponding to v. Let $\Pi(u)$ denote the full cell at scale α', which contains p. Since $\theta(u) \neq v$, there exists another input point $p' \in \Pi(u)$ at scale α' such that p' is the closest input point to u. Then $(\theta(u), v)$ is a split induced by the SIP (p, p'), implying that v was created from a split.

There are at most $n(6d^2)^{O(d)} = n2^{O(d \log d)}$ pairs in the WSPD, so the total number of expected splits is upper bounded by $n2^{O(d \log d)} \cdot 2^{O(d)}$, using Lemma 4.2.6. ■

Theorem 4.2.8. *The expected size of the tower is upper bounded by* $n2^{O(d \log d)}$.

Proof. By definition, the size of a tower is the number of simplex inclusions involved. From Lemma 4.2.7, we know that the expected number of vertex inclusions in the tower is upper bounded by $n2^{O(d \log d)}$. Each simplex included in the tower is attached to one of these vertices. From Theorem 4.2.1 we know that each vertex has at most $2^{O(d \log k)}$ k-simplices attached to it. Therefore, using a simple charging argument, the expected number of simplex inclusions is upper bounded by $n2^{O(d \log d)} \left(\sum_{i=1}^{d} 2^{O(d \log i)}\right) = n2^{O(d \log d)}$. ■

Note that we do not explicitly construct the WSPD W to argue about the size of the tower. The existence of W suffices to prove our claims.

For the computation of the approximate tower, we need another result, whose proof is similar to that of Lemma 4.2.6. We show that under the images of the simplicial maps, each simplex inclusion collapses to a vertex very soon, within the next few scales:

Lemma 4.2.9. *Let σ be any k-simplex ($k \geq 1$), and let δ_1 denote the scale at which it is included in the tower. Let $\delta_{i+1} = c^i\delta_1$, $i \geq 1$ denote the next scales. Let the simplicial complexes and simplicial maps at these scales be*

$$X_{\delta_1} \xrightarrow{\theta_1} X_{\delta_2} \xrightarrow{\theta_2} \ldots .$$

Let $\theta^i = \theta_i \circ \theta_{i-1} \circ \ldots \circ \theta_1$ denote the i-fold composition of simplicial maps over i consecutive scales. Then,

- *$\theta^1(\sigma)$ is a vertex with probability greater than $1/2$.*
- *Let j denote the smallest integer such that $\theta^j(\sigma)$ is a vertex. We say that σ* survives *for j scales. Then, the expected value of j is at most four.*

Proof. Let σ be a simplex with vertices as full cells $(\pi_0, \ldots, \pi_k)$, included in the tower at scale δ_1. The diameter of this collection of cells is no more than $2\delta_1\sqrt{d}$, from Lemma 3.1.10. Let s be any point of $\mathbb{R}^d$ in π_0, and denote by Π the permutahedron at scale δ_2 that contains s. Without loss of generality, assume that Π is centered at the origin.

Let Pr_i denote the probability that $\theta^i(\sigma)$ is a vertex given that $\theta^{i-1}(\sigma)$ was not. Then, Pr_1 is the probability that $\theta^1(\sigma) = \Pi$. If s lies at distance at least $2\delta_1\sqrt{d}$ from the facets of Π, then $\{\pi_0 \cup \ldots \cup \pi_k\}$ lies inside Π, guaranteeing that $\theta^1(\sigma) = \Pi$. Set $x := \delta_2 - 6\delta_1\sqrt{d}$ and denote by Π' the permutahedron centered at the origin at scale x. From Lemma 3.1.21, the Minkowski sum of Π' with a ball of radius $\frac{\delta_2 - x}{2}\sqrt{\frac{d}{d+1}}$ lies inside Π. We see that $\frac{\delta_2 - x}{2}\sqrt{\frac{d}{d+1}} > 2\delta_1\sqrt{d}$, so if s lies in Π', then it is further than $2\delta_1\sqrt{d}$ from the facets of Π. Since the origin is randomly translated at each scale, the position of s is uniformly distributed in Π. Let Qr denote the probability that s lies in Π'. Then,

$$\begin{aligned} Qr = \frac{Vol(\Pi')}{Vol(\Pi)} &= \left(\frac{x}{\delta_2}\right)^d \\ &= \left(1 - \frac{6\delta_1\sqrt{d}}{\delta_2}\right)^d = \left(1 - \frac{6\sqrt{d}}{c}\right)^d \\ &= \left(1 - \frac{3\sqrt{d}}{2(d+1)^2}\right)^d . \end{aligned}$$

Using Bernoulli's inequality [Ber], $Qr > 1 - \frac{3d\sqrt{d}}{2(d+1)^2} > 1/2$. Since $Pr_1 \geq Qr > 1/2$, the first claim follows.

If $\theta^i(\sigma)$ is a vertex, it remains so for all higher scales. Because the origin is chosen uniformly at random at each scale, and the ratio of any two consecutive scales is a constant, we have that $Pr_i \geq Qr$, for all i.

Let Pr'_i denote the complement of Pr_i. Then, the probability that σ survives for j scales is $(Pr'_1 \ldots Pr'_{j-1})Pr_j$. Since $Pr_i > 1/2$, we have $(Pr'_1 \ldots Pr'_i) < 1/2^i$. The expected number of scales for which σ survives is

$$\sum_{j=1}^{\infty} j(Pr'_1 \ldots Pr'_{j-1})Pr_j < \sum_{j=1}^{\infty} j(Pr'_1 \ldots Pr'_{j-1}) < \sum_{j=1}^{\infty} j/2^{j-1} = 4.$$

The claim follows. ■

4.2.2 Computing the tower

We now discuss two algorithms to compute the tower. As a precursor, we first need to determine the range of relevant scales.

Determining the range of scales If the range of scales $[\beta_0, \beta_m]$ is provided as an input, we construct the tower by building the approximation complexes at each of the relevant scales. If the range is not provided, we calculate $diam(P)$ and $CP(P)$ to determine the relevant scales. For our purpose, it suffices to calculate constant-factor approximations of these quantities. Taking an arbitrary point $p \in P$ and calculating $max_{q \in P} \|p - q\|$ gives a 1/2-approximation of $diam(P)$. $CP(P)$ can be computed exactly using a randomized algorithm in $n2^{O(d)}$ expected time [KM95]. Using this information, we calculate the range of scales $[\beta_0, \beta_m]$.

Algorithm A We construct the tower scale-by-scale, inductively. At the lowest scale β_0, we locate the nearest lattice points for points of P using the algorithm in Sub-section 3.1.3. The complex consists of n vertices at this scale.

Let $\alpha' < \alpha$ be any two consecutive scales and X', X the respective complexes, with $\theta : X' \to X$ being the induced simplicial map. Suppose we have already constructed X'. There are two steps in constructing the complex X:

- *Adding vertices and edges to X:* we translate the lattice by picking a point uniformly at random from the cell at the origin, which can be done using random walks in polytopes [LS93]. We compute the set of full permutahedra by finding the closest lattice point for each point in P, using Algorithm 2 (Chapter 4). Then, for each full cell π, we go over $\text{NBR}(\pi) \setminus \pi$ to find neighboring full cells; whenever a full neighbor is found, we add an edge between π and its neighbor. This completes the 1-skeleton of X.
- *Adding simplices to X:* each simplex in X is one of two kinds:
    - those which are in the image θ: to add these, we first construct θ for vertices of X'. To do this, we simply compute the nearest input point for each full cell of X', and then compose it with the pre-computed map from P to vertices of X.

      After this, we go over each simplex $\sigma = (v_0, \ldots, v_k) \in X'$ and add the simplex $\theta(\sigma)$ on the vertices $(\theta(v_0), \ldots, \theta(v_k))$ to X.
    - those which are not in the image of θ, that is, the simplices which are included in the tower at scale α. Each such simplex σ must contain at least one edge which is not in the image of θ, since otherwise all edges of σ and hence σ itself would be in the image of θ, since X is a flag complex (Lemma 3.1.19).

      We first enumerate all the edges which are not in the image of θ. To do this, for each edge $(u, v) \in X'$, we exclude $(\theta(u), \theta(v))$ from the list of edges of X, to get the list of new edges.

      We construct the k-skeleton from the 1-skeleton, by going over the new edges of the complex in an arbitrary order, and at each step we add the new simplices induced by the current edge. Let $e = (u, v)$ be the edge under consideration. We construct the simplices incident to e inductively by dimension.

      The base case is the 1-skeleton, with simplex e. Assume that we have completed the $(j-1)$-simplices incident to e. Let σ be a j-simplex incident to e. Then, σ is of the form $\sigma = (w, \gamma)$, where γ is a $(j-1)$-simplex incident to e and w is a full cell which is a common neighbor of u and v. To find σ, we go over each of the $2^{O(d)}$ common neighbors of u and v and each $(j-1)$-simplex γ containing e, and test whether (w, γ) is a j-simplex in the complex. The test works by checking whether each (w, γ_i) is a $(j-1)$-simplex in the complex, where γ_i is a facet of γ. Since we enumerate all the simplices attached to each new edge, this step generates all simplices included in the tower at scale α. It is easy to see that the order in which the edges are processed is irrelevant.

Theorem 4.2.10. *Algorithm A takes*

$$n2^{O(d)} \log \Delta + M2^{O(d)}$$

time in expectation, and M space to compute the k-skeleton, where M is the size of the tower. Further, the expected runtime is upper bounded by

$$n2^{O(d)} \log \Delta + n2^{O(d \log d)}$$

and the expected space is upper bounded by $n2^{O(d \log d)}$.

Proof. At each scale, picking the origin takes $poly(d)$ time [LS93]. Finding the closest lattice vertex for any given input point takes $O(d^2)$ time (see Sub-section3.1.3). Therefore, finding the full vertices at each scale takes $O(nd^2)$ time per scale, and in total $O(nd^2 \log \Delta)$ time. Each cell has $2^{O(d)}$ neighbors, so finding the full neighbors and adding the edges takes $n2^{O(d)}$ per scale. Computing the map θ for the vertices of X' takes time $O(nd)$ per scale. In total, these steps take $n2^{O(d)} \log \Delta$ time.

For each simplex of X', we compute the image under θ. This takes time $O(d)$ per simplex of X', since the vertex map has already been established. From Lemma 4.2.9, each simplex in the tower survives at most four scales (in expectation), until it collapses to a vertex. Therefore, for each simplex in the tower, we compute its related images four times in expectation. This step takes $4MO(d)$ time over the tower, in expectation.

Computing θ for the edges of X' takes time $O(1)$ time per edge, since we already computed the vertex map. Finding new edges takes $n2^{O(d)}$ time, since that is the maximum number of edges at any scale. In total, finding new edges takes $n2^{O(d)} \log \Delta$ time. To complete the k-skeleton, the testing technique requires an overhead of $k^2 2^{O(d)} = 2^{O(d)}$ for each simplex in the tower. Since we do the k-completion only for newly added edges, the test is not repeated for any simplex. The time bound follows.

The space complexity follows by storing the tower. The expected size of the tower is upper bounded by $n2^{O(d \log d)}$, from Theorem 4.2.8. The claims follow. ■

In Algorithm A, we scan the neighborhood of each full cell to construct the edges of the complex at each scale. By adding the edges in more careful method, we reduce the complexity of this step. Let W denote a $\frac{1}{6d^2}$-WSPD on P. Let $\alpha' < \alpha$ be any two consecutive scales of the tower, with X', X being the complexes at the respective scales. Let $\theta : X' \to X$ be the induced simplicial map. For any permutahedron π, let $\mathrm{NBR}(\mathrm{NBR}(\pi))$ denote the union of the collections of cells $\mathrm{NBR}(\pi_i)$, for each cell $\pi_i \in \mathrm{NBR}(\pi)$.

Definition 4.2.11 (types of edges). *Let $\pi_1 \neq \pi_2$ be any pair of distinct full cells at scale α such that (π_1, π_2) is an edge in X. There are three possibilities:*

- *There exist adjacent full cells $\{u, v\} \in X'$ such that $\theta(u, v) = (\pi_1, \pi_2)$, that is, (π_1, π_2) is the image of an edge from the previous scale. In such a case, we call (π_1, π_2) an* inherited edge.

- *There exist full cells $\{u, v\} \in X'$ such that $\theta(u) = \pi_1$ and $\theta(v) = \pi_2$, but (u, v) is not an edge in X'. We call (π_1, π_2) an* interactive edge.
- *At least one of $\{\pi_1, \pi_2\}$ have no pre-image in X' under θ, that is, there do not exist cells $\{u, v\} \in X'$, such that $\theta(u) = \pi_1$ and $\theta(v) = \pi_2$ both hold. In such a case we call (π_1, π_2) a* split edge.

Since the three edge classes are exhaustive, each edge of X is either an inherited edge, or an interactive edge, or a split edge.

Lemma 4.2.12. *Let (π_1, π_2) be an interactive edge of X. Then,*

- *There exists a pair $(A, B) \in W$ such that α is a critical scale for (A, B).*
- *Let π_3 be the permutahedron containing rep(A) at scale α. Then, π_1 and π_2 are cells in* $\mathrm{NBR}(\mathrm{NBR}(\pi_3))$.

Proof. For the first claim, let $\{u, v\}$ be distinct non-adjacent full cells at scale α' such that $\theta(u) = \pi_1$ and $\theta(v) = \pi_2$. Since u and v are full cells, there exist points $\{p_1, p_2\} \in P$ such that $p_1 \in u, p_2 \in v$, and p_1 and p_2 are the closest points to centers of u and v, respectively. At scale α, $p_1 \in \pi_1$ and $p_2 \in \pi_2$, by the definition of θ. Let $(A, B) \in W$ be a WSPD pair which covers (p_1, p_2), that is, $\{p_1 \in P_A, p_2 \in P_B\}$. Using Lemma 4.2.4 (setting $a := p_1, b := p_2$ in the lemma), it immediately follows that α is a critical scale for (A, B).

For the second claim, using Lemma 4.2.3, points of P_A lie in $\mathrm{NBR}(\pi_3)$, so $\pi_1 \in \mathrm{NBR}(\pi_3)$. Since $\pi_2 \in \mathrm{NBR}(\pi_1)$, the claim follows. ■

Algorithm B There are two stages in the algorithm.

Stage 1 We compute a $\frac{1}{6d^2}$-WSPD W on P. For each WSPD pair $(A, B) \in W$, the two critical scales are determined using $\hat{d}(A, B)$ (see Definition 4.2.2). For each critical scale in the tower, we store the WSPD pairs for which the scale is critical.

Stage 2 We construct the complex scale-by-scale. For this, let $\alpha' < \alpha$ be any two consecutive scales. Suppose we have constructed the complex X' at α'. We choose the origin at α as in Algorithm A. To construct the complex X at α, we start by finding the full vertices by mapping points of P to their closest lattice point. Then we calculate the vertex map from X' to X which induces the simplicial map $\theta : X' \to X$.

The simplices in X are of two kinds: those which are images of θ and those which are not. For simplices of the former kind, we use the vertex map to compute the image under θ, and add it to X. For the latter case, each simplex must contain a new edge, since otherwise the simplex would already be in the image of θ. To compute these new edges at α, we use Lemma 4.2.12: the only new edges at this scale are the interactive and split edges.

Step 1 We process all WSPD pairs which are critical at scale α, one by one. Let $(A, B) \in W$ be the current pair and let π denote the permutahedron which contains *rep*(A). For each cell $\pi' \in \mathrm{NBR}(\pi)$, we add edges of π' with full cells of $\mathrm{NBR}(\pi') \setminus \pi'$. This amounts to adding edges between all pairs of adjacent full cells in $\mathrm{NBR}(\mathrm{NBR}(\pi))$. By Lemma 4.2.12, all interactive edges are added by this procedure.

Step 2 We collect the full cells which do not have a pre-image under θ. This is done by excluding the images of the vertices of X' under θ, from the set of vertices of X. For each such full cell π, we go over $\mathrm{NBR}(\pi) \setminus \pi$ and add edges with full cells. This step enumerates all split edges.

Step 3 Steps 1 and 2 generate the new edges of X. Together with the image of the 1-skeleton of X', this completes the 1-skeleton of X. With this information, we enumerate the k-skeleton of X, using the technique from Algorithm A.

Theorem 4.2.13. *Algorithm B takes*

$$(O(nd^2) + poly(d)) \log \Delta + n \log n 2^{O(d)} + (M + |W|)2^{O(d)}$$

time in expectation and $M + O(|W|)$ *space, where* M *is the size of the tower and* $|W|$ *is the size of the WSPD. Further, the expected runtime is upper bounded by*

$$(O(nd^2) + poly(d)) \log \Delta + n \log n 2^{O(d)} + n 2^{O(d \log d)}$$

and the expected space is upper bounded by $n2^{O(d \log d)}$.

Proof. In Stage 1, we compute a $\frac{1}{6d^2}$-WSPD, which takes time $n \log n 2^{O(d)} + |W|$. For each WSPD pair we calculate two critical scales. This takes $O(1)$ time per pair, so $O(|W|)$ in total. Stage 1, therefore, takes $n \log n 2^{O(d)} + O(|W|)$ time.

In Stage 2, at each scale, we select the origin as in Theorem 4.2.10, which takes $poly(d)$ time per scale [LS93]. Then, we compute the full vertices at each scale. This takes time $O(nd^2)$ per scale. Computing the vertex map which induces θ also takes $O(nd^2)$ per scale. In total, these steps take $(O(nd^2) + poly(d)) \log \Delta$ time. Taking the image of simplices of X' takes $O(d)$ time per simplex, as the vertex map is already computed. As argued in Theorem 4.2.10, this step takes $4MO(d)$ time in expectation. For the remaining simplices of X,

- In Step 1, we add edges between adjacent full cells of $\mathrm{NBR}(\mathrm{NBR}(\pi))$. There are $2^{O(d)}$ such cells, so it takes $2^{O(d)}$ time per WSPD pair per critical scale. Since there are $2|W|$ such instances, in total this step takes $2^{O(d)}|W|$ time.

- In Step 2, we inspect the neighbors of full cells which do not have a pre-image under θ. The number of such full cells is the number of vertex inclusions in the tower, which is upper bounded by M. Per cell, this takes $2^{O(d)}$ time, so this step takes no more than $2^{O(d)}M$ time in total.

- In Step 3, the new edges are the inherited and split edges. Each such edge survives four scales in expectation, from Lemma 4.2.9, so the expected number of new edges in the tower is upper bounded by $4M$. This is also the time required to find the new edges. Completing the k-skeleton has an overhead of $k^2 2^{O(d)}$ per simplex in the tower as in Algorithm A, so it takes $M2^{O(d)}$ time in total.

In total, Stage 2 takes time $(O(nd^2) + poly(d)) \log \Delta + (M + |W|)2^{O(d)}$. The time bound follows.

Storing the critical scales for each WSPD pair takes $O(1)$ space per pair. Additionally, we store the tower. The space bound follows.

In the worst case, $|W| = n(6d^2)^{O(d)} = n2^{O(d \log d)}$ and M is upper-bounded by $n2^{O(d \log d)}$ in expectation. The claim follows. ■

It is possible to compute the persistence barcode of towers in a streaming setting [KS17], where instead of storing the entire tower in memory, the complex is constructed at each scale and fed to the output stream. In this setting, the memory consumption of Algorithm B is $O(|W|) + M_i$, where M_i is the size of the complex at any scale. Since $|W|$ can be as large as $n2^{O(d \log d)}$ and M_i can be at most $n2^{O(d \log k)}$(Theorem 4.2.1), the space requirement is at most $n2^{O(d \log d)}$. The same bound also holds for Algorithm A, although it does not need to compute and store the WSPD.

If the spread is a constant, then Algorithm A has better a runtime, since it does not compute the WSPD. Also, Algorithm A does not have to store the critical scales of the WSPD, neither in the normal setting nor in the streaming environment, so it is more space-efficient. However, if the spread is large, then Algorithm B achieves a better runtime, since it avoids the $n2^{O(d)} \log \Delta$ factor in the complexity of Algorithm A.

4.2.3 Dimension reduction

When the ambient dimension d is large, our approximation scheme plays nicely together with dimension reduction techniques, which were discussed in Section 3.5. We show that we can shrink the expected approximation size from Theorem 4.2.8 for the case $d \gg \log n$, only worsening the approximation quality by a constant factor.

Theorem 4.2.14. *Let P be a set of n points in $\mathbb{R}^d$.*

- *There exists a constant c and a tower of the form $\left(\bar{X}_{(c \log n)^{2k}}\right)_{k \in \mathbb{Z}}$ which gives rise to a $(3c \log n)$-approximation of the Rips persistence module of P.*
- *The approximation tower has only $n^{O(\log \log n)}$ simplices in expectation.*
- *With high success probability, we can compute the tower in deterministic expected running time*
$$n(\log n)^2 O(\log \Delta) + n^{O(\log \log n)}$$
using Algorithm B.

Proof. The Johnson-Lindenstrauss Lemma [JLS86] (also Theorem 3.5.2) asserts the existence of a dimension-reduction map for Rips filtrations, f as in Lemma 3.5.5. The map f has the parameter $m = \lambda \log n / \varepsilon^2$ with some absolute constant λ and $\xi_1 = (1 - \varepsilon)$, $\xi_2 = (1 + \varepsilon)$. Choosing $\varepsilon = 1/2$, we obtain that $m = O(\log n)$ and $\xi_2 / \xi_1 = 3$.

Let $\overline{\mathcal{R}}_\alpha$ denote the Rips complex after the Johnson-Lindenstrauss transform on P. From Lemma 3.5.5, we have that $(H(\overline{\mathcal{R}}_\alpha))_{\alpha \geq 0}$ is a 3-approximation of $(H(\mathcal{R}_\alpha))_{\alpha \geq 0}$. Moreover, using the approximation scheme from Section 4.1, we can define a tower $(X_\beta)_{\beta \geq 0}$ on the Johnson-Lindenstrauss transform of P, whose induced persistence module is a $6(m+1)$-approximation of $(H(\overline{\mathcal{R}}_\alpha))_{\alpha \geq 0}$. Substituting $m = O(\log n)$ and using the property of transitivity of persistence modules from Lemma 2.3.7, the first claim follows.

The expected size of the approximation tower is upper bounded by $n2^{O(m \log m)} = n^{O(\log \log n)}$, from Theorem 4.2.8. The second claim follows.

The Johnson-Lindenstrauss lemma further implies that an orthogonal projection to a randomly chosen subspace of $\mathbb{R}^d$ of dimension m will yield a map f as above, with high probability. Our algorithm picks such a subspace, projects all points into it (this requires $O(dn \log n)$ time) and then applies the approximation scheme for the projected point set. The runtime bound follows from Theorem 4.2.13, by substituting the value of m. ■

The approximation complex from the previous theorem has size $n^{O(\log\log n)}$, which is super-polynomial in n. Using a slightly more elaborate dimension reduction result by Matoušek [Mat90] (Theorem 3.5.3), we can get a size bound polynomial in n, at the price of an additional $O(\log n)$-factor in the approximation quality. First, we note that by setting $k := \frac{4\log n}{\log\log n}$ in Matoušek's result in Theorem 3.5.3, we get an embedding of n points of $\mathbb{R}^d$ into k dimensions with a distortion of at most $O(\sqrt{\log n \log\log n})$.

Theorem 4.2.15. *Let P be a set of n points in $\mathbb{R}^d$.*

- *There exists a constant c and a discrete tower of the form*

  $$\left(\bar{X}_{\left(c\frac{\log n}{\log\log n}\right)^{2k}}\right)_{k\in\mathbb{Z}},$$

  which gives rise to a $\left(3c\log n\left(\frac{\log n}{\log\log n}\right)^{1/2}\right)$-approximation of the Rips persistence module on P.

- *The expected size of the approximation tower is upper bounded by $n^{O(1)}$.*

- *Moreover, we can compute, with high success probability, the approximation tower in deterministic expected running time*

  $$n(\log n)^2 O(\log\Delta) + n^{O(1)}$$

  using Algorithm B.

Proof. The proof follows the same pattern of Theorem 4.2.14 with a few changes. We use Matoušek's dimension reduction result described in Theorem 3.5.3 with the projection dimension being $m := \frac{4\log n}{\log\log n}$. Hence, $\xi_2/\xi_1 = O(\sqrt{\log n\log\log n})$ for the Rips construction. The final approximation factor is $6(m+1)\xi_2/\xi_1$ which simplifies to $O(\log n(\frac{\log n}{\log\log n})^{1/2})$. The size and runtime bounds follow by substituting the value of m in the respective bounds. ■

Finally, we consider the important generalization that P is not given as an embedding in $\mathbb{R}^d$, but as a point sample from a general metric space. Recall the classical result by Bourgain [Bou85] (also Theorem 3.5.6) to embed P in Euclidean space with small distortion. Bourgain's result permits an embedding into $m = O(\log^2 n)$ dimensions with a distortion $\xi_2/\xi_1 = O(\log n)$, where the constants are independent of n and d.

As discussed in Section 3.5, we can use this embedding in a pipeline to approximate Rips filtrations. The results are similar to Theorems 4.2.14 and 4.2.15, except that the approximation quality further worsens by a factor of $O(\log n)$. Note that we could have used Theorem 4.2.14 as the dimension reduction step, but that does not lead to a polynomial complexity in n. Therefore, we only state the generalized version of Theorem 4.2.15. The proof is straight-forward with the same techniques as before.

Theorem 4.2.16. *Let P be a general metric space with n points.*

- *There exists a constant c and a discrete tower of the form*

  $$\left(\bar{X}_{\left(c\frac{\log n}{\log\log n}\right)^{2k}}\right)_{k\in\mathbb{Z}},$$

  which gives a persistence module that $\left(3c\log^2 n\left(\frac{\log n}{\log\log n}\right)^{1/2}\right)$-approximates the Rips persistence module on P.

- *The expected size of the approximation tower is upper bounded by* $n^{O(1)}$.
- *Moreover, we can compute, with high success probability the tower with this property in deterministic expected running time*

  $$n(\log n)^2 O(\log \Delta) + n^{O(1)}$$

  using Algorithm B.

Remark 4.2.17. *The Čech filtration is a* $\sqrt[4]{2}$*-approximation of the Rips filtration, from Lemma 2.3.9. While the approximation results in Section 4.1 and Sub-section 4.2.3 have been presented for the Rips filtration, they also apply to Čech filtrations on the same space with a multiplicative factor of* $\sqrt[4]{2}$ *in the approximation quality. This is true because of the transitivity of interleavings from Lemma 2.3.7.*

4.3 A lower bound for approximation schemes

Recall that the Čech filtration is associated with a barcode, which represents persistent features (Definition 2.2.12). We construct a point configuration P in Euclidean space such that its Čech filtration gives rise to a large number (say N) of features with "large" persistence, relative to the scales on which the features appear. From the definition of bottleneck matching (Definition 2.3.1), it is easy to see that any ε-approximation of such a Čech filtration has to contain at least one interval in its persistent barcode per such persistent feature of the Čech filtration, if ε is sufficiently small. As a result, any such approximation will yield a barcode of size at least N.

If the approximation stems from a simplicial tower, then the appearance of any interval in the barcode requires that a new simplex be added to the tower. Therefore, N is a lower bound on the number of simplices in the approximation. Also more generally, because of the interval decomposition of persistence modules (see Definition 2.2.10), it makes sense to assume that any representation of a persistence module is at least as large as the size of the resulting persistence barcode. We formalize the intuition of large persistent features and the corresponding lower bounds:

Definition 4.3.1. *For a point set P, we call an interval $[\alpha, \alpha']$ of of the Čech persistence module $(H(\mathcal{C}_\alpha(P)))_{\alpha \geq 0}$* δ-significant *for $0 < \delta < \frac{\alpha' - \alpha}{2\alpha'}$.*

Lemma 4.3.2. *For $0 < \delta < 1/2$, let N denote the number of δ-significant intervals of $(H(\mathcal{C}_\alpha))_{\alpha \geq 0}$. Then, any persistence module $(V_\alpha)_{\alpha \geq 0}$ that is an $(1+\delta)$-approximation of $(H(\mathcal{C}_\alpha))_{\alpha \geq 0}$ has at least N intervals in its barcode.*

Proof. First, we claim that

- If $[\alpha, \alpha']$ is δ-significant, then there exists some $\varepsilon > 0$ and $c \in (\alpha, \alpha')$ such that

  $$\frac{\alpha}{(1-\varepsilon)} \leq \frac{c}{(1+\delta)} < c(1+\delta) \leq \alpha'. \tag{4.3}$$

- Any persistence module that is an $(1+\delta)$-approximation of $(H(\mathcal{C}_\alpha))_{\alpha \geq 0}$ needs to represent an approximation of the interval $[\alpha, \alpha']$ in the range $(c(1-\varepsilon), c)$. In other words, there is an interval in the approximation corresponding to $[\alpha, \alpha']$.

We first argue that δ-significance implies the existence of $\varepsilon > 0$ and $c \in [\alpha, \alpha']$ such that Equation (4.3) holds. We choose $c := \alpha'/(1+\delta)$, so that the last inequality is satisfied. For the first inequality, we first note that $(1-2\delta) < \frac{1}{(1+\delta)^2}$ for all $\delta < 1/2$. By assumption, $\alpha' - \alpha > 2\alpha'\delta$, so $\alpha < \alpha'(1-2\delta) < \frac{\alpha'}{(1+\delta)^2} = \frac{c}{1+\delta}$. Since the inequality is strict, we can choose some small $\varepsilon > 0$, such that $\alpha/(1-\varepsilon) \leq \frac{c}{1+\delta}$.

By the definition of $(1+\delta)$-approximation using strong interleaving, we have a commutative diagram

$$\begin{tikzcd}
H(\mathcal{C}_{\frac{c(1-\varepsilon)}{(1+\delta)}}) \arrow[rrr, "inc"] \arrow[dr, "\phi"] & & & H(\mathcal{C}_{c(1+\delta)}) \\
& V_{c(1-\varepsilon)} \arrow[r, "h"] & V_c \arrow[ur, "\psi"] &
\end{tikzcd} \tag{4.4}$$

where *inc* is the linear map corresponding to the inclusion map, h is the linear map connecting the approximation module, and ϕ, ψ are the maps connecting the two modules. Let γ be the element in $H(\mathcal{C}_{\frac{c(1-\varepsilon)}{(1+\delta)}})$, corresponding to the δ-significant interval $[\alpha, \alpha']$. By definition, $inc(\gamma) \neq 0$. It follows that $h(\phi(\gamma)) \neq 0$ either, since the diagram needs to commute; so there is a interval corresponding to γ in the approximation. ■

Setup We next define our point set for a fixed dimension d. Consider the A_d^* lattice with origin $\mathbb{O}$. $\mathbb{O}$ has $2^{d+1} - 2$ neighbors in the Delaunay triangulation $\mathcal{D}$ of A_d^*, because its dual Voronoi polytope, the permutahedron Π, has that many facets. We define $P := \mathcal{NBR}(\mathbb{O})$, that is, the union of $\mathbb{O}$ with all its Delaunay neighbors, yielding a point set of cardinality $2^{d+1} - 1$. As usual, we set $n := |P|$, so that $d = \Theta(\log n)$.

We write $\mathcal{D}_P$ for the Delaunay triangulation of P. Since P contains $\mathbb{O}$ and all its neighbors, the Delaunay simplices of $\mathcal{D}_P$ incident to $\mathbb{O}$ are the same as the Delaunay simplices of $\mathcal{D}$ incident to $\mathbb{O}$. Thus, according to Proposition 3.1.14, a $(k-1)$-simplex of $\mathcal{D}_P$ incident to $\mathbb{O}$ corresponds to a $(d-k+1)$-face of Π, and thus to an ordered k-partition of $[d+1]$.

Fix a integer parameter $\ell \geq 3$, which we define later.

Definition 4.3.3 (good partition). *We call an ordered k-partition $(S_1, \ldots, S_k)$* good, *if $|S_i| \geq \ell$ for every $i = 1, \ldots, k$. We define good Delaunay simplices and good permutahedron faces accordingly using Proposition 3.1.14.*

Our proof has two main ingredients. First, in Sub-section 4.3.1 we show that a good Delaunay simplex either gives birth to or kills an interval in the Čech persistence module that has a lifetime of at least $\frac{\ell}{8(d+1)^2}$. This justifies our notion of "good", since good k-simplices create features that have to be preserved by a sufficiently precise approximation. Secondly, in Sub-section 4.3.2 we show that there are $2^{\Omega(d \log \ell)}$ good k-partitions, so good faces are abundant in the permutahedron.

4.3.1 Persistence of good simplices

Let us consider our first statement. Recall from Sub-section 2.4.2 that for any simplex σ, $\mathrm{rad}(\sigma)$ is the radius value of σ in the Čech filtration, which means that σ is included in the tower at scale $\mathrm{rad}(\sigma)$. For our arguments, it will be convenient to have an upper bound for $\mathrm{rad}(\sigma)$. Clearly, such a value is given using the diameter of P. It is not hard to see the following bound (compare Lemma 3.1.10), which we state for reference:

Lemma 4.3.4. *The diameter of P is at most $2\sqrt{d}$, which implies that $\operatorname{rad}(\sigma) \leq 2\sqrt{d}$ for each simplex σ of the Čech filtration.*

From Sub-section 2.4.2, we know that a Čech filtration can be transformed into a simplex-wise tower. In such a case, for a Čech filtration, it makes sense to talk about the persistence of an interval associated to a simplex. Fix any $(k-1)$-simplex σ of $\mathcal{D}_P$ incident to $\mathbb{O}$. σ lies in the Čech filtration for some scale.

Lemma 4.3.5. *Let f be the $(d-k)$-face of Π dual to σ, and let $bc(f)$ denote its barycenter. Then, $\operatorname{rad}(\sigma)$ is the distance $\|bc(f) - \mathbb{O}\|$.*

Proof. $bc(f)$ is the closest point to $\mathbb{O}$ on f because the vector $bc(f) - \mathbb{O}$ is orthogonal to the vector $bc(f) - v$ for any boundary vertex v of f. Since f is dual to σ, all vertices of σ are in same distance to $bc(f)$. ■

Recall from Sub-section 2.4.2 that D_σ and D^*_σ denote the difference of the radius values of σ and its (co-)facets, that is,

$$D_\sigma := \min_{\tau \text{ is facet of } \sigma} \operatorname{rad}(\sigma) - \operatorname{rad}(\tau), \quad \text{and} \quad D^*_\sigma := \min_{\tau \text{ is co-facet of } \sigma} \operatorname{rad}(\tau) - \operatorname{rad}(\sigma).$$

Theorem 4.3.6. *For each good simplex $\sigma \in \mathcal{D}_P$, both D_σ and D^*_σ are at least $\frac{\ell}{24(d+1)^{3/2}}$.*

Proof. We start with D^*_σ. Let $\sigma \in \mathcal{D}_P$ be a $(k-1)$-simplex and let τ be a co-facet of σ.

First, we will bound the quantity $\operatorname{rad}(\tau)^2 - \operatorname{rad}(\sigma)^2$. Let e be the face of Π dual to τ and let $bc(e)$ denote the barycenter of e. Let f be the face of Π dual to σ. By Lemma 4.3.5, the radius values of τ and σ are the squared norms of the barycenters $bc(e)$ and $bc(f)$, respectively. It is possible to derive an explicit expression of the coordinates of $bc(f)$ and $bc(e)$.

Let $S_1, \ldots, S_k$ be the ordered partition of $[d+1]$ corresponding to σ. We obtain the partition corresponding to τ by splitting some set S_i in the corresponding partition $(S_1, \ldots, S_k)$ of σ, into two non-empty parts, from Proposition 3.1.14. Assume without loss of generality that

- S_k is split into S'_k and S'_{k+1}, that is, $S_k = (S'_k, S'_{k+1})$ (splitting any other S_i yields the same bound),
- and that S_k is of size exactly ℓ (a larger cardinality only leads to a larger difference in the quantity $\operatorname{rad}(\tau)^2 - \operatorname{rad}(\sigma)^2$).

Let s_i and p_i denote the quantities $s_i := |S_i|$ and $p_i = \sum_{j=1}^{i-1} s_j$, for all i. Π is spanned by the permutations of a particular point in $\mathbb{R}^{d+1}$, defined in Section 3.1; we order these coordinates values by size in increasing order. Then, the indices in S_i will contain the coordinate values of order $p_i + 1, \ldots, p_i + s_i$. Let a_i denote the average of the coordinate values of orders $p_i + 1, \ldots, p_i + s_i$. The symmetric structure of Π implies that $bc(f)$ has value a_i in each coordinate $j \in S_i$. Doing the same construction for τ, we observe that the coordinates of $bc(f)$ and $bc(e)$ coincide for every coordinate $j \in S_1, \ldots, S_{k-1}$. The only differences appear for coordinate indices of S_k, that is, the partition set that was split to obtain τ from σ. Writing a_k, a'_k, a'_{k+1} for the average values for S_k, S'_k, S'_{k+1}, respectively, and $t := |S'_k|$, we get

$$\begin{aligned}\operatorname{rad}^2(\tau) - \operatorname{rad}^2(\sigma) &= \sum_{i=1}^{t} \left((a'_k)^2 - a_k^2\right) + \sum_{i=t+1}^{\ell} \left((a'_{k+1})^2 - a_k^2\right) \\ &= t\left((a'_k)^2 - a_k^2\right) + (\ell - t)\left((a'_{k+1})^2 - a_k^2\right)\end{aligned}$$

To obtain a_k, a'_k, and a'_{k+1}, we only need to explicitly compute the average of the appropriate coordinate values. A simple calculation shows that

$$a_k = \frac{(d+1)-\ell}{2(d+1)}, \quad a'_k = \frac{(d+1)-i}{2(d+1)}, \quad \text{and } a'_{k+1} = \frac{(d+1)-\ell-i}{2(d+1)}.$$

Plugging in these values yields

$$\mathrm{rad}^2(\tau) - \mathrm{rad}^2(\sigma) = \frac{(d+1+\ell)t(\ell-t)}{4(d+1)^2},$$

whose minimum is achieved for $t = 1$ and $t = \ell - 1$. Therefore,

$$\mathrm{rad}^2(\tau) - \mathrm{rad}^2(\sigma) \geq \frac{(d+1+\ell)(\ell-1)}{4(d+1)^2} \geq \frac{\ell-1}{4(d+1)}.$$

Moreover, $\mathrm{rad}(\tau) \leq 2\sqrt{d}$ by Lemma 4.3.4. This yields

$$\begin{aligned}\mathrm{rad}(\tau) - \mathrm{rad}(\sigma) &= \frac{\mathrm{rad}^2(\tau) - \mathrm{rad}^2(\sigma)}{\mathrm{rad}(\tau) + \mathrm{rad}(\sigma)} \geq \frac{\mathrm{rad}^2(\tau) - \mathrm{rad}^2(\sigma)}{2\mathrm{rad}(\tau)} \\ &\geq \frac{\ell-1}{16(d+1)\sqrt{d}} \geq \frac{\ell}{24(d+1)^{3/2}}\end{aligned}$$

for $\ell \geq 3$. The claim for D^*_σ follows.

For D_σ, note that $\min_{\gamma \text{ is facet of } \sigma} D^*_\gamma \leq D_\sigma$, so it is enough to bound D^*_γ for all facets γ of σ. The $(k-1)$-simplex σ has two kind of facets:

- $(k-2)$-facets which are attached to the origin $\mathbb{O}$. The faces of Π dual to these $(k-2)$-facets of σ are co-faces of f. As a result, these $(k-2)$-facets are obtained by merging two consecutive S_i and S_{i+1}. However, the obtained partition is again good (because σ is good), so the first claim yields the lower bound result for all these facets.

- The facet of σ which is opposite to $\mathbb{O}$. The face of the permutahedral tessellation dual to this facet of σ is not contained in Π, so the previous argument does not apply directly. To handle this case, we change the origin to any vertex of σ. Through the combinatorial properties of Π, it can be observed that with respect to the new origin, σ has the representation of the form $(S_j, \ldots, S_k, S_1, \ldots, S_{j-1})$, one for each choice of the $(k-1)$ origins, thus the partition is cyclically shifted.

  In particular, σ is still good with respect to the new origin. We obtain the missing facet by merging the (now consecutive) sets S_k and S_1, which is also a good face, and the first part of the statement implies the result.

The claims follow. ■

As a consequence of Theorem 4.3.6, the interval associated with a good simplex has length at least $\frac{\ell}{24(d+1)^{3/2}}$ using Lemma 2.4.6 and 2.4.7. Moreover, the interval cannot persist beyond the scale $2\sqrt{d}$ by Lemma 4.3.4. It follows

Corollary 4.3.7. *The interval associated to a good simplex is δ-significant for*

$$\delta < \frac{\ell}{24(d+1)^{3/2}} \frac{1}{4\sqrt{d}} < \frac{\ell}{192(d+1)^2}.$$

4.3.2 The number of good simplices

We assume for simplicity that $d+1$ is divisible by ℓ. We call a good partition $(S_1, \ldots, S_k)$ *uniform*, if each set consists of *exactly* ℓ elements. This implies that $k = (d+1)/\ell$.

Lemma 4.3.8. *The number of uniform good partitions is exactly* $\frac{(d+1)!}{\ell!^{(d+1)/\ell}}$.

Proof. Choose an arbitrary permutation and place the first ℓ entries in the S_1, the second ℓ entries in S_2, and so forth. In each S_i, we can interchange the elements and obtain the same k-simplex. Thus, we have to divide out $\ell!$ choices for each of the $(d+1)/\ell$ bins. ■

We use this result to bound the number of good k-simplices in the upcoming theorem.

Theorem 4.3.9. *For any constant* $\rho \in (0,1)$, $\ell = (d+1)^{\rho}$, $k = (d+1)/\ell$ *and* d *large enough, there exists a constant* $\lambda \in (0,1)$ *that only depends only on* ρ, *such that the number of good* k*-simplices is at least* $(d+1)^{\lambda(d+1)} = 2^{\Omega(d \log d)}$.

Proof. From Lemma 4.3.8, we know that the number of good simplices is at least $\frac{(d+1)!}{\ell!^{(d+1)/\ell}}$. Stirling's approximation [RD69] states that for any positive integer n,

$$\sqrt{2\pi} n^{n+1/2} e^{-n+1/(12n+1)} < n! < \sqrt{2\pi} n^{n+1/2} e^{-n+1/(12n)},$$

where $e \approx 2.71828$ is the Euler's number. We rephrase the upper bound as

$$\sqrt{2\pi} n^{n+1/2} e^{-n+1/(12n)} \le \sqrt{2\pi} e^{1/(12n)} n^{n+1/2} e^{-n} \le e n^{n+1/2} e^{-n}$$

for $n \ge 2$ and the lower bound simply as

$$\sqrt{2\pi} n^{n+1/2} e^{-n+1/(12n+1)} \ge n^n e^{-n}.$$

In this way, we can lower bound the number of good simplices as

$$\begin{aligned}
\frac{(d+1)!}{\ell!^{(d+1)/\ell}} &\ge \frac{(d+1)^{(d+1)} e^{-(d+1)}}{(e \ell^{\ell+1/2} e^{-\ell})^{(d+1)/\ell}} \\
&\ge \frac{(d+1)^{(d+1)} e^{-(d+1)}}{e^{(d+1)/\ell} \ell^{(d+1)+(d+1)/(2\ell)} e^{-(d+1)}} \\
&\ge \exp\left((d+1)\log(d+1) - \frac{(d+1)}{\ell} - (d+1)\log \ell (1 + \frac{1}{2\ell}) \right).
\end{aligned}$$

Choose $\ell = (d+1)^{\rho}$ with some constant $0 < \rho < 1$. The above simplifies to

$$\begin{aligned}
&\exp\left((d+1)\log(d+1) - (d+1)^{1-\rho} - \rho(d+1)\log(d+1)(1 + \frac{1}{2(d+1)^{\rho}}) \right) \\
&= \exp\left((d+1)\log(d+1)(1 - \rho(1 + \frac{1}{2(d+1)^{\rho}})) - (d+1)^{1-\rho} \right).
\end{aligned}$$

Now, pick some $\lambda \in [0,1]$ such that $\rho < 1 - 2\lambda < 1$. We have that

$$\rho \left(1 + \frac{1}{2(d+1)^{\rho}} \right) < 1 - 2\lambda$$

for d large enough. Thus, for d large enough,

$$\begin{aligned}
&\exp\left((d+1)\log(d+1)(1 - \rho(1 + \frac{1}{2(d+1)^{\rho}})) - (d+1)^{1-\rho} \right) \\
&\ge \exp\left(2\lambda(d+1)\log(d+1) - (d+1)^{1-\rho} \right) \\
&\ge \exp\left(\lambda(d+1)\log(d+1) \right),
\end{aligned}$$

which proves the claim. ■

Putting everything together, we prove our lower bound theorem:

Theorem 4.3.10. *There exists a point set of n points in $d = \Theta(\log n)$ dimensions, such that any $(1+\delta)$-approximation of its Čech filtration contains $2^{\Omega(d\log d)}$ intervals in its persistent barcode, provided that $\delta < \frac{1}{192(d+1)^{1+\varepsilon}}$ with an arbitrary constant $\varepsilon \in (0,1)$.*

Proof. Setting $\rho := 1-\varepsilon$, Theorem 4.3.9 guarantees the existence of $2^{\Omega(d\log d)}$ good simplices, all in a fixed dimension k. In particular, the intervals of the Čech persistence module associated to these intervals are all distinct. Since $\ell = (d+1)^{1-\varepsilon}$, Corollary 4.3.7 states that all these intervals are significant because $\delta < \frac{1}{192d^{1+\varepsilon}} = \frac{\ell}{192(d+1)^2}$. Therefore, by Lemma 4.3.2, any $(1+\delta)$-approximation of the Čech filtration has $2^{\Omega(d\log d)}$ intervals in its barcode. ■

Replacing d by $\log n$ in the bounds of theorem, we see the number of intervals appearing in any approximation is $n^{\Theta(\log\log n)}$, which is super-polynomial in n if δ is small enough.

4.4 Discussion

In this chapter, we presented upper and lower bound results on approximating Rips and Čech filtrations of point sets in arbitrarily high dimensions. For Čech complexes, the major result can be summarized as: for a dimension-independent bound on the complex size, there is no way to avoid a super-polynomial complexity for fine approximations of quality about $O(\log^{-1} n)$, while polynomial size can be achieved for rough approximation of quality about $O(\log^2 n)$.

Filling in the large gap between the two approximation factors is an attractive avenue for future work. A possible approach is to look at other lattices. It seems that lattices with good covering properties are correlated with a good approximation quality, and it may be worthwhile to study lattices in higher dimension which improve largely on the covering density of A^* (e.g., the Leech lattice [CSB87]).

It is however, noteworthy that the Permutahedral lattice is one among a very small number of lattices that are in general position. As a result, any technique which works with non-degenerate lattices has to use a smarter triangulation than just using the nerve of the non-empty Voronoi cells. Also, efficient algorithms for closest point location are only known for a few lattices. From an algorithmic viewpoint, any alternative lattice should be efficient in this regard.

Chapter 5

Approximation using Grids

In this chapter we present a new scheme to approximate the Rips filtration in Euclidean space, improving upon the approximation results of Chapter 4. To build our approximation tower, we use scaled and shifted versions of the integer lattice, which we defined earlier in Section 3. A particularly interesting contribution in this chapter is the application of acyclic carriers (see Definition 2.4.2) to prove the interleaving of the approximation tower with the Rips filtration. In Section 5.1 we define our approximation scheme and detail the connection with the Rips filtration. In Section 5.2 we discuss the computational aspects of computing the approximation tower.

5.1 Approximation scheme

We define our approximation complex for a finite set of points in $\mathbb{R}^d$. Recall from Definition 3.2.1 that we can define a collection of scaled and shifted integer grids G_{α_s} over a collection of scales $I := \{\alpha_s = 2^s \mid s \in \mathbb{Z}\}$ in $\mathbb{R}^d$. For shorter notation, we write s instead of α_s, and G_s instead of G_{α_s} when it is clear from context. Also, let $\square_s$ denote the cubical complex corresponding to G_s (see Definition 3.2.3). Let sd_s denote the barycentric subdivision of $\square_s$ (see Definition 3.2.6 for more details). To make the exposition simple, we define our complex in a slightly generalized form.

5.1.1 Barycentric spans

Fix some $s \in \mathbb{Z}$ and let V denote any non-empty subset of G_s.

Definition 5.1.1 (vertex span). *We say that a face $f \in \square_s$ is* spanned *by V, if the set of vertices $V(f) := f \cap V$*

- *is non-empty, and*
- *not contained in any facet of f.*

Trivially, the vertices of $\square_s$ which are spanned by V are precisely the points in V. Any face of $\square_s$ which is not a vertex must contain at least two vertices of V in order to be active. We point out that the set of spanned faces of $\square_s$ is *not* closed under taking sub-faces. For instance, if V consists of two antipodal points of a d-cube, the only faces spanned by V are the d-cube and the two vertices; all other faces of the d-cube contain at most one vertex and hence are not active.

A simple technique reveals whether any k-face $f \in \square_s$ is spanned. For any axis-aligned k-cube $\square$, there exists a partition of $\{1, \ldots, d\}$ into sets S, S' with $|S| = d - k$ and $|S'| = k$

such that $\square$ spans the sub-space of $\mathbb{R}^d$ defined by the co-ordinate directions of S'. The co-ordinates of all points of $\square$ are common for the indices in S. Since f is a k-cube, it also has a partition S, S' as in the case of $\square$. Let T be the set of common co-ordinates of the points in $V(f)$. T is always a superset of S. If T is a strict superset of S, then the points $V(f)$ lie in a lower dimensional face of f, which means that f is not active. If T is precisely S, then f is active. T can be computed in $|V(f)|O(d^2) = O(2^k d^2)$ time.

Definition 5.1.2 (barycentric span). *The* barycentric span *of V is the subcomplex of sd_s defined by all flags $\{f_0, \ldots, f_k\}$ of $\square_s$ such that each f_i is spanned by V.*

The barycentric span of V is indeed a subcomplex of sd_s because it is closed under taking subsets of V. For any face $f \in \square_s$ that is spanned by V, we define the *f-local barycentric span* of V as the set of all flags of the form $\{f_0, \ldots, f_k\}$ in the barycentric span such that $f_i \subseteq f$ for all i. This is a subcomplex both of the barycentric subdivision of $sd(f)$, and of the barycentric span of V. An elementary observation is that the f-local barycentric span is a flag complex.

Lemma 5.1.3. *For each face $f \in \square_s$, the f-local barycentric span of V is either empty or acyclic.*

Proof. We assume that the f-local barycentric span of V is not empty. Then, f contains a unique active face e of maximal dimension that is spanned by V.

A simplex in a simplicial complex K is called *maximal* if no other simplex in K contains it. It is well-known that if a simplicial complex K contains a vertex x that lies in every maximal simplex, then K is acyclic (in this case, K is called *star-shaped*).

In our situation, the vertex e belongs to every maximal simplex in the f-local barycentric span, because every simplex not containing e is a flag that can be extended by adding e to it. The claim follows. ■

Furthermore, for any non-empty subset $W \subseteq V$, it is easy to see that the faces of $\square_s$ that are spanned by W are also spanned by V. Consequently, the barycentric span of W is a subcomplex of the barycentric span of V.

5.1.2 Approximation complex

We denote by $P \subset \mathbb{R}^d$ a finite set of points. We define two maps:

- $a_s : P \to G_s$: for each point $p \in P$, we let $a_s(p)$ denote the grid point in G_s that is closest to p, that is, $p \in Vor_{G_s}(a_s(p))$. We assume for simplicity that this closest point is unique, which can be ensured using well-known methods [EM90]. We define the *active vertices of G_s* as

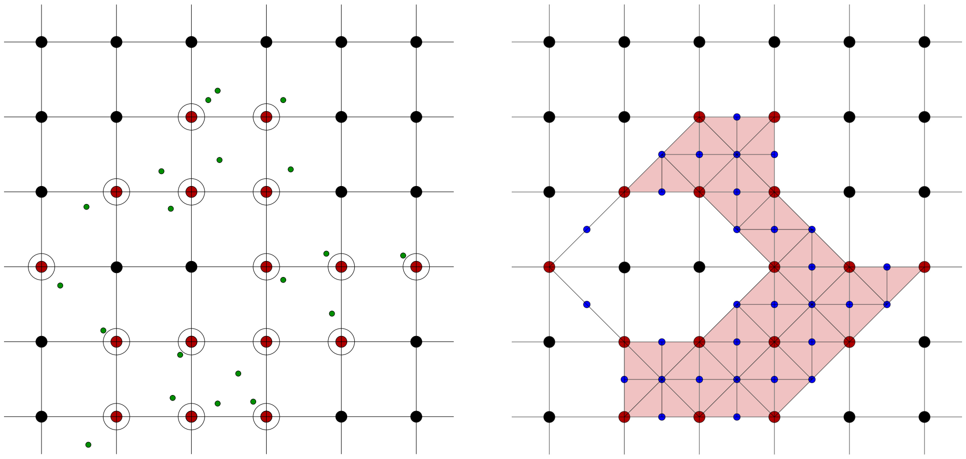
  $$V_s := \mathrm{im}(a_s) = a_s(P) \subset G_s,$$

  that is, the set of grid points that have at least one point of P in their Voronoi cells.

- $b_s : V_s \to P$: b_s takes an active vertex of G_s to its closest point in P . We assume for simplicity that the assignment is unique, which is easy to ensure by using an arbitrary total order on P.

Naturally, $b_s(v)$ is a point inside $Vor_{G_s}(v)$ for any $v \in V_s$. With this, it is easy to see that the map b_s is a section of a_s, that is, $a_s \circ b_s : V_s \to V_s$ is the identity on V_s. However, this is not true for $b_s \circ a_s$ in general.

Recall that the map $g_s : \square_s \to \square_{s+1}$ from Section 3.2 maps grid points of G_s to grid points of G_{s+1}. Using Lemma 3.2.2, it follows at once that:



(a) A two-dimensional grid, shown along with its cubical complex. The green points (small dots) denote the points in P and the red vertices (encircled) are the active vertices.

(b) The generated approximation complex, whose vertices are the encircled vertices from the left figure and the blue vertices (small dots). The blue vertices are the barycenters of active faces.

Figure 5.1: The approximation complex.

Lemma 5.1.4. *For all* $\alpha_s \in I$ *and each* $x \in V_s$, $g_s(x) = (a_{s+1} \circ b_s)(x)$.

Recall from Chapter 2 that $\mathcal{R}^{\infty}_{\alpha}$ denotes the Rips complex at scale α for the L_∞-norm. The next statement is a direct application of the triangle inequality; let $diam_\infty()$ denote the diameter in the L_∞-norm.

Lemma 5.1.5. *Let* $Q \subseteq P$ *be a non-empty subset such that* $diam_\infty(Q) \leq \alpha_s$. *Then, the set of grid points* $a_s(Q)$ *is contained in a face of* $\square_s$.

Equivalently, for any simplex $\sigma = (p_0, \ldots, p_k) \in \mathcal{R}^{\infty}_{\alpha_s/2}$ *on* P, *the set of active vertices* $\{a_s(p_0), \ldots, a_s(p_k)\}$ *is contained in a face of* $\square_s$.

Proof. We prove the claim by contradiction. Assume that the set of active vertices $a_s(Q)$ is not contained in a face of $\square_s$. Then, there exists at least one pair of points $\{x, y\} \in Q$ such that $a_s(x)$, $a_s(y)$ are not in a common face of $\square_s$. By the definition of the grid G_s, the grid points $a_s(x)$, $a_s(y)$ therefore have L_∞-distance at least $2\alpha_s$. Moreover, x has L_∞-distance less than $\alpha_s/2$ from $a_s(x)$, and the same is true for y and $a_s(y)$. By triangle inequality, the L_∞-distance of x and y is more than α_s, which is a contradiction to the fact that $diam_\infty(Q) \leq \alpha_s$. ■

We now define our approximation tower. For any scale α_s, we define $\mathcal{X}_{\alpha_s}$ as the barycentric span (Definition 5.1.2) of the active vertices $V_s \subset G_s$. See Figure 5.1 for a simple illustration.

To simplify notations, we call

- the faces of $\square_s$ spanned by V_s as *active faces*, and
- the simplices of $\mathcal{X}_{\alpha_s}$ as *active flags*.

To complete the description of the approximation tower, we need to define simplicial maps of the form $\tilde{g}_s : \mathcal{X}_{\alpha_s} \to \mathcal{X}_{\alpha_{s+1}}$, which connect the simplicial complexes at consecutive scales. We show that such maps are induced by g_s.

Lemma 5.1.6. *Let f be any active face of $\square_s$. Then, $g_s(f)$ is an active face of $\square_{s+1}$.*

Proof. Using Lemma 3.2.4 from Chapter 3, $e := g_s(f)$ is a face of $\square_{s+1}$. If e is a vertex, then it is active, because f contains at least one active vertex v, and $g_s(v) = e$ in this case. If e is not a vertex, we assume for a contradiction that it is not active. Then, it contains a facet e_1 that contains all active vertices in e. Let e_2 denote the opposite facet of e_1 in e. By Lemma 3.2.4, f contains opposite facets f_1, f_2 such that $g_s(f_1) = e_1$ and $g_s(f_2) = e_2$. Since f is active, both f_1 and f_2 contain active vertices; in particular, f_2 contains an active vertex v. But then the active vertex $g_s(v)$ must lie in e_2, contradicting the fact that e_1 contains all active vertices of e. ■

By definition, a simplex $\sigma \in \mathcal{X}_{\alpha_s}$ is a flag $(f_0 \subseteq \ldots \subseteq f_k)$ of active faces in $\square_s$. We set

$$\tilde{g}_s(\sigma) := (g_s(f_0), \ldots, g_s(f_k)),$$

where $(g_s(f_0) \subseteq \ldots \subseteq g_s(f_k))$ is the active flag of faces in $\square_{s+1}$ by Lemma 5.1.6, and hence is a simplex in $\mathcal{X}_{\alpha_{s+1}}$. It follows that $\tilde{g}_s : \mathcal{X}_{\alpha_s} \to \mathcal{X}_{\alpha_{s+1}}$ is a simplicial map. This completes the description of the simplicial tower

$$(\mathcal{X}_{2^s})_{s\in\mathbb{Z}}.$$

5.1.3 Interleaving with Rips module

First, we relate our tower with the L_∞-Rips filtration of P by showing that it gives rise to a constant-factor approximation. We then show that this also relates our approximation tower to the Euclidean Rips filtration of P.

We start by defining two acyclic carriers. We abbreviate $\alpha := \alpha_s = 2^s$ to simplify notation.

- $C_1^\alpha : \mathcal{R}^\infty_{\alpha/2} \to \mathcal{X}_\alpha$: for any simplex $\sigma = (p_0, \ldots, p_k)$ in $\mathcal{R}^\infty_{\alpha/2}$, we set $C_1^\alpha(\sigma)$ as the barycentric span of $U := \{a_s(p_0), \ldots, a_s(p_k)\}$, which is a subcomplex of $\mathcal{X}_\alpha$. Using Lemma 5.1.5, U lies in a face f of $\square_s$, so that $C_1^\alpha(\sigma)$ is also the f-local barycentric span of U. Using Lemma 5.1.3, we see that $C_1^\alpha(\sigma)$ is acyclic. The barycentric span of any subset of U is a subcomplex of the barycentric span of U (see the remark after Lemma 5.1.3), so C_1^α is a carrier. Therefore, C_1^α is an acyclic carrier.
- $C_2^\alpha : \mathcal{X}_\alpha \to \mathcal{R}^\infty_\alpha$: let σ be any flag $(e_0 \subseteq \ldots \subseteq e_k)$ of $\mathcal{X}_\alpha$. Let $\{q_0, \ldots, q_m\}$ be the set of active vertices of e_k. We set $C_2^\alpha(\sigma) := \{b_s(q_0), \ldots, b_s(q_m)\}$. We see that

  $$\begin{aligned}\|b_s(q_i) - b_s(q_j)\|_\infty &\le \|b_s(q_i) - q_i\|_\infty + \|q_i - q_j\|_\infty + \|q_j - b_s(q_j)\|_\infty \\ &< \alpha/2 + \alpha + \alpha/2 \le 2\alpha.\end{aligned}$$

  So, $C_2^\alpha(\sigma)$ is a simplex in $\mathcal{R}^\infty_\alpha$, hence it is acyclic. C_2^α is also a carrier by definition, so it is an acyclic carrier.

Using the acyclic carrier theorem (Theorem 2.4.3), there exist chain maps

$$c_1^\alpha : \mathcal{C}_*(\mathcal{R}^\infty_{\alpha/2}) \to \mathcal{C}_*(\mathcal{X}_\alpha) \quad \text{and} \quad c_2^\alpha : \mathcal{C}_*(\mathcal{X}_\alpha) \to \mathcal{C}_*(\mathcal{R}^\infty_\alpha),$$

between the chain complexes, which are carried by C_1^α and C_2^α respectively, for each $\alpha \in I$. Aggregating the chain maps, we have the following diagram:

$$\begin{tikzcd}
 & \ldots \arrow[r] & \mathcal{C}_*(\mathcal{R}^\infty_\alpha) \arrow[r, "inc"] \arrow[d, "c_1"] & \mathcal{C}_*(\mathcal{R}^\infty_{2\alpha}) \arrow[r] & \ldots \\
\ldots \arrow[r] & \mathcal{C}_*(\mathcal{X}_\alpha) \arrow[r, "\tilde{g}"] \arrow[ur, "c_2"] & \mathcal{C}_*(\mathcal{X}_{2\alpha}) \arrow[r] \arrow[ur, "c_2"] & \ldots &
\end{tikzcd} \tag{5.1}$$

where *inc* corresponds to the chain map for inclusion maps, and $\tilde{g}$ denotes the chain map for the corresponding simplicial maps (we removed indices of the maps for readability). The chain complexes give rise to a diagram of the corresponding homology groups, connected by the induced linear maps $c_1^*, c_2^*, inc^*, \tilde{g}^*$:

$$\begin{tikzcd}
 & \cdots \arrow[r] & H(\mathcal{R}^\infty_\alpha) \arrow[r, "inc^*"] \arrow[d, "c_1^*"] & H(\mathcal{R}^\infty_{2\alpha}) \arrow[r] & \cdots \\
\cdots \arrow[r] & H(\mathcal{X}_\alpha) \arrow[r, "\tilde{g}^*"] \arrow[ur, "c_2^*"] & H(\mathcal{X}_{2\alpha}) \arrow[r] \arrow[ur, "c_2^*"] & \cdots &
\end{tikzcd} \tag{5.2}$$

Lemma 5.1.7. *For all $\alpha \in I$, the linear maps in the lower triangle of Diagram* (5.2) *commute, that is,*

$$\tilde{g}^* = c_1^* \circ c_2^*.$$

Proof. To prove the claim, we look at the corresponding triangle in Diagram (5.1). We show that the chain maps $\tilde{g}$ and $c_1 \circ c_2$ are carried by a common acyclic carrier. The claim then follows from the acyclic carrier theorem.

We choose the map $C_1 \circ C_2 : \mathcal{X}_\alpha \to \mathcal{X}_{2\alpha}$. Since $C_2(\sigma)$ is a simplex for any simplex $\sigma \in \mathcal{X}_\alpha$, it is easy to see that $C_1 \circ C_2$ is an acyclic carrier. Clearly, $C_1 \circ C_2$ carries the map $c_1 \circ c_2$ by definition. We show that it also carries the map $\tilde{g}$.

Let σ be any flag $(f_0 \subseteq \ldots \subseteq f_k)$ in $\mathcal{X}_\alpha$ and let $V(f_i)$ denote the active vertices of f_i, for all i. Then, $C_1 \circ C_2(\sigma)$ is the barycentric span of

$$U := \{a_{s+1} \circ b_s(q) \mid q \in V(f_k)\} = \{g_s(q) \mid q \in V(f_k)\}$$

(using Lemma 5.1.4). On the other hand, $V(f_i) \subseteq V(f_k)$ so that $g(V(f_i)) \subseteq U$. Then, $g(f_i)$ is spanned by U: indeed, since f_i is active, $g(f_i)$ is active and hence spanned by all active vertices, and it remains spanned if we remove all active vertices not in U, since they are not contained in f_i. It follows that the flag $(g(f_0) \subseteq \ldots \subseteq g(f_k))$, which is equal to $\tilde{g}(\sigma)$, is in the barycentric span of U. This shows that $\tilde{g}$ is carried by $C_1 \circ C_2$, as required. ■

Lemma 5.1.8. *For all $\alpha \in I$, the linear maps in the upper triangle of Diagram* (5.2) *commute, that is,*

$$inc^* = c_2^* \circ c_1^*.$$

Proof. The proof technique is analogous to the proof of Lemma 5.1.7. We define an acyclic carrier $D : \mathcal{R}^\infty_\alpha \to \mathcal{R}^\infty_{2\alpha}$ which carries both *inc* and $c_2 \circ c_1$.

Let $\sigma = (p_0, \ldots, p_k) \in \mathcal{R}^\infty_\alpha$ be any simplex. The active vertices

$$U := \{a(p_0), \ldots, a(p_k)\} \subset G_{s+1}$$

lie in a face f of G_{s+1}, using Lemma 5.1.5. We can assume that f is active, as otherwise, we argue about a facet of f that contains U. We set $D(\sigma)$ as the simplex on the subset of points in P, whose closest grid point in G_{s+1} is any vertex of f. Using a simple application of triangle inequalities, $D(\sigma) \in \mathcal{R}^\infty_{2\alpha}$, so D is an acyclic carrier. The vertices of σ are a subset of $D(\sigma)$, so D carries the map *inc*. Showing that D carries $c_2 \circ c_1$ requires further explanation.

Let δ be any simplex in $\mathcal{X}_{2\alpha}$ for which the chain $c_1(\sigma)$ takes a non-zero value. Since $c_1(\sigma)$ is carried by $C_1(\sigma)$, we have that $\delta \in C_1(\sigma)$, which is a subcomplex of the f-local barycentric span. Furthermore, for any $\tau \in C_1(\sigma)$, $C_2(\tau)$ is of the form $\{b(q_0), \ldots, b(q_m)\}$ with $\{q_0, \ldots, q_m\}$ being vertices of f. It follows that $C_2(\tau) \subseteq D(\sigma)$. In particular, since c_2 is carried by C_2, $c_2(c_1(\sigma)) \subseteq D(\sigma)$ as well. ■

Using Lemma 5.1.7 and Lemma 5.1.8, we see that the persistence modules $\left(H(\mathcal{X}_{2^s})\right)_{s\in\mathbb{Z}}$ and $\left(H(\mathcal{R}^{\infty}_{\alpha})\right)_{\alpha\geq 0}$ are weakly 2-interleaved. Applying the scale balancing technique from Sub-section 2.3.3, this improves to a weak $\sqrt{2}$-interleaving.

With a minor modification in the definition of $\mathcal{X}$ and $\tilde{g}$, we can get a tower of the form $(\mathcal{X}_{\alpha})_{\alpha\geq 0}$. Further, with minor changes in the interleaving arguments, we show that the corresponding persistence module is strongly 2-interleaved with the L_{∞}-Rips module (Lemma A.0.5). Since the techniques used in the proof are very similar to the concepts used in this section, we defer all further details to Appendix A.

Using the strong stability theorem for persistence modules (Theorem 2.3.6) and taking scale balancing into account, we immediately get that:

Theorem 5.1.9. *The persistence module* $\left(H(\mathcal{X}_{2\alpha})\right)_{\alpha\geq 0}$ *and the* L_{∞}*-Rips persistence module* $\left(H(\mathcal{R}^{\infty}_{\alpha})\right)_{\alpha\geq 0}$ *are* 2*-approximations of each other.*

For any pair of points $p, p' \in \mathbb{R}^d$, it holds that

$$\|p - p'\|_2 \leq \|p - p'\|_{\infty} \leq \sqrt{d}\,\|p - p'\|_2.$$

With this, it is easy to infer that the L_2- and the L_{∞}-Rips filtrations are strongly $\sqrt{d}$-interleaved. Using the scale balancing technique for strongly interleaved persistence modules, we get:

Lemma 5.1.10. *The persistence modules* $(H(\mathcal{R}_{\alpha/d^{0.25}}))_{\alpha\geq 0}$ *and* $(H(\mathcal{R}^{\infty}_{\alpha}))_{\alpha\geq 0}$ *are strongly* $d^{0.25}$*-interleaved.*

Using Theorem 5.1.9, Lemma 5.1.10 and the fact that interleavings satisfy the triangle inequality [BS14, Theorem 3.3], we see that $(H(\mathcal{X}_{2\alpha}))_{\alpha\geq 0}$ is strongly $2d^{0.25}$-interleaved with the scaled Rips module $(H(\mathcal{R}_{\alpha/d^{0.25}}))_{\alpha\geq 0}$. We can remove the scaling in the Rips filtration simply by multiplying the scales on both sides with $d^{0.25}$ and obtain our final approximation result:

Theorem 5.1.11. *The persistence module* $\left(H(\mathcal{X}_{2\sqrt[4]{d}\alpha})\right)_{\alpha\geq 0}$ *and the Euclidean Rips persistence module* $\left(H(\mathcal{R}_{\alpha})\right)_{\alpha\geq 0}$ *are* $2d^{0.25}$*-approximations of each other.*

5.2 Computational complexity

In this section we discuss the computational aspects of constructing the approximation tower. In Sub-section 5.2.1 we discuss the size complexity of the tower. An algorithm to compute the tower efficiently is present in Sub-section 5.2.2.

Range of relevant scales Set $n := |P|$ and let $CP(P)$ denote the closest pair distance of P. At scale $\alpha_0 := \frac{CP(P)}{3d}$ and lower, no two active vertices lie in the same face of the grid, so the approximation complex consists of n isolated 0-simplices. At scale $\alpha_m := diam(P)$ and higher, points of P map to active vertices of a common face (by Lemma 5.1.5), so the generated complex is acyclic using Lemma 5.1.3. We inspect the range of scales $[\alpha_0, \alpha_m]$ to construct the tower, since the barcode is explicitly known for scales outside this range. The total number of scales is

$$\lceil \log_2 \alpha_m/\alpha_0 \rceil = \lceil \log_2 \Delta + \log_2 3d \rceil = O(\log \Delta + \log d).$$

5.2.1 Size of the tower

The size of a tower is the number of simplices that do not have a preimage, that is, the number of simplex inclusions in the tower (see Chapter 2). We start by considering the case of 0-simplices of the approximation tower.

Lemma 5.2.1. *The number of vertices included in the tower is at most $n2^{O(d)}$.*

Proof. Recall that the vertices of $\mathcal{X}_\alpha$ are the active faces of the cubical complex $\square_\alpha$ at the same scale, and that the simplicial map $\tilde{g}$ restricted to the vertices corresponds to the cubical map g acting on the active faces of $\square_\alpha$.

We first consider the active vertices. At scale α_0, there are n inclusions of 0-simplices in the tower, due to n active vertices. Using Lemma 3.2.2, g is surjective on the active vertices of $\square$ (for any scale). Hence, no further active vertices are added to the tower.

It remains to count the active faces of dimension ≥ 1 without preimage. We will use a charging argument, charging the existence of such an active face to one of the points in P. We show that each point of P is charged at most 3^d times, which proves the claim. For that, we first fix an arbitrary total order $\prec$ on P. Each active vertex on any scale has a non-empty subset of P in its Voronoi region; we call the maximal such point with respect to the order $\prec$ the *representative* of the active vertex. For an active face f (of dimension one or higher) without preimage under g, f has at least two incident active vertices, with distinct representatives. We charge the inclusion of f to the minimal representative among the incident active vertices.

Let M be the number of faces incident to a vertex in the cubical complex $\square$ (for any scale). As one can easily see with combinatorial arguments, $M = 3^d$. Assume for a contradiction that a point $p \in P$ is charged more than M times. Whenever any face f_i is charged to p, there is an active vertex v_i whose representative is p. We enumerate these as the set of active vertices $\{v_0, \ldots, v_m\}$ on the scales $\alpha_0, \ldots, \alpha_m$ such that p is the representative of v_i on scale α_i, for all i. Naturally, for any v_i and v_j, there is a canonical isomorphism between the M faces incident to v_i and the M faces incident to v_j.

Since we assumed that p is charged for $> M$ active faces, by pigeonhole principle, there must be two vertices v_i and v_j with $i < j$ such that a pair of isomorphic incident faces are charged for v_i and for v_j. There is a sequence of isomorphic faces $f_i, f_{i+1}, \ldots, f_j$ corresponding to $v_i, v_{i+1}, \ldots, v_j$, respectively, such that p is charged for f_i and f_j. Since f_i and f_j both have no preimage, there must be some f_ℓ with $i < \ell < j$ such that f_ℓ is not active. That means, however, that the Voronoi region of v_ℓ is the union of at least two Voronoi regions of vertices incident to v_i. In that case, because we choose the representative by minimizing over the maximal representatives, we see that p is not the representative of v_ℓ, and hence, not of v_j. This is a contradiction to our claim that p is the representative of v_ℓ, so p can not be charged more than M times. ∎

The next lemma follows from a simple combinatorial counting argument for the number of flags in a d-dimensional cube.

Lemma 5.2.2. *Each vertex of $\mathcal{X}_\alpha$ has at most $2^{O(d \log k)}$ incident k-simplices.*

Proof. A vertex in $\mathcal{X}_\alpha$ corresponds to an active face f in the cubical complex $\square_\alpha$. A simplex incident to f corresponds to an active flag of $\square_\alpha$ involving f. Let c be a d-cube of $\square_\alpha$ that contains f. We simply count the number of flags of length $(k+1)$ contained in c (regardless of whether they contain f or not) and show that this number is $2^{O(d \log k)}$. Since f is contained in at most 2^d d-cubes, the bound in our claim follows.

To count the number of flags contained in c, we use similar ideas as in [EK12]: first, we fix any vertex v of c and count the flags of the form $v \subseteq \ldots \subseteq c$. Every ℓ-face in c incident to v corresponds to a subset of ℓ coordinate indices, in the sense that the coordinates not chosen are fixed to the coordinates of v for the face. With this correspondence, it is not hard to see that a flag from v to c of length $(k+1)$ corresponds to an ordered k-partition of $\{1, \ldots, d\}$. The number of such partitions is known as $k!$ times the quantity $\left\{ \begin{matrix} d \\ k \end{matrix} \right\}$, which is the Stirling number of second kind [RD69], and is upper bounded by $2^{O(d \log k)}$ [CKR16]. Since c has 2^d vertices, the total number of flags $v \subseteq \ldots \subseteq c$ of length $(k+1)$ with any vertex v is hence $2^d k! 2^{O(d \log k)} = 2^{O(d \log k)}$.

For flags which do not start with a vertex and do not end with c, we can simply extend them by adding a vertex and/or the d-cube and obtain flags of length $k+2$ or $k+3$. The above argument again shows that the number of such flags is bounded by $2^{O(d \log k)}$. ■

Theorem 5.2.3. *The k-skeleton of the tower has size at most $n 2^{O(d \log k)}$.*

Proof. Let $\sigma = (f_0 \subseteq \ldots \subseteq f_k)$ be a flag included at some scale α in the tower. The crucial insight is that this can only happen if at least one face f_i in the flag is included in the tower at scale α. Indeed, if each f_i has a preimage e_i on the previous scale, then $(e_0 \subseteq \ldots \subseteq e_k)$ is a flag on the previous scale that maps to σ under $\tilde{g}$, which is a contradiction to the inclusion of σ at scale α.

We charge the inclusion of the flag to the inclusion of f_i. Clearly, f_i is only charged at the scale at which it is included in the tower. By Lemma 5.2.2, the vertex f_i of $\mathcal{X}$ is charged at most $\sum_{i=1}^{k} 2^{O(d \log i)} = 2^{O(d \log k)}$ times in this way, and by Lemma 5.2.1, there are at most $n 2^{O(d)}$ vertices that can be charged. The claim follows. ■

5.2.2 Computing the tower

From Section 3.2, we know that G_{s+1} is built from G_s by making use of an arbitrary translation vector $(\pm 1, \ldots, \pm 1) \in \mathbb{Z}^d$. In our algorithm, we pick the components of this translation vector uniformly at random from $\{+1, -1\}$, and independently for each scale. The choice behind choosing this vector randomly becomes more clear in the next lemma.

From the definition, it is easy to see that the cubical maps $g_s : \square_s \to \square_{s+1}$ can be composed for multiple scales. For a fixed s, we denote by $g^{(j)} : \square_s \to \square_{s+j}$ the j-fold composition of g, that is,

$$g^{(j)} = g_{s+j-1} \circ g_{s+j-2} \circ \ldots \circ g_{s+1} \circ g_s,$$

for $j \geq 1$.

Lemma 5.2.4. *For any k-face $f \in \square_s$ with $1 \leq k \leq d$, let Y denote the minimal integer j such that $g^{(j)}(f)$ is a vertex, for a given choice of the randomly chosen translation vectors. Then, the expected value of Y satisfies*

$$\mathbb{E}[Y] \leq 3 \log k,$$

which implies that no face of $\square_s$ survives *more than $3 \log d$ scales in expectation.*

Proof. Without loss of generality, assume that the grid under consideration is $\mathbb{Z}^d$ and f is the k-face spanned by the vertices $\{\underbrace{\{0,1\}, \ldots, \{0,1\}}_{k}, 0, \ldots, 0\}$, so that the origin is a vertex of f. The proof for the general case is analogous.

Let $y_1 \in \{-1, 1\}$ denote the randomly chosen first coordinate of the translation vector, so that the corresponding shift is one of $\{-1/2, 1/2\}$.

- If $y_1 = 1$, then the grid G' on the next scale has some grid point with x_1-coordinate $1/2$. Clearly, the closest grid point in G' to the origin is of the form $(+1/2, \pm 1/2, \ldots, \pm 1/2)$, and thus, this point is also closest to $(1, 0, 0, \ldots, 0)$. The same is true for any point $(0, *, \ldots, *)$ and its corresponding point $(1, *, \ldots, *)$ on the opposite facet of f. Hence, for $y_1 = 1$, $g(f)$ is a face where all points have the same x_1-coordinate.
- Contrarily, if $y_1 = -1$, the origin is mapped to some point which has the form $(-1/2, \pm 1/2, \ldots, \pm 1/2)$ and $(1, 0, \ldots, 0)$ is mapped to $(3/2, \pm 1/2, \ldots, \pm 1/2)$, as one can directly verify. Hence, in this case, in $g(f)$, points do not all have the same x_1 coordinate.

We say that the x_1-coordinate *collapses* in the first case and *survives* in the second. Both events occur with the same probability $1/2$. Because the shift is chosen uniformly at random for each scale, the probability that x_1 did not collapse after j iterations is $1/2^j$.

f spans k coordinate directions, so it must collapse along each such direction to contract to a vertex. Once a coordinate collapses, it stays collapsed at all higher scales. As the random shift is independent for each coordinate direction, the probability of a collapse is the same along all coordinate directions that f spans. Using union bound, the probability that $g^j(f)$ has not collapsed to a vertex is at most $k/2^j$. With Y as in the statement of the lemma, it follows that

$$P(Y \geq j) \leq k/2^j.$$

Hence,

$$\begin{aligned}
\mathbb{E}[Y] &= \sum_{j=1}^{\infty} jP(Y = j) = \sum_{j=1}^{\infty} P(Y \geq j) \\
&\leq \log k + \sum_{c=1}^{\infty} \sum_{j=c\log k}^{(c+1)\log k} P(Y \geq j) \\
&\leq \log k + \sum_{c=1}^{\infty} \sum_{j=c\log k}^{(c+1)\log k} P(Y \geq c\log k) \\
&\leq \log k + \sum_{c=1}^{\infty} \log k \frac{k}{2^{c\log k}} \\
&\leq \log k + \log k \sum_{c=1}^{\infty} \frac{1}{k^{c-1}} \\
&\leq \log k + 2\log k \leq 3\log k.
\end{aligned}$$

■

As a consequence of the lemma, the expected "lifetime" of k-simplices in our tower with $k > 0$ is rather short: given a flag $e_0 \subseteq \ldots \subseteq e_\ell$, the face e_ℓ will be mapped to a vertex after $O(\log d)$ steps, and so will be all its sub-faces, turning the flag into a vertex. It follows that summing up the total number of k-simplices with $k > 0$ over all $\mathcal{X}_\alpha$ yields an upper bound of $n2^{O(d\log k)}$ as well.

Algorithm description

Recall that a simplicial map can be written as a composition of simplex inclusions and contractions of vertices [DFW14, KS17] (see also Chapter 2). That means, given the complex $\mathcal{X}_{\alpha_s}$, to describe the complex at the next scale α_{s+1}, it suffices to specify

- which pairs of vertices in $\mathcal{X}_{\alpha_s}$ map to the same image under $\tilde{g}$, and
- which simplices in $\mathcal{X}_{\alpha_{s+1}}$ are included at scale $\mathcal{X}_{\alpha_{s+1}}$.

The input is a set of n points $P \subset \mathbb{R}^d$. The output is a list of *events*, where each event is of one of the three following types:

- A *scale event* defines a real value α and signals that all upcoming events happen at scale α (until the next scale event).
- An *inclusion event* introduces a new simplex, specified by the list of vertices on its boundary (we assume that every vertex is identified by a unique integer).
- A *contraction event* is a pair of vertices (i, j) from the previous scale, and signifies that i and j are identified as the same from that scale.

In a first step, we estimate the range of scales that we are interested in. We compute a 2-approximation of $diam(P)$ by taking any point $p \in P$ and calculating $max_{q\in P}\|p - q\|$. Then we compute $CP(P)$ using a randomized algorithm in $n2^{O(d)}$ expected time [KM95].

Next, we proceed scale-by-scale and construct the list of events accordingly. On the lowest scale, we simply compute the active vertices by point location for P in a cubical grid, and enlist n inclusion events (this is the only step where the input points are considered in the algorithm).

For the data structure, we use an auxiliary container S and maintain the invariant that whenever a new scale is considered, S consists of all simplices of the previous scale, sorted by dimension. In S, for each vertex, we store an id and a coordinate representation of the active face to which it corresponds. Every ℓ-simplex with $\ell > 0$ is stored just as a list of integers, denoting its boundary vertices. We initialize S with the n active vertices at the lowest scale.

Let $\alpha < \alpha'$ be any two consecutive scales with $\square, \square'$ the respective cubical complexes and $\mathcal{X}, \mathcal{X}'$ the approximation complexes, with $\tilde{g} : \mathcal{X} \to \mathcal{X}'$ being the simplicial map connecting them. Suppose we have already constructed all events at scale α.

- First, we enlist the scale event for α'.
- Then, we enlist the contraction events. For that, we iterate through the vertices of $\mathcal{X}$ and compute their value under g, using point location in a cubical grid. We store the results in a list S' (which contains the simplices of $\mathcal{X}'$). If for a vertex j, $g(j)$ is found to be equal to $g(i)$ for a previously considered vertex i, we choose the minimal such i and enlist a contraction event for (i, j).
- We turn to the inclusion events:
    - We start with the case of vertices. Every vertex of $\mathcal{X}'$ is an active face of $\square'$ and must contain an active vertex, which is also a vertex of $\mathcal{X}'$. We iterate through the elements in S'. For each active vertex v encountered, we go over all faces of the cubical complex $\square'$ that contain v as a vertex, and check whether they are active. For every active face encountered that is not in S' yet, we add it to S' and enlist an inclusion event of a new 0-simplex. At termination, all vertices of $\mathcal{X}'$ have been detected.
    - Next, we iterate over the simplices of S of dimension ≥ 1, and compute their image under $\tilde{g}$ using the pre-computed vertex map; we store the result in S'.

- To find the simplices of dimension ≥ 1 included at $\mathcal{X}'$, we exploit our previous insight that they contain at least one vertex that is included at the same scale (see the proof of Theorem 5.2.3). Hence, we iterate over the vertices included in $\mathcal{X}'$ and find the included simplices inductively in dimension.

  Let v be the current vertex under consideration; assume that we have found all $(p-1)$-simplices in $\mathcal{X}'$ that contain v. Each such $(p-1)$-simplex σ is a flag of length p in $\square'$. We iterate over all faces e that extend σ to a flag of length $p+1$. If e is active, we have found a p-simplex in $\mathcal{X}'$ incident to v. If this simplex is not in S' yet, we add it and enlist an inclusion event for it. We also enqueue the simplex in our inductive procedure, to look for $(p+1)$-simplices in the next round. At the end of the procedure, we have detected all simplices in $\mathcal{X}'$ without preimage, and S' contains all simplices of $\mathcal{X}'$. We set $S \leftarrow S'$ and proceed to the next scale.

This ends the description of the algorithm.

Theorem 5.2.5. *To compute the k-skeleton, the algorithm takes*

$$n2^{O(d)} \log \Delta + 2^{O(d)} M$$

time in expectation and M space, where M denotes the size of the tower. In particular, the expected time is bounded by

$$n2^{O(d)} \log \Delta + n2^{O(d \log k)}$$

and the space is bounded by $n2^{O(d \log k)}$.

Proof. In the analysis, we ignore the costs of point locations in grids, checking whether a face is active, and searches in data structures S, since all these steps have negligible costs when appropriate data structures are chosen.

Computing the image of a vertex of $\mathcal{X}$ costs $O(2^d)$ time. Moreover, there are at most $n2^{O(d)}$ vertices altogether in the tower in expectation (Lemma 5.2.1), so this bound in particular holds on each scale. Hence, the contraction events on a fixed scale can be computed in $n2^{O(d)}$ time. Finding new vertices requires iterating over the cofaces of a vertex in a cubical complex. There are 3^d such cofaces for each vertex. This has to be done for a subset of the vertices in $\mathcal{X}'$, so the running time is also $n2^{O(d)}$. Since there are $O(\log \Delta + \log d)$ scales considered, these steps require $n2^{O(d)} \log \Delta$ over all scales.

Computing the image of $\tilde{g}$ for a fixed scale costs at most $O(2^d|\mathcal{X}|)$. M is the size of the tower, that is, the simplices without preimage, and I is the set of scales considered. The expected bound for $\sum_{\alpha \in I} |\mathcal{X}_\alpha| = O(\log dM)$, because every simplex has an expected lifetime of at most $3 \log d$ by Lemma 5.2.4. Hence, the cost of these steps is bounded by $2^{O(d)} M$.

In the last step of the algorithm, we find the simplices of $\mathcal{X}'$ included at α'. We consider a subset of simplices of $\mathcal{X}'$, and for each, we iterate over a collection of faces in the cubical complex of size at most $2^{O(d)}$. Hence, this step is also bounded by $2^{O(d)}|\mathcal{X}|$ per scale, and hence bounded $2^{O(d)} M$ as well.

For the space complexity, the auxiliary data structure S gets as large as $\mathcal{X}$, which is clearly bounded by M. For the output complexity, the number of contraction events is at most the number of inclusion events, because every contraction removes a vertex that has been included before. The number of inclusion events is the size of the tower. The number of scale events as described is $O(\log \Delta + \log d)$. However, it is simple to get rid of this factor by only including scale events in the case that at least one inclusion or contraction takes place at that scale. The space complexity bound follows. ■

5.2.3 Dimension reduction

When the ambient dimension d is large, our approximation scheme can be combined dimension reduction techniques to reduce the final complexity, very similar to the application in Chapter 4. For a set of n points $P \subset \mathbb{R}^d$, we apply the dimension reduction schemes of Johnson-Lindenstrauss (JL) (Theorem 3.5.2), Matoušek (Theorem 3.5.3), and Bourgain's embedding (Theorem 3.5.6). We only state the main results in Table 5.1, leaving out the proofs since they are very similar to those from Chapter 4 and Section 3.5.

technique	approximation ratio	size	runtime
JL	$O(\log^{0.25} n)$	$n^{O(\log k)}$	$n^{O(1)} \log \Delta + n^{O(\log k)}$
Matoušek	$O((\log n)^{3/4} (\log \log n)^{1/4})$	$n^{O(1)}$	$n^{O(1)} \log \Delta$
Bourgain + Matoušek	$O((\log n)^{7/4} (\log \log n)^{1/4})$	$n^{O(1)}$	$n^{O(1)} \log \Delta$

Table 5.1: Comparison of dimension reduction techniques: here JL stands for Johnson-Lindenstrauss, the approximation ratio is for the Rips module, and the size refers to the size of the k-skeleton of the approximation.

5.3 Discussion

In this chapter we presented an approximation scheme for the Rips filtration, with improved approximation ratio, size and computational complexity than previous approaches for the case of high-dimensional point clouds. In particular, we improved upon the scheme of using the permutahedral tessellation in Chapter 4.

An important technique that we used in our scheme is the application of acyclic carriers to prove interleaving results. An alternative would to be explicitly construct chain maps between the Rips and the approximation towers; unfortunately, this make the interleaving analysis significantly more complex. While the proof of the interleaving in Section 5.1.3 is still technically challenging, it greatly simplifies by the usage of acyclic carriers. There is also no benefit in knowing the interleaving maps because they are only required for the analysis of the interleaving, and not for the actual computation of the approximation tower. We believe that this technique is of general interest for the construction of approximations of cell complexes.

Our tower is connected by simplicial maps; there are (implemented) algorithms to compute the barcode of such towers [DFW14, KS17]. It is also quite easy to adapt our tower construction to a streaming setting [KS17], where the output list of events is passed to an output stream instead of being stored in memory.

Our approximation scheme has more freedom over previous approaches [CKR16, DFW14, She13], since they used simplicial maps for the interleaving, which induce an elementary form of chain maps and are therefore more restrictive. An interesting question is whether persistence can be computed efficiently for more general chain maps, which would allow even more freedom in building approximation schemes.

Chapter 6

Digitization

This chapter presents a slightly different approach towards approximating filtrations, compared to the techniques presented in chapters 4 and 5. We present a $(1+\delta)$-approximation of the Čech filtration of point sets in Euclidean spaces, for any desired constant $0 < \delta \leq 1$. This is in strong contrast to the previous schemes presented in this thesis, where the approximation ratio depended on the ambient dimension of the point set. On the flip side, the scheme has a significantly higher computational complexity. Our scheme uses a digitization of $\mathbb{R}^d$ using the integer lattice.

6.1 Approximation with cubical pixels

We describe our approximation scheme for a set of n points $P \subset \mathbb{R}^d$. For a fixed scale $\alpha > 0$, let $L_\alpha := (\frac{\varepsilon\alpha}{3\sqrt{d}})\mathbb{Z}^d$ denote a scaled grid lattice in $\mathbb{R}^d$, that is, the grid whose basis vectors have been scaled by $\frac{\varepsilon\alpha}{3\sqrt{d}}$, for some $\varepsilon \in (0, \sqrt{2}-1]$. Throughout this chapter, we assume ε to be a quantity in this interval. The Voronoi cells of this lattice are cubes centered at the grid points, each of sidelength $\frac{\varepsilon\alpha}{3\sqrt{d}}$ and diameter $\varepsilon\alpha/3$.

Let $B(p,\alpha)$ denote the Euclidean ball of radius α, centered at any input point $p \in P$. We denote by

$$\mathcal{B}_\alpha := (B(p,\alpha))_{p\in P},$$

the set of α-balls centered at the input points. Naturally, the Čech complex on P at scale α is the nerve of $\mathcal{B}_\alpha$, that is, $\mathcal{C}_\alpha = \mathrm{nerve}(\mathcal{B}_\alpha)$ (Definition 2.1.13). Let $\alpha_0 > 0$ denote a finite real. We denote by

$$I := \{\alpha_k = \alpha_0(1+\varepsilon)^k \mid k \in \mathbb{Z}\}$$

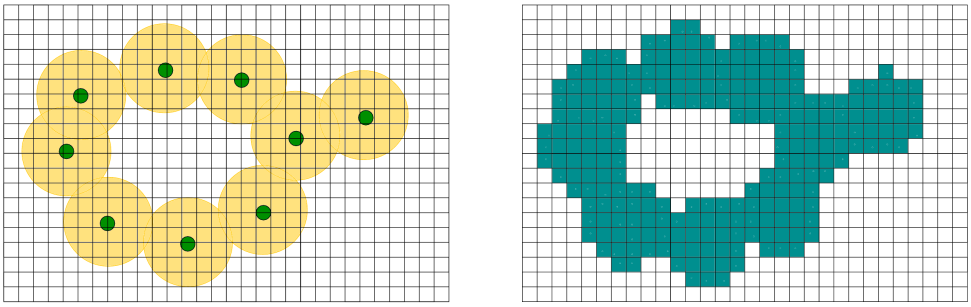
a discrete set of scales.

6.1.1 Approximation complex

We define our approximation structures for each scale in $\mathbb{R}^+$:

- for any $\alpha \in I$, consider the set of lattice points $V_\alpha := L_\alpha \cap \mathcal{B}_\alpha$, that is, those lattice points which lie in $\mathcal{B}_\alpha$. We call V_α the *digital* vertices at scale α. Let $\mathcal{S}_\alpha$ denote the set of cubes (Voronoi cells) centered at points of V_α; we call these cubes as the *pixels* at scale α. We define our approximation complex at scale α as the nerve $\mathcal{X}_\alpha := \mathrm{nerve}(\mathcal{S}_\alpha)$.

- for any $\alpha \in [\alpha_i, \alpha_{i+1})$, we define $\mathcal{S}_\alpha := \mathcal{S}_{\alpha_i}$, and $\mathcal{X}_\alpha := \mathcal{X}_{\alpha_i}$, for all $i \in \mathbb{Z}$.



(a) A collection of balls (shaded disks) centered at the input points (small shaded disks).

(b) The pixels corresponding to the union of balls.

Figure 6.1: An example of digitization in the plane.

Therefore, the pixels and the approximation complexes change only at discrete scales. In particular, all pixels and the respective complexes for the range $[\alpha_i, \alpha_{i+1})$ are the same. For an elementary example, see Figure 6.1.

For simplicity, we identify a pixel with its corresponding lattice point, which we call the *center* of the cube. Also, we drop the scale subscript from $\mathcal{S}_\alpha$ and $\mathcal{X}_\alpha$, when it is clear from context. Before we build towers on $\mathcal{S}$ and $\mathcal{X}$ at different scales, we need the following result:

Lemma 6.1.1 (sandwich lemma). *For any* $\alpha \in I$*,*

$$\mathcal{B}_{(1-\varepsilon/6)\alpha} \subseteq \mathcal{S}_\alpha \subseteq \mathcal{B}_{(1+\varepsilon/6)\alpha}.$$

In particular, this implies that

$$\mathcal{S}_\alpha \subseteq \mathcal{B}_{(1+\varepsilon/6)\alpha} \subseteq \mathcal{B}_{(1+\varepsilon)(1-\varepsilon/6)\alpha} \subseteq \mathcal{S}_{(1+\varepsilon)\alpha}.$$

Proof. For the first statement, we show two inclusions:

- For the first inclusion $\mathcal{B}_{(1-\varepsilon/6)\alpha} \subseteq \mathcal{S}_\alpha$, fix any point $x \in \mathcal{B}_{(1-\varepsilon/6)\alpha}$. Let p denote the closest input point to x. Then, $\|x-p\| \leq (1-\varepsilon/6)\alpha$. The point x lies in some cube of the grid L_α. Let the center of that cube be c. Since the diameter of the cube is at most $\varepsilon\alpha/3$, $\|x-c\| \leq \varepsilon\alpha/6$. By triangle inequality, $\|c-p\|$ is at most α. Hence, the cube, and thus x, lies in $\mathcal{S}_\alpha$.

- For the second inclusion $\mathcal{S}_\alpha \subseteq \mathcal{B}_{(1+\varepsilon/6)\alpha}$, let y be any point in $\mathcal{S}_\alpha$. Let y lie in some cube of $\mathcal{S}_\alpha$ with center c, so that $\|y-c\| \leq \varepsilon\alpha/6$. Since the cube is in $\mathcal{S}_\alpha$, $\|c-p\| \leq \alpha$ for some $p \in P$. Using triangle inequality, we obtain $\|x-p\| \leq (1+\varepsilon/6)\alpha$, which implies that $x \in \mathcal{B}_{(1+\varepsilon/6)\alpha}$.

The second statement follows from the fact that $\frac{(1+\varepsilon/6)}{(1-\varepsilon/6)} \leq (1+\varepsilon)$ holds for $0 \leq \varepsilon \leq 1$. ■

Therefore, there is a natural inclusion from $\mathcal{S}_\alpha$ to $\mathcal{S}_{(1+\varepsilon)\alpha}$, when $\alpha \in I$. In the range $[\alpha, (1+\varepsilon)\alpha)$, $\mathcal{S}$ does not change, so in general we have an inclusion from $\mathcal{S}_\beta$ to $\mathcal{S}_\gamma$ for any $0 < \beta \leq \gamma$. With this observation and the definition of $\mathcal{B}$, we can construct two continuous filtrations,

$$(\mathcal{S}_\alpha)_{\alpha>0} \quad \text{and} \quad (\mathcal{B}_\beta)_{\beta\geq 0},$$

both connected with inclusion maps.

Theorem 6.1.2. *The persistence module* $(H(\mathcal{S}_\alpha))_{\alpha>0}$ *is a* $(1+\delta)$*-approximation of the persistence module* $(H(\mathcal{B}_\alpha))_{\alpha\geq 0}$*; here* $(1+\delta) := (1+\epsilon)^2$ *which implies that* $0<\delta\leq 1$.

Proof. Consider the diagrams

$$\begin{tikzcd}
\mathcal{B}_{\frac{\alpha}{c}} \arrow[rrr, hook] \arrow[dr, hook] & & & \mathcal{B}_{c\alpha'} \\
& \mathcal{S}_\alpha \arrow[r, hook] & \mathcal{S}_{\alpha'} \arrow[ur, hook] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathcal{B}_{c\alpha} \arrow[r, hook] & \mathcal{B}_{c\alpha'} \\
\mathcal{S}_\alpha \arrow[ur, hook] \arrow[r, hook] & \mathcal{S}_{\alpha'} \arrow[ur, hook] &
\end{tikzcd} \tag{6.1}$$

$$\begin{tikzcd}
& \mathcal{B}_\alpha \arrow[r, hook] & B_{\alpha'} \arrow[dr, hook] & \\
\mathcal{S}_{\frac{\alpha}{c}} \arrow[ur, hook] \arrow[rrr, hook] & & & \mathcal{S}_{c\alpha'}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{B}_\alpha \arrow[r, hook] \arrow[dr, hook] & \mathcal{B}_{\alpha'} \arrow[dr, hook] & \\
& \mathcal{S}_{c\alpha} \arrow[r, hook] & \mathcal{S}_{c\alpha'}
\end{tikzcd}$$

where each arrow is an inclusion, $c=(1+\varepsilon)^2$ and $0<\alpha\leq\alpha'$. A quick inspection shows that each diagram commutes, for all $0<\alpha\leq\alpha'$, using Lemma 6.1.1. The claim follows from the strong-interleaving result for the persistence modules induced by the two filtrations (Theorem 2.3.6). ■

6.1.2 Connecting scales

We turn our attention to the complex $\mathcal{X}$. Let $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$ be the approximation complexes defined at any pair of consecutive scales $\alpha\in I$ and $\beta := (1+\varepsilon)\alpha \in I$, respectively. Let L_α and L_β be the scaled grid lattices at the respective scales. We define a map $\hat{g}_\alpha : L_\alpha \to L_\beta$ as: each grid point $v\in L_\alpha$ lies in the Voronoi cell of some point of L_β, and we set $\hat{g}_\alpha(v)$ as this lattice point. Without loss of generality, we assume that v does not lie on the boundary of any cell of $\mathcal{S}_\beta$, so that this assignment is unique. This map is well-defined by definition. We are interested in the image of this map when the domain is restricted to V_α, which are the vertices of $\mathcal{X}_\alpha$. To simplify notation, we write $\hat{g} := \hat{g}_\alpha$. We show that:

Lemma 6.1.3. $\hat{g}$ *maps vertices of* $\mathcal{X}_\alpha$ *to vertices of* $\mathcal{X}_\beta$.

Proof. Let v be a vertex of $\mathcal{X}_\alpha$. By definition, v is in distance at most α from some input point $p\in P$. Let $\hat{g}(v)$ denote the lattice point in L_β that v maps to. By definition, $\|v-\hat{g}(v)\|\leq\varepsilon\beta/6$. By triangle inequality,

$$\|p-\hat{g}(v)\| \leq \alpha + \varepsilon\beta/6 \leq (1+\varepsilon(1+\varepsilon)/6)\alpha < (1+\varepsilon)\alpha = \beta,$$

which implies that $\hat{g}(v)$ is indeed a vertex of $\mathcal{X}_\beta$. ■

Since $\hat{g}$ is well-behaved on vertices of $\mathcal{X}_\alpha$, we take its restriction $\hat{g}_{|V_\alpha} : V_\alpha \to V_\beta$, and we denote it by $\hat{g}$ as well. We show that $\hat{g}$ extends to a simplicial map $g:\mathcal{X}_\alpha\to\mathcal{X}_\beta$ on the simplices of $\mathcal{X}_\alpha$. Before we present the proof, we need the result:

Lemma 6.1.4. *The nerve of the Voronoi cells of the grid lattice is a flag complex.*

Proof. We show that if there is a set of $(k+1)$ cubes $\{A_0,\ldots,A_k\}$ (which are Voronoi cells of the grid lattice) such that they pairwise intersect, then all of them have a common intersection. The claim follows immediately.

Since all cubes are axis-aligned, each m-cube can be written as a product of d intervals $I_1\times\ldots\times I_d$, such that m (non-degenerate) intervals are of the form $[y_1,y_2]$ for some $y_1\neq y_2\in\mathbb{R}$, and $(d-m)$ (degenerate) intervals are of the form $[y_3,y_3]$ for some $y_3\in\mathbb{R}$.

Whenever two cubes intersect, they intersect in d intervals out of which some are (at least one is) degenerate. Consider any co-ordinate direction x. Along x, each pair of cubes A_i, A_j has an intersection interval.

Helly's theorem [Hel23] states that if there are $n > d$ convex sets in $\mathbb{R}^d$ such that the intersection of every $(d+1)$-subset is non-empty, then all the sets have a common intersection. We apply Helly's theorem in our context, for the 1-dimensional space along the co-ordinate direction x and the set of intersecting intervals along x. Since an interval is a convex object, it follows that all cubes have a common interval intersection along x, which proves the claim. ■

Using Lemma 6.1.4 on the appropriately scaled grid, it follows that $\mathcal{X}$ is a flag complex. Therefore, to show that the map g is simplicial on $\mathcal{X}_\alpha$, it is enough to argue that it is simplicial for edges.

Lemma 6.1.5. *Let c_1, c_2 be vertices of $\mathcal{X}_\alpha$ such that (c_1, c_2) is an edge in $\mathcal{X}_\alpha$. Then, either $g(c_1) = g(c_2)$, or there is an edge $(g(c_1), g(c_2))$ in $\mathcal{X}_\beta$.*

Proof. Let c_1 lie in a cube $M \in \mathcal{S}_\beta$. There are two cases:

- c_2 lies in M: in this case, $g(c_1) = g(c_2)$ so we are done.
- c_2 does not lie in M. We prove that c_2 lies in a cell adjacent to M, so that $(g(c_1), g(c_2))$ is an edge in $\mathcal{X}_\beta$. Assume for contradiction that it is not the case, so that c_1 and c_2 lie in non-adjacent cells of $\mathcal{S}_\beta$. In that case, $\|c_1 - c_2\|_\infty \geq \beta(\frac{\varepsilon}{3\sqrt{d}})$, which is the sidelength of the pixels at scale β. Since the grids at both scales are axis-aligned, it follows that $\|c_1 - c_2\|_\infty > \alpha(\frac{\varepsilon}{3\sqrt{d}})$. This implies that c_1 and c_2 are centers of non-adjacent cells of $\mathcal{S}_\alpha$, which is a contradiction.

The claim follows. ■

Hence, we have proven that

Lemma 6.1.6. *The map $g : \mathcal{X}_\alpha \to \mathcal{X}_\beta$ is a well-defined simplicial map.*

Moreover, since $\mathcal{X}$ is the same for all scales $\delta, \gamma \in [\alpha, \beta)$, there is an identity map between $\mathcal{X}_\delta$ and $\mathcal{X}_\gamma$. We set g to be identity in the range $[\alpha, \beta)$. This gives rise to the tower

$$(\mathcal{X}_\alpha)_{\alpha>0}.$$

6.1.3 Interleaving with the Čech filtration

We establish a connection between the approximation tower and the Čech filtration on P. The main step in our analysis is the establishment of a relationship between the towers $(\mathcal{S}_\alpha)_{\alpha>0}$ and $(\mathcal{X}_\alpha)_{\alpha>0}$. From the nerve theorem (Theorem 2.1.12), at each scale $\mathcal{S} \overset{h}{\simeq} \mathcal{X}$, so they have the same homology. We investigate the relationship when the two spaces are connected using inclusion maps (for $\mathcal{S}$), and g maps (for $\mathcal{X}$) to towers. We make use of the acyclic carrier theorem from [Wal81] for the analysis (see also Sub-section 2.4.1).

Let $\alpha \in I$ and $\beta := (1+\varepsilon)\alpha \in I$ be any two scales of I. Let $\{A_0, \ldots, A_N\}$ denote the pixels of $\mathcal{S}_\alpha$. Without loss of generality, we assume an order $\prec$ on these cells, $A_0 \prec \cdots \prec A_N$. Also, let $g(A_i) = B_i$ denote cells in $\mathcal{S}_\beta$ for all i, that is, the centre of A_i lies within the cube B_i (the B_is need not be all unique).

To make matters simple, we look the barycentric subdivision of the two spaces. Let $\bar{\mathcal{S}} := sd(\mathcal{S})$ and $\bar{\mathcal{X}} := sd(\mathcal{X})$ denote the abstract barycentric subdivisions of $\mathcal{S}$ and $\mathcal{X}$, respectively. We use the space and its subdivision interchangeably when clear from context. We now define two continuous maps which relate $\bar{\mathcal{S}}$ and $\bar{\mathcal{X}}$.

- $\psi : \bar{\mathcal{X}} \to \bar{\mathcal{S}}$: first, we define an acyclic carrier $CR : \bar{\mathcal{X}} \to \bar{\mathcal{S}}$. This guarantees the existence of the continuous map ψ, which is carried by CR [Wal81]. The precise form of ψ is irrelevant for us.

  Each simplex in $\mathcal{X}$ can be written in the form $(A_0 \ldots A_m)$, which corresponds to the intersection $A_0 \cap \ldots \cap A_m$ in the nerve and is geometrically realized on the vertices $\{A_0, \ldots, A_m\}$. To define CR, consider any simplex $\sigma \in \bar{\mathcal{X}}$; without loss of generality, let the flag representing σ be $\{(A_0A_1 \ldots A_m), \ldots, (A_0A_1 \ldots A_k)\}$ (in increasing order of dimension). The lowest dimensional simplex here is $(A_0 \ldots A_m)$. We set $CR(\sigma) = A_0 \cap \ldots \cap A_m$. It is easy to see that this is an acyclic carrier.

- $\phi : \bar{\mathcal{S}} \to \bar{\mathcal{X}}$: first, we define a function $\hat{\phi} : \mathcal{S} \to \mathcal{X}$ which assigns faces of cells of $\mathcal{S}$ to simplices of $\mathcal{X}$. Let $\Box$ be a face of $\mathcal{S}$ such that $\Box$ is a face of the d-cubes $\{A_{i_1}, \ldots, A_{i_m}\}$ of $\mathcal{S}$. We set $\hat{\phi}(\Box) = (A_{i_1} \ldots A_{i_m})$, which is a simplex of $\mathcal{X}$.

  Any simplex $\sigma \in \bar{\mathcal{S}}$ can be represented as a flag of faces of $\mathcal{S}$, which are of the form $\{\Box_0, \ldots, \Box_k\}$. We set $\phi(\sigma) := \{\hat{\phi}(\Box_k), \ldots, \hat{\phi}(\Box_0)\}$, which is a flag of simplices of $\mathcal{X}$, and hence lies in $\bar{\mathcal{X}}$.

In [Bjö03], it was shown that the maps ϕ and ψ induce isomorphisms of the homology groups $H(\bar{\mathcal{S}})$ and $H(\bar{\mathcal{X}})$ (in the language of partially ordered sets and isotone maps). Using the persistence equivalence theorem (Theorem 2.3.8), if the diagram

$$\begin{tikzcd}
\ldots \arrow[r] & H(\bar{\mathcal{X}}_\alpha) \arrow[r, "g^*"] \arrow[d, "\psi^*"] & H(\bar{\mathcal{X}}_\beta) \arrow[r] \arrow[d, "\psi^*"] & \ldots \\
\ldots \arrow[r] & H(\bar{\mathcal{S}}_\alpha) \arrow[r, "inc^*"] & H(\bar{\mathcal{S}}_\beta) \arrow[r] & \ldots
\end{tikzcd} \tag{6.2}$$

commutes, where *inc* is the inclusion map, then the two persistence modules are isomorphic (the asterisks denote the corresponding induced linear maps). In the diagram, we have ignored scales other than those in I, since the diagram captures those cases implicitly.

Lemma 6.1.7. *Diagram* (6.2) *commutes, that is,*

$$inc^* \circ \psi^* = \psi^* \circ g^*.$$

Proof. To prove the claim, we look at the corresponding diagram of spaces:

$$\begin{tikzcd}
\ldots \arrow[r] & \bar{\mathcal{X}}_\alpha \arrow[r, "g"] \arrow[d, "\psi"] & \bar{\mathcal{X}}_\beta \arrow[r] \arrow[d, "\psi"] & \ldots \\
\ldots \arrow[r] & \bar{\mathcal{S}}_\alpha \arrow[r, "inc"] & \bar{\mathcal{S}}_\beta \arrow[r] & \ldots
\end{tikzcd}$$

We define an acyclic carrier $D : \bar{\mathcal{X}}_\alpha \to \bar{\mathcal{S}}_\beta$, which carries both $inc \circ \psi$ and $\psi \circ g$. Then, the acyclic carrier theorem shows that $inc \circ \psi \overset{h}{\simeq} \psi \circ g$, which implies our claim.

Let σ be any simplex of $\bar{\mathcal{X}}_\alpha$; suppose that the flag of simplices of $\mathcal{X}$ representing σ is: $\{(A_0A_1 \ldots A_m), \ldots, (A_0A_1 \ldots A_k)\}$, in ascending order of dimension. Under the order $\prec$, the lowest ordered cube incident to σ is A_0 and under the map g, A_0 maps to $B_0 = g(A_0)$. We set $D(\sigma)$ as the union of the pixels in the neighborhood of B_0, $D(\sigma) := \mathrm{NBR}(B_0) \cap \mathcal{S}_\beta$.

$D(\sigma)$ is acyclic because its nerve is star-shaped. It is straightforward to see that for any simplex $\tau \subseteq \sigma$, $D(\tau) \subseteq D(\sigma)$ holds, so D is an acyclic carrier.

By definition, $\psi(\sigma) = A_0 \cap \ldots \cap A_m$, which is a subset of A_0. Using a simple triangle inequality, we see that A_0 lies within $D(\sigma)$. It follows that $inc \circ \psi$ is carried by D.

We now argue about $\psi \circ g$. From the definition of g, we see that $g(\gamma)$ is the flag of simplices $\{(B_0 B_1 \ldots B_m), \ldots, (B_0 B_1 \ldots B_k)\}$. Then, $\psi(g(\sigma)) = B_0 \cap \ldots \cap B_m \subset D(\sigma)$, so D carries $\psi \circ g$. ■

We thus see that the two persistence modules of $\bar{\mathcal{S}}$ and $\bar{\mathcal{X}}$ are isomorphic. Since $H(\mathcal{S}) \simeq H(\bar{\mathcal{S}})$ and $H(\mathcal{X}) \simeq H(\bar{\mathcal{X}})$, it follows that the persistence modules $(H(\mathcal{X}_\alpha))_{\alpha>0}$ and $(H(\mathcal{S}_\alpha))_{\alpha>0}$ are isomorphic. Applying Theorem 6.1.2, the transitivity of interleavings, and the fact that $(H(\mathcal{B}_\alpha))_{\alpha\geq 0}$ and $(H(\mathcal{C}_\alpha))_{\alpha\geq 0}$ are isomorphic, we conclude that

Theorem 6.1.8. *The persistence module $(H(\mathcal{X}_\alpha))_{\alpha>0}$ and the Čech persistence module $(H(\mathcal{C}_\alpha))_{\alpha\geq 0}$ are $(1+\delta)$-approximations of each other.*

6.1.4 Computation

At scales below $CP(P)/2$, the balls centered at the input points do not intersect. Also, at scales at and beyond $diam(P)$, all balls intersect, leading to an acyclic complex. Therefore, we construct the approximation complexes only for the range of scales in $[CP(P)/2, diam(P)]$. Since consecutive scales jump by a factor of $(1+\varepsilon)$, the number of scales to be considered is $\lceil \log_{(1+\varepsilon)} \frac{diam(P)}{CP(P)/2} \rceil = \lceil \log_{(1+\varepsilon)} 2\Delta \rceil$.

Theorem 6.1.9. *The k-skeleton of the approximation tower has size*

$$n \left(\frac{1}{\varepsilon}\right)^d 2^{O(d \log d + dk)} \lceil \log_{(1+\varepsilon)} 2\Delta \rceil.$$

Proof. At any scale $\beta > 0$, the sidelength of any cube is $\varepsilon\beta/(3\sqrt{d})$, so a ball of radius half this value is contained inside the cube. Using a simple packing argument, we see that each ball in $\mathcal{B}_\beta$ is covered by no more than $(\frac{2\beta(1+\varepsilon)}{\varepsilon\beta/(3\sqrt{d})})^d = (12\sqrt{d}/\varepsilon)^d$ cubes. There are n balls of $\mathcal{B}_\beta$. Hence, in total there are no more than $n(12\sqrt{d}/\varepsilon)^d$ pixels. Each such cube has $(3^d - 1)$ neighbors, which implies that the number of k-simplices incident to a cube is $2^{O(dk)}$. In total, the k-skeleton of $\mathcal{X}$ has size

$$n(12\sqrt{d}/\varepsilon)^d 2^{O(dk)} = n(1/\varepsilon)^d 2^{O(d \log d + dk)}.$$

Since there are $\lceil \log_{(1+\varepsilon)} 2\Delta \rceil$ scales of the tower, the bound follows by multiplying the size at each scale with the number of scales. ■

Let $P = \{p_1, \ldots, p_n\}$ and let β be a given scale of the tower. The scales of the tower can be computed using the techniques mentioned in Chapter 4. To construct the complex at scale β, we use point location in L_β, to find the closest grid points to P. Let $\{x_1, \ldots, x_n\}$ denote the cubes whose centers are closest to $\{p_1, \ldots, p_n\}$ respectively. For each x_i, we inspect the set of cubes at L_∞-distance at most $(1+\varepsilon)\beta$ and check whether the inspected cube lies within $B(p_i, \beta)$: if true, we add a digital vertex to $\mathcal{X}_\beta$. Since the inspected set of cubes is a superset of $B(p_i, \beta)$, each digital vertex is correctly identified. To construct the edges of $\mathcal{X}_\beta$, we simply go over the $3^d - 1 = 2^{O(d)}$ neighbors of each digital vertex and add edges between adjacent vertices. Having the 1-skeleton of $\mathcal{X}_\beta$, we complete the k-skeleton using the technique used in Algorithm A from Chapter 4. Constructing the simplicial map g is straightforward, by point location for V_β in the grid $L_{(1+\epsilon)\beta}$. The cost of the remaining operations is dominated by the upper bound on the cost of constructing the k-skeleton. Accounting for the fact that the complex has to be constructed independently for each scale, we get the result:

Theorem 6.1.10. *The k-skeleton of the approximation tower can be computed in time*

$$n\left(\frac{1}{\varepsilon}\right)^d 2^{O(d\log d+dk)}\lceil\log_{(1+\varepsilon)}2\Delta\rceil.$$

6.2 Discussion

In this chapter, we presented a scheme to approximate the Čech filtration with an approximation quality of $(1+\delta)$. The approximation quality is better than the schemes of Chapter 4 and Chapter 5, but at the cost of significantly worse complexity. At the same time, this is an improvement over the previously known $(1+\delta)$-approximation schemes, since the exponent of δ in the complexity bound is de-coupled from the dimension of the skeleton.

Our method can be interpreted as adding Steiner points to approximate the union of balls. This digitization accounts for a factor of $2^{O(d\log d+d\log(1/\varepsilon))}$ in the size of the complex. An important question is whether it is possible to compress the pixels to get an approximation complex on the input points, so that this factor is mitigated. For instance, a natural idea is to associate each pixel to its nearest input point. More formally, we define a coloring $\mathcal{D}:\mathcal{S}\to P$ as: for a cube $\square\in\mathcal{S}$, $\mathcal{D}(\square)=p_i$ if p_i is the closest point in P to $\square$. This gives a partition of the pixels of $\mathcal{S}$ into n colors. With this coloring, a natural choice for constructing a simplicial complex on the input points can be: $\mathcal{D}$ is a vertex map from $\mathcal{X}$ to P. We can interpret $\mathcal{D}$ as a map on the simplices of $\mathcal{X}$. The image of $\mathcal{D}$ can be considered as a candidate for approximation. A common issue with this approach is that these colored regions may not be connected, that is, $\mathcal{D}^{-1}(p_j)$ may not be a connected set (for an example in 2D, see [CET15]). Such issues can also arise for any k-wise intersection of the partitions. In general, the partition may not be a good cover of space, and that makes it unclear whether such an approach can guarantee the desired topological properties.

Another question is whether such complexes can be constructed without going through the digitization step. This would make the algorithm computationally attractive.

Digitizing with Permutahedra

We mention another line of thought, that has the potential to bring down the cost of the approximation significantly. In our digitizing scheme, we can replace the cubical pixels by pixels coming from any other lattice. In particular, we can use the A^* lattice for this purpose: for a fixed scale α, we scale the Voronoi vectors of this lattice by $\frac{\varepsilon\alpha}{3\sqrt{d}}$, to get permutahedra whose diameters are less than $\varepsilon\alpha/3$. The pixels are then defined as the permutahedral cells whose centers lie in the union of α-balls. The approximation complex is the nerve of the pixels.

Most of the properties of the cubical pixels carry over to the permutahedral pixels. For instance, Lemma 6.1.1, Theorem 6.1.2 and Lemma 6.1.3 hold for pixelization by any lattice. A counterpart of Lemma 6.1.4 for permutahedra was proved in Chapter 3 (see Lemma 3.1.19). Unfortunately, we do not yet have a counterpart of Lemma 6.1.5 for permutahedra: we need the counterpart for permutahedra to show that the (corresponding) map g is simplicial. In a more general setting, we claim that

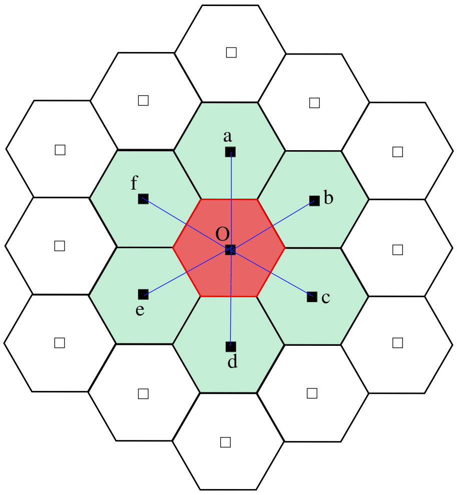


Figure 6.2: O represents the origin and the red (darkly shaded) hexagon is the Voronoi polytope of the lattice. The points a, b, c, d, e, f are the neighbors of O. The green (lightly shaded) hexagons intersect the Voronoi polytope. The vectors $Oa, \dots, Of$ are the Voronoi vectors. The Minkowski sum of π with the six corresponding line segments is the shaded region. This shows that any line segment parallel to (and shorter than) one of these vectors and with an endpoint inside π must lie in the shaded region (This figure was also used in Chapter 3 in a different context).

Claim 6.2.1. *Let π be the permutahedron centered at the origin and $\{v_1, \dots, v_m\}$ be the collection of Voronoi vectors of the origin. Consider a line segment ℓ parallel to any given Voronoi vector v_i with an endpoint in the interior of π. If $\|\ell\| < \|v_i\|$, then ℓ is completely contained in the union of π and the permutahedra adjacent to π.*

We illustrate the claim for $d = 2$ in Figure 6.2. The claim is also easy to verify for $d = 3$. Surprisingly, we could not find a way to prove the claim for $d > 3$. Also, we are unsure whether this claim holds for any lattice in general. The corresponding claim for cubes is easy to prove, since the union of a cube and its adjacent cells is convex. However, this convenient property does not hold for permutahedra. We did some experiments to test the claim and had positive observations for the permutahedral lattice. We hope to find a proof in the future. For now, assuming that the claim is true, we get a tower $(\mathcal{K}_\alpha)_{\alpha>0}$ constructed on permutahedral pixels, connected with the map g.

With some minor modifications in the interleaving analysis, we arrive at the result:

Theorem 6.2.2. *Under the assumption that Claim 6.2.1 is true, the persistence module $(H_*(\mathcal{K}_\alpha))_{\alpha>0}$ is a $(1+\delta)$-approximation of the Čech persistence module $(H_*(\mathcal{C}_\alpha))_{\alpha\geq 0}$.*

Moreover, at any given scale, the k-skeleton of $\mathcal{X}$ has size $n(\frac{1}{\varepsilon})^d 2^{O(d \log d)}$. The computation time for the k-skeleton has the same bound. Here, $(1+\delta) = (1+\varepsilon)^2$ as before.

Since the A^* lattice is in general position, the approximation complex has significantly smaller size than the cubical case.

Part II

Techniques for Doubling Spaces

Chapter 7

Well-Separated Simplicial Decomposition

In chapters 4 to 6, we investigated approximation techniques for filtrations built on arbitrary point clouds in Euclidean space. For practical data sets, it is often the case that inputs come from spaces of low intrinsic dimension, such as the case of point samples of manifolds embedded in high dimensional Euclidean space. In such cases, the aforementioned techniques do not correctly exploit the structure of the point cloud and therefore, do not give attractive theoretical guarantees. In this chapter and the next, we ameliorate this issue by presenting a few techniques which make use of these structural properties. We use the notion of doubling dimension (Definition 3.3.2) to capture the intrinsic dimension of the input.

In the current chapter, we present a generalization of the well-separated pair decomposition from Chapter 3 to higher dimensions. We use the concept of net-trees from Chapter 3 for our construction. Using this concept, we present a simplicial tower which $(1+\varepsilon)$-approximates the Čech filtration and whose complexity primarily depends on the doubling dimension of the underlying point cloud. The results presented in this chapter are a generalization of [KS13] to doubling spaces.

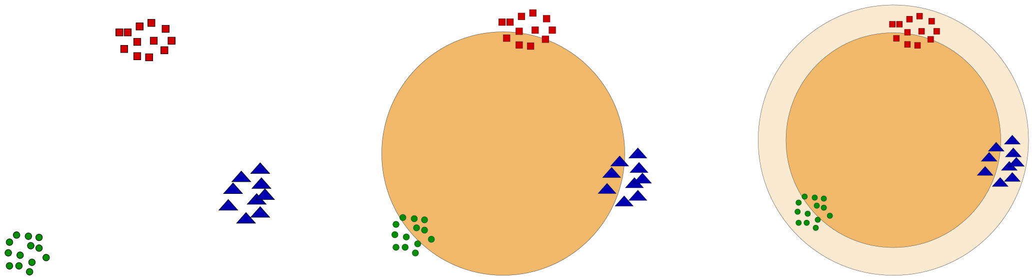
7.1 Well-separated simplicial decomposition

First, we look at some notation. Let $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ be a set of n points with doubling dimension Υ and doubling constant λ (Definition 3.3.2). Let $\mathcal{T}$ denote a net-tree on P (Definition 3.3.5). Let $\{v_0, v_1, \ldots\}$ denote the nodes of $\mathcal{T}$. Let P_{v_i} denote the set of input points covered by v_i, and $\mathrm{rep}_{v_i} \in P_{v_i}$ denote the representative of v_i, for all i.

Definition 7.1.1 (minimum enclosing ball for tuples). *Let $\gamma = (v_0, \ldots, v_k)$ be a tuple of net-tree nodes. We say that the* minimum enclosing ball of the tuple γ*, denoted by* $\mathrm{meb}(\gamma)$*, is the smallest ball enclosing the point set* $\{\mathrm{rep}_{v_0}, \ldots, \mathrm{rep}_{v_k}\}$*. We denote the radius of* $\mathrm{meb}(\gamma)$ *by* $\mathrm{rad}(\gamma)$*.*

For any $e > 0$ and an Euclidean ball $\mathbb{B}$ of radius $r > 0$, we denote by $e\mathbb{B}$ the ball of radius er concentric to $\mathbb{B}$. We state the following property, which is a simple consequence of the triangle inequality, and is used several times in our arguments:

Observation 7.1.2. *Let $\mathbb{B}$ be a ball of radius r such that it has a non-empty intersection with some point set $M \subset \mathbb{R}^d$. If the diameter of M is at most er for some $e > 0$, then $M \subseteq (1+e)\mathbb{B}$.*



(a) A 3-tuple of nodes; each node of the tuple is represented by points of different colors (and shapes).

(b) A ball which contains at least one point of each node.

(c) An ε-inflation of the ball covers all points.

Figure 7.1: A well-separated tuple.

In this chapter, whenever we refer to the quantity ϵ, we always assume that it lies in the interval $(0, 1)$. Consider any ε-WSPD W on $\mathcal{T}$ (Definition 3.3.7). Recall that the elements of W are pairs of nodes $(u, v) \in \mathcal{T}$ such that they are *ε-well-separated*, which means that they satisfy $max\{\mathrm{diam}(u), \mathrm{diam}(v)\} \leq \varepsilon \mathrm{dist}(u, v)$, where $\mathrm{dist}(u, v)$ denotes the minimal distance between P_u and P_v. Informally speaking, all pairs of points (p, q) with $p \in P_u$, $q \in P_v$ have a similar distance to each other. Each pair of nodes in $\mathcal{T} \times \mathcal{T}$ is covered by some pair of W. Intuitively, W is a 1-dimensional structure, since it covers the set of pairs of nodes.

We state a simple consequence of well-separation which may appear as somewhat unrelated at first, but is useful in generalizing WSPDs to longer tuples:

Lemma 7.1.3. *Let $(q, q') \in W$ be any pair. For any ball $\mathbb{B}$ that contains at least one point each of P_q and $P_{q'}$, it holds that*

$$P_q \cup P_{q'} \in (1 + 2\varepsilon)\mathbb{B}.$$

Proof. Let $\mathbb{B}$ be a ball with radius r which intersects both q and q', which means that $\mathrm{dist}(q, q') \leq 2r$. Because (q, q') is well-separated,

$$\mathrm{diam}(q) \leq \varepsilon \mathrm{dist}(q, q') \leq 2\varepsilon r,$$

which implies that $(1+2\varepsilon)\mathbb{B}$ contains all of P_q, from Observation 7.1.2. The same argument applies for q'. ■

Definition 7.1.4 (well-separated tuple). *A (unordered) $(k + 1)$-tuple $(v_0, \ldots, v_k)$ of net-tree nodes of $\mathcal{T}$ is called an ε-*well separated tuple* (ε-WST), if for any ball $\mathbb{B}$ which contains at least one point of each P_{v_i}, it holds that*

$$P_{v_0} \cup \ldots \cup P_{v_k} \in (1 + \varepsilon)\mathbb{B}.$$

See Figure 7.1 for an intuitive example. We say that a WST $(v_0, \ldots, v_k)$ *covers* a k-simplex $\sigma = (p_0, \ldots, p_k)$ if there is a permutation π of $(0, \ldots, k)$ such that $p_{\pi(\ell)} \in P_{v_\ell}$ for all $0 \leq \ell \leq k$.

Definition 7.1.5 (well-separated simplicial decomposition). *For $k \geq 1$, we define an ε-*Well-Separated Simplicial Decomposition* ((ε, k)-WSSD) as a set of $(k + 1)$-tuples $\Gamma_k = \{\gamma_1, \ldots, \gamma_l\}$ such that*

- *each $\gamma_i \in \Gamma_k$ is an ε-WST, and*
- *each k-simplex on P is covered by some $\gamma_i \in \Gamma$.*

An ε-WSSD *is the union of (ε, k)-WSSDs for all $1 \leq k \leq \Upsilon$.*

It is straightforward to see with Lemma 7.1.3 that an $\frac{\varepsilon}{2}$-WSPD is an $(\varepsilon, 1)$-WSSD. On an intuitive level, an (ε, k)-WSSD is a k-dimensional structure since it captures precisely the set of k-simplices, and hence can be seen as a higher dimensional analogue of WSPDs.

7.1.1 Construction

We describe an algorithm to construct an ε-WSSD for P, using the net-tree $\mathcal{T}$. The algorithm proceeds inductively in dimension. For $k = 1$, we simply compute an $\frac{\varepsilon}{2}$-WSPD using the WSPD algorithm from [HPM06].

For any $k > 1$, we construct an (ε, k)-WSSD Γ_k by extending each tuple of the previously constructed $(\varepsilon, k-1)$-WSSD Γ_{k-1}. Consider any tuple $\gamma = (v_0, \dots, v_{k-1}) \in \Gamma_{k-1}$. Recall that $\mathrm{meb}(\gamma)$ is the minimum enclosing ball of the representative points of the nodes. We compute an ε-approximation of $\mathrm{rad}(\gamma)$ using an algorithm of [BC03]; more precisely, we compute a value r such that there is a ball of radius r covering all representatives and

$$\mathrm{rad}(\gamma) \leq r \leq (1+\varepsilon)\mathrm{rad}(\gamma) \tag{7.1}$$

Recall from Definition 3.3.5 that any node $u \in T$ is associated with a *level* $\ell(u)$ and a *scale* scale_u, which are related as $\mathrm{scale}_u = \frac{2\tau^{\ell(u)+1}}{\tau - 1}$. The diameter of P_u satisfies $\mathrm{diam}(u) \leq 2\mathrm{scale}_u$. We find the lowest ancestor u of v_0 in $\mathcal{T}$ that satisfies

$$\frac{\tau - 5}{4\tau^2}\mathrm{scale}_u \geq 8r. \tag{7.2}$$

To find suitable nodes to be added to γ, we inspect all the nodes $w \in \{u \cup \mathrm{Rel}(u)\}$. For each w, we traverse its sub-tree to find the highest descendants whose diameters are small enough for them to be valid candidates. Specifically, we find w's highest descendants x satisfying

$$\frac{\varepsilon r}{4} \geq 2\mathrm{scale}_x. \tag{7.3}$$

For each such node x, we add a $(k+1)$-tuple $(v_0, \dots, v_{k-1}, x)$ to Γ_k. A simple example is shown in Figure 7.2.

7.1.2 Correctness

In order to prove the correctness of our construction procedure, we need to show that the generated tuples indeed form a (ε, k)-WSSD. This requires showing that each generated tuple is an ε-WST, and that all k-simplices on P are covered by the set of $(k+1)$-tuples.

Lemma 7.1.6. *Each tuple added by the algorithm is an ε-WST.*

Proof. We prove the claim by induction: for $k = 1$, the statement is true because an $(\varepsilon$,1$)$-WSSD is an $\frac{\varepsilon}{2}$-WSPD, each tuple of which is ε-well-separated.

For $k > 1$, suppose that all $(k-1)$-tuples are ε-well-separated. Consider a $(k-1)$-tuple $\gamma = (v_0, \dots, v_{k-1})$. Suppose that the algorithm appends x to this tuple to form the k-tuple $\gamma' = (v_0, \dots, v_{k-1}, x)$. Now, consider any ball $\mathbb{B}$ that contains a point of each of P_{v_i} and a point of P_x. By induction hypothesis, $(1+\varepsilon)\mathbb{B}$ contains all of P_{v_i}. To show that γ' is ε-well-separated, it suffices to show that $(1+\varepsilon)\mathbb{B}$ also contains all of P_x.

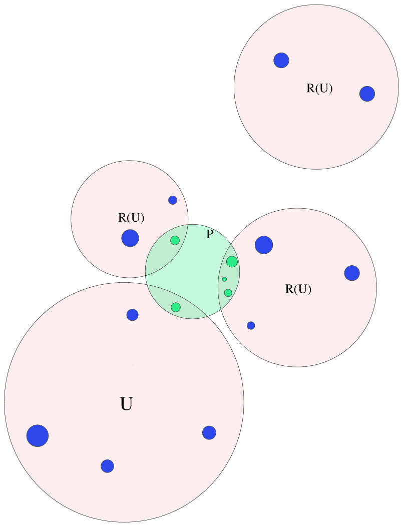


Figure 7.2: An example of constructing tuples of Γ_5 from a tuple of Γ_4: the green (small shaded) disks inside P are the nodes of a tuple $\gamma \in \Gamma_4$. Region U along with $\mathrm{Rel}(U)$ (depicted as $R(U)$) in the figure is examined for appending nodes to γ. In this example, the nodes represented by blue disks (small disks outside P) are individually appended to γ to form tuples of Γ_5.

Let r' denote the radius of $\mathbb{B}$. By Observation 7.1.2, it suffices to show that $\mathrm{diam}(x) \leq \varepsilon r'$. Using induction hypothesis again, the ball $(1+\varepsilon)\mathbb{B}$ contains all representatives rep_{v_i}. That means, $(1+\varepsilon)\mathbb{B}$ is an enclosing ball of the representatives and henceforth, $(1+\varepsilon)r' \geq \mathrm{rad}(\gamma)$. Recall that we approximate $\mathrm{rad}(\gamma)$ by r, that satisfies $r \leq (1+\varepsilon)\mathrm{rad}(\gamma)$, so $r \leq (1+\varepsilon)^2 r' \leq 4r'$ for $\varepsilon \leq 1$. From the algorithm, x is chosen such that

$$\mathrm{diam}_x \leq 2\mathrm{scale}_x \leq \frac{\varepsilon r}{4} \leq \varepsilon r',$$

which proves the claim. ■

For showing that all k-simplices on P are covered by the generated $(k+1)$-tuples, we need several preparatory results. The first result is taken from [BC08]: while our claim also follows as a simple corollary of the main result of [BC08], we give a more detailed argument for clarity.

Lemma 7.1.7. *Let $M \subset \mathbb{R}^d$ be a point set with $|M| \geq 3$. Then, there exists some point $p \in M$ such that*

$$p \in \left(\frac{\sqrt{1+1/d}}{\sqrt{1-1/d}}\right) \mathrm{meb}(M \setminus \{p\}).$$

In particular, $p \in 2\mathrm{meb}(M \setminus \{p\})$ for $d \geq 2$.

Proof. The claim is trivial if there exists a point $p \in M$ whose removal does not change the minimum enclosing ball. Therefore, assume without loss of generality that $|M| \leq d+1$, and that all points of M are on the boundary of $\mathrm{meb}(M)$. Let c be the center of $\mathrm{meb}(M)$ and $r := \mathrm{rad}(M)$.

Let Π denote the convex hull of M. Take the largest ball $\mathbb{B}$ centered at c that is contained in Π; y [BC08, Lemma 3.2], its radius is at most r/d. Moreover, $\mathbb{B}$ touches at least one facet of Π. Let p be the point of M opposite to this facet, set $M' := M \setminus \{p\}$ and let c' and r' denote the center and radius of meb(M'). Following the arguments of [BC08, Lemma 3.3], it holds that

$$r' \geq r\sqrt{1 - 1/d^2}$$

and moreover, c' is the point where $\mathbb{B}$ touches the facet, so that $\|c - c'\| \leq r/d$. By the triangle inequality,

$$\begin{aligned}\|p - c'\| &\leq \|p - c\| + \|c - c'\| \leq r + r/d \\ &\leq (1 + 1/d)\frac{r'}{\sqrt{1 - 1/d^2}} \leq \left(\frac{\sqrt{1+1/d}}{\sqrt{1-1/d}}\right)\mathrm{rad}(M'),\end{aligned}$$

which implies the first part of the claim.

The second part follows simply by noting that $\left(\frac{\sqrt{1+1/d}}{\sqrt{1-1/d}}\right) \leq 2$ for all $d \geq 2$. ■

Lemma 7.1.8. *Let u be any internal node of the net-tree $\mathcal{T}$. Then*

$$\mathrm{diam}(u) \geq \frac{\tau - 5}{4\tau^2}\mathrm{scale}_u$$

Proof. Since u is internal, it has at least two distinct children v_1, v_2. Let $p = \mathrm{rep}_{v_1}$. By the packing property of the net-tree (Definition 3.3.5), the ball centered at p with radius $\frac{\tau-5}{4\tau^2}\mathrm{scale}_{\mathrm{par}(v_1)}$ only contain points of P_{v_1}. Since $\mathrm{par}(v_1) = u$ and v_2 contains at least one point not in P_{v_1}, the statement follows. ■

Lemma 7.1.9. *Let u be any node of the net-tree $\mathcal{T}$ with* $\mathrm{diam}(u) \geq 4R$*, for some given $R > 0$. Then, for any ball $\mathbb{B}$ of radius R that contains at least one point of P_u, the set $\{u \cup \mathrm{Rel}(u)\}$ covers the set of points in $P \cap 2\mathbb{B}$.*

Proof. Consider any point $x \in P \cap 2\mathbb{B}$. The net-tree $\mathcal{T}$ contains a path from the root to a leaf such that $x \in P_w$ for all nodes w on that path. In particular, there exists a node v with $x \in P_v$ and $\ell(v) \leq \ell(u) < \ell(\mathrm{par}(v))$. It can be observed with the help of Figure 7.3 that

$$\begin{aligned}\|\mathrm{rep}_u - \mathrm{rep}_v\| &\leq \mathrm{diam}(u) + 3R + \mathrm{diam}(v) < 3\mathrm{diam}(u) \\ &\leq 6\mathrm{scale}_u \leq \frac{12\tau}{\tau - 1}\tau^{\ell(u)} < 14\tau^{\ell(u)}.\end{aligned}$$

From the definition of the Rel set, it follows that $v \in \mathrm{Rel}(u)$. ■

Lemma 7.1.10. *Γ_k covers all k-simplices over P.*

Proof. We prove the claim by induction over the dimension of the tuple. Base case: for $k = 1$, the 1-simplices are pairs of points from P which are covered by the WSPD, by definition.

Induction case: suppose that Γ_{k-1} cover all $(k-1)$-simplices. Consider any k-simplex $\sigma = (p_0, \ldots, p_k)$. From Lemma 7.1.7, it follows that there exists a point, say p_k, which lies in $2\mathrm{meb}(\sigma')$, where $\sigma' := \sigma \setminus \{p_k\}$. σ' is a $(k-1)$-simplex and hence is covered by some k-tuple $\gamma = (v_0, \ldots, v_{k-1}) \in \Gamma_{k-1}$. We show that when processing γ, the algorithm produces a $(k+1)$-tuple (γ, x) with $p_k \in P_x$, implying that σ is covered by that tuple. This completes the proof.

Let $r' := \mathrm{rad}(\sigma')$, and let r denote the approximate radius for meb(γ) as computed in the algorithm. Let $\mathbb{B}$ denote the corresponding enclosing ball of radius r, which contains the

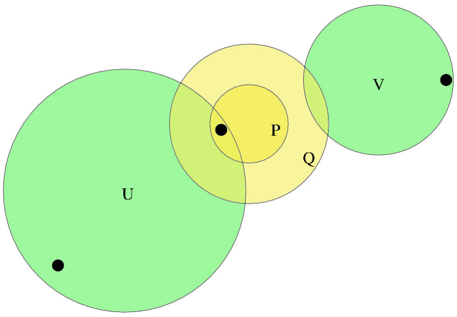


Figure 7.3: P represents a ball $\mathbb{B}$ of radius R and Q denotes $2\mathbb{B}$. Disk U denotes a node such that $diam(u) \geq 4R$ and it intersects $\mathbb{B}$. Disk U represents a node covering a point in $2\mathbb{B}$. The dark disks in U and V are their representatives. A simple triangle inequality gives a bound on the distance $\|\text{rep}_U - \text{rep}_V\|$.

representatives of γ. Since γ is an ε-WST, $(1+\varepsilon)\mathbb{B}$ contains $\{p_0, \dots, p_{k-1}\}$. Therefore, we have that $r' \leq (1+\varepsilon)r \leq 2r$. Because of Lemma 7.1.8 and Equation 7.2, the net-tree node u (ancestor of v_0) computed in the algorithm satisfies $\text{diam}(u) \geq \frac{\tau-5}{4\tau^2}\text{scale}_u \geq 8r \geq 4r'$. Applying Lemma 7.1.9 on u, r', and $\mathbb{B}$ yields that the set $\{u \cup \text{Rel}(u)\}$ covers $P \cap 2\mathbb{B}$. Thus, there is some node $w \in \{u \cup \text{Rel}(u)\}$ that covers p_k. By construction, one of the descendants x of w for which a tuple (γ, x) is created satisfies $p_k \in P_x$. ■

With Lemma 7.1.6 and Lemma 7.1.10, it follows that the constructed set Γ_k is an (ε, k)-WSSD, for all $1 \leq k \leq \Upsilon$.

7.1.3 Properties

We bound the size of the (ε, k)-WSSD generated by our algorithm and the total time taken to compute it.

Lemma 7.1.11. *The size of (ε, k)-WSSD Γ_k computed by the algorithm is*

$$n(2/\varepsilon)^{O(\Upsilon k)}.$$

Proof. We proceed by induction on k to prove our claim. For $k = 1$, size of a $\frac{\varepsilon}{2}$-WSPD is $n(2/\varepsilon)^{O(\Upsilon)}$ which conforms with our claim.

Induction case: assume that the size of Γ_{k-1} is $n(2/\varepsilon)^{O(\Upsilon(k-1))}$. Consider any tuple $\gamma = (v_0, \dots, v_{k-1}) \in \Gamma_{k-1}$. Let r be the approximate radius of $\text{meb}(\gamma)$ as computed in the algorithm. We bound the number of nodes explored by the algorithm for extending γ.

In the course of the algorithm, we find the lowest ancestor u of v_0 which satisfies Equation (7.2) and then explore the highest descendants of $\{u \cup \text{Rel}(u)\}$ until we reach nodes x satisfying Equation (7.3). Let us denote by $\tau' := \frac{\tau-5}{4\tau^2}$. The child u' of u that is an ancestor of v_0 satisfies $\text{scale}_{u'}\tau' < 8r$ because the algorithm would not have chosen u otherwise. By a similar argument the parent x' of x satisfies $2\text{scale}_{x'} > \frac{\varepsilon r}{4}$. It follows that

$$\frac{\text{scale}_{u'}}{\text{scale}_{x'}} < \frac{\frac{8r}{\tau'}}{\frac{\varepsilon r}{8}} < \frac{64}{\varepsilon\tau'}.$$

Because $\frac{\text{scale}_{\text{par}(v)}}{\text{scale}_v} \geq \tau$ for any node $v \in \mathcal{T}$, u and x are at most $\ell := 2 + \lceil \log_\tau \frac{64}{\varepsilon\tau'} \rceil$ levels apart in the net-tree. Since any node has at most $\lambda^{O(1)}$ children, it immediately follows that the total number of nodes explored is $\lambda^{O(\ell)}$ which simplifies to $(2/\varepsilon)^{O(\Upsilon)}$, since $\lambda = 2^{O(\Upsilon)}$. Because we add at most one tuple for each explored node, the bound follows. ■

Corollary 7.1.12. *The total size of an ε-WSSD is upper bounded by*

$$\sum_{k=1}^{\Upsilon} n(2/\varepsilon)^{O(\Upsilon k)} = n(2/\varepsilon)^{O(\Upsilon^2)}.$$

Lemma 7.1.13. *Computing an (ε, k)-WSSD takes*

$$2^{O(\Upsilon)} dn \log n + nd(2/\varepsilon)^{O(\Upsilon k)}$$

time.

Proof. Again, the proof follows an inductive argument on k. For $k = 1$, we compute Γ_1, which is an $\frac{\varepsilon}{2}$-WSPD. This requires

$$d\left(2^{O(\Upsilon)} n \log n + n(2/\varepsilon)^{O(\Upsilon)}\right)$$

time in expectation to construct [HPM06, Section 5]; the multiplicative factor of d comes from distance computations in $\mathbb{R}^d$ which requires $O(d)$ time.

For $k \geq 2$, consider the construction of Γ_k from Γ_{k-1}. For each tuple $\gamma \in \Gamma_{k-1}$, we compute the approximate minimum enclosing ball $\mathbb{B}$ and then explore ancestors and the descendants of their Rel sets to find suitable nodes to be added.

Computing the approximate minimum enclosing ball of the representatives takes $O(kd/\varepsilon + \varepsilon^{-5})$ time [BC03]. From the arguments of Lemma 7.1.11, the number of nodes explored is $(2/\varepsilon)^{O(\Upsilon)}$, and only a constant amount of time is spent per node. Hence, the total time spent for each $\gamma \in \Gamma_{k-1}$ is:

$$O(kd/\varepsilon + \varepsilon^{-5}) + (2/\varepsilon)^{O(\Upsilon)}.$$

As the size of Γ_{k-1} is $n(2/\varepsilon)^{O(\Upsilon(k-1))}$, the additional time required to compute Γ_k from Γ_{k-1} is

$$n(2/\varepsilon)^{O(\Upsilon(k-1))}\left(O(kd/\varepsilon + \varepsilon^{-5}) + (2/\varepsilon)^{O(\Upsilon)}\right) = n(2/\varepsilon)^{O(\Upsilon k)} d.$$

The total time required to compute Γ_1,…,Γ_k is

$$d\left(2^{O(\Upsilon)} n \log n + n(2/\varepsilon)^{O(\Upsilon)}\right) + \sum_{i=2}^{k} n(2/\varepsilon)^{O(\Upsilon i)} d = d\left(2^{O(\Upsilon)} n \log n + n(2/\varepsilon)^{O(\Upsilon k)}\right).$$

■

We conclude the section with a property of our computed WSTs which will be useful in the next section.

Lemma 7.1.14. *For any ε-WST t=$(v_0, \ldots, v_k)$ generated by our algorithm, it holds that* $\mathrm{scale}_{v_i} \leq \varepsilon \mathrm{rad}(t)$ *for all v_i.*

Proof. Again, the proof is by induction: for $k = 1$, we construct a $\frac{\varepsilon}{2}$-WSPD using the algorithm of [HPM06, Section 5]. For any pair (a, b) in the generated WSPD, their construction ensures that $max(\mathrm{scale}_a, \mathrm{scale}_b) \leq \frac{\varepsilon}{2}\mathrm{dist}(a, b)$. Since $\mathrm{dist}(a, b) \leq 2\mathrm{rad}(a, b)$, it follows that $max(\mathrm{scale}_a, \mathrm{scale}_b) \leq \varepsilon \mathrm{rad}(a, b)$.

For $k \geq 2$, assume that the statement is true for each tuple of Γ_{k-1}. Consider any tuple $\gamma = (v_0, \ldots, v_{k-1}) \in \Gamma_{k-1}$ and let r denote the radius of the ε-approximation of $\mathrm{meb}(\gamma)$ as computed by our algorithm. A node x is appended to γ to form $\gamma' = (v_0, \ldots, v_{k-1}, x)$ only if

$$2\mathrm{scale}_x \leq \frac{\varepsilon}{4} r \leq \frac{\varepsilon(1+\varepsilon)}{4}\mathrm{rad}(\gamma) < \varepsilon \mathrm{rad}(\gamma)$$

from Equation (7.1). As $\mathrm{rad}(\gamma) \leq \mathrm{rad}(\gamma')$, it follows that $\mathrm{scale}_x \leq \varepsilon \mathrm{rad}(\gamma')$, which proves the claim for γ'. The result follows by induction. ■

Indeed, this result shows that for any node in a well-separated tuple generated by our algorithm, the diameter of the node is much smaller than the radius of the ball which covers the tuple.

7.2 Čech approximation

In this section, we present a simplicial tower to $(1+\varepsilon)$-approximate the Čech filtration of P. The construction is based on the concept of WSSDs from the previous section. First, we introduce some further notation to simplify the presentation.

For a level $\ell \in \mathbb{Z}$ of the net-tree $\mathcal{T}$, we denote by

$$\mathcal{T}_\ell := \{u \in \mathcal{T} \mid \ell(u) \leq \ell < \ell(\mathrm{par}(u))\},$$

the highest set of nodes in $\mathcal{T}$, whose levels are at most ℓ. For any node $v \in \mathcal{T}$ of level i or less, we denote by $\mathrm{vcell}(v,i)$ its ancestor in $\mathcal{T}_i$, that is, its highest ancestor having level at most i. Then, $\mathcal{T}_{-\infty}$ simply denotes the leaves of $\mathcal{T}$ representing the input.

We fix the following additional parameters:

- for any integer i, set $\theta_i := \left(1+\frac{2\varepsilon}{5}\right)^i$. The set of discrete values $\{\theta_i | i \in \mathbb{Z}\}$ determine the scales at which our approximation complex may change. More concretely, the approximation complex will be the same for all scales in the interval $\alpha \in [\theta_i, \theta_{i+1})$, for all i.

- For a given $\alpha > 0$, let Δ_α denote the integer

  $$\Delta_\alpha = \lfloor \log_{(1+\frac{2\varepsilon}{5})} \alpha \rfloor, \quad \text{that is,} \quad \theta_{\Delta_\alpha} \leq \alpha < \theta_{\Delta_\alpha+1}.$$

  Δ_α helps in determining the interval of scales in which α falls.

- We define h_α as the integer which satisfies

  $$\frac{2\tau}{\tau-1}\tau^{h_\alpha} \leq \frac{\varepsilon\theta_{\Delta_\alpha}}{7} < \frac{2\tau}{\tau-1}\tau^{h_\alpha+1}. \tag{7.4}$$

  The parameter h_α determines the level of the net-tree using which the approximation is constructed at scale α. Note that h_α depends on Δ_α, rather than on α itself. Consequently, for any $\alpha \in [\theta_k, \theta_{k+1})$, the same h_α is chosen.

When there is no ambiguity about α, we skip the subscripts and simply write $\Delta := \Delta_\alpha$ and $h := h_\alpha$. Let W denote an $\frac{\varepsilon}{154}$-WSSD on $\mathcal{T}$. Before we formally describe our construction, we prove the following useful lemma:

Lemma 7.2.1. *For some $\alpha > 0$, let $\Delta := \Delta_\alpha$ and $h := h_\alpha$. If an $\frac{\varepsilon}{154}$-WST $t = (v_0, \ldots, v_k)$ satisfies* $\mathrm{rad}(t) \leq \theta_{\Delta+1}$, *then the level of each v_i in $\mathcal{T}$ is h or smaller.*

Proof. By Lemma 7.1.14, each v_i of t satisfies

$$\begin{aligned}\mathrm{scale}_{v_i} &\leq \frac{\varepsilon}{154}\mathrm{rad}(t) \leq \frac{\varepsilon}{154}\theta_{\Delta+1} \\ &\leq \frac{\varepsilon}{154}\left(1+\frac{2\varepsilon}{5}\right)\theta_\Delta \leq \frac{\varepsilon\theta_\Delta}{77}.\end{aligned}$$

Using the fact that $\tau = 11$ and Equation 7.4, we have

$$\frac{2\tau^{\ell(v_i)+1}}{\tau-1} = \mathrm{scale}_{v_i} \leq \frac{1}{11}\left(\frac{\varepsilon\theta_\Delta}{7}\right) \leq \frac{1}{11}\left(\frac{2\tau}{\tau-1}\tau^{h+1}\right) = \frac{2\tau^{h+1}}{\tau-1}.$$

It follows that $\ell(v_i) \leq h$, for all i. ■

7.2.1 Approximation tower

For any scale $\alpha > 0$, we denote our approximation complex at scale α by $\mathcal{A}_\alpha$. The vertices of $\mathcal{A}_\alpha$ are the nodes of $\mathcal{T}_h$. The simplices of $\mathcal{A}_\alpha$ can be found as: for any WST $t = (v_0, \ldots, v_k)$ with all v_i at level h or less, $\mathcal{A}_\alpha$ contains the simplex $t' = (\mathrm{vcell}(v_0, h), \ldots, \mathrm{vcell}(v_k, h))$, if $\mathrm{rad}(t') \le \theta_\Delta$.

Lemma 7.2.2. *$\mathcal{A}_\alpha$ is a simplicial complex.*

Proof. Consider any simplex $\gamma = (v_0, \ldots, v_k) \in \mathcal{A}_\alpha$. To prove the claim, we show that all faces of γ lie in $\mathcal{A}_\alpha$. Without loss of generality, let $t = (v_0, \ldots, v_l)$ be a face of γ. To show that t belongs to $\mathcal{A}_\alpha$, we first note that $\mathrm{rad}(t) \le \mathrm{rad}(\gamma)$ by definition. We only need to show that there exists a WST with nodes at level h or lower, such that taking the ancestor of each node in $\mathcal{T}_h$ gives the simplex t.

Consider the simplex $\sigma = (\mathrm{rep}_{v_0}, \ldots, \mathrm{rep}_{v_l})$. There exists a WST $t' = (v'_0, \ldots, v'_l)$ which covers σ, from Lemma 7.1.10. The ball $(1 + \varepsilon/154)\mathrm{meb}(\sigma)$ is an enclosing ball for all points of the $(\varepsilon/154)$-WST t'. Therefore,

$$\begin{aligned}\mathrm{rad}(t') &\le (1 + \frac{\varepsilon}{154})\mathrm{rad}(\sigma) = (1 + \frac{\varepsilon}{154})\mathrm{rad}(t)\\ &\le (1 + \frac{\varepsilon}{154})\mathrm{rad}(\gamma) \le (1 + \frac{\varepsilon}{154})\theta_\Delta \le \theta_{\Delta+1},\end{aligned}$$

which implies that the level of each v'_i is at most h, from Lemma 7.2.1. Then, for all i we have $\mathrm{vcell}(v'_i, h) = v_i$ because v'_i and v_i share the point rep_{v_i}, and v_i has level at most h by construction. Hence, by definition, the simplex t belongs to $\mathcal{A}_\alpha$. ■

We define maps connecting $\mathcal{A}_\alpha$ at different scales. Consider any two scales $0 \le \alpha_1 < \alpha_2$. We set $\Delta_1 := \Delta_{\alpha_1}, h_1 := h_{\alpha_1}$, and define Δ_2, h_2 accordingly. Since $h_1 \le h_2$, there is a natural vertex map $g_{\alpha_1}^{\alpha_2} : \mathcal{T}_{h_1} \to \mathcal{T}_{h_2}$, which takes a net-tree node of $\mathcal{T}_{h_1}$ to its ancestor in $\mathcal{T}_{h_2}$. This naturally extends to a map on the complexes

$$g_{\alpha_1}^{\alpha_2} : \mathcal{A}_{\alpha_1} \to \mathcal{A}_{\alpha_2},$$

by mapping any simplex $\sigma = (v_0, \ldots, v_k) \in \mathcal{A}_\alpha$ to $g_{\alpha_1}^{\alpha_2}(\sigma) := (g_{\alpha_1}^{\alpha_2}(v_0), \ldots, g_{\alpha_1}^{\alpha_2}(v_k))$.

Lemma 7.2.3. *$g := g_{\alpha_1}^{\alpha_2} : \mathcal{A}_{\alpha_1} \to \mathcal{A}_{\alpha_2}$ is a simplicial map.*

Proof. Let $t = (v_0, \ldots, v_k)$ be any k-simplex of $\mathcal{A}_{\alpha_1}$. Let $g(v_i) = v'_i$ denote the ancestor of v_i in $\mathcal{T}_{h_2}$, for all i. We show that $t' := (v'_0, \ldots, v'_k) \in \mathcal{A}_{\alpha_2}$. If $\Delta_1 = \Delta_2$, the statement is trivial since $h_1 = h_2$, $\mathcal{A}_{\alpha_1} = \mathcal{A}_{\alpha_2}$ and g is the identity map. So, we assume that $\Delta_1 < \Delta_2$.

Consider the minimum enclosing ball of t. It contains the representatives of all nodes v_i and hence at least one point of each of v'_i. If we inflate the ball by the largest diameter of a node at level h_2, all v'_i will be covered completely. We show that the inflated radius is less than θ_{Δ_2}, which immediately implies the claim using Lemma 7.2.1.

The diameter of a node u at level h_2 is at most $2\mathrm{scale}_u \le \frac{2\varepsilon\theta_{\Delta_2}}{7}$. Also, since $\Delta_1 < \Delta_2$, we have $\theta_{\Delta_1} \le \frac{\theta_{\Delta_2}}{1+\frac{2\varepsilon}{5}}$. Therefore,

$$\begin{aligned}\mathrm{rad}(t') &\le \mathrm{rad}(t) + \frac{2\varepsilon\theta_{\Delta_2}}{7} \le \theta_{\Delta_1} + \frac{2\varepsilon\theta_{\Delta_2}}{7}\\ &\le \frac{\theta_{\Delta_2}}{1 + \frac{2\varepsilon}{5}} + \frac{2\varepsilon\theta_{\Delta_2}}{7} \le \frac{1 + \frac{2\varepsilon}{7} + \frac{4\varepsilon^2}{35}}{1 + \frac{2\varepsilon}{5}}\theta_{\Delta_2} < \theta_{\Delta_2},\end{aligned}$$

for $\varepsilon \le 1$. Hence $t' \in \mathcal{A}_{\alpha_2}$. ■

When the scales α_1, α_2 are clear from the context, we abbreviate $g_{\alpha_1}^{\alpha_2}$ as g. This complete the description of the approximation tower $(\mathcal{A}_\alpha)_{\alpha \ge 0}$.

7.2.2 Interleaving

To establish the approximation result, we first define maps between the Čech complexes and the approximation complexes.

We define a map $\phi : \mathcal{C}_{\frac{\alpha}{1+\varepsilon}} \to \mathcal{A}_\alpha$ for any $\alpha \geq 0$. For any vertex $p \in \mathcal{C}_{\frac{\alpha}{1+\varepsilon}}$ (which is a point of P), we set $\phi(p)$ as the net-tree node in $\mathcal{T}_h$ that covers p. For any simplex $(v_0, \ldots, v_k) \in \mathcal{C}_{\frac{\alpha}{1+\varepsilon}}$, define $\phi(v_0, \ldots, v_k) = (\phi(v_0), \ldots, \phi(v_k))$.

Lemma 7.2.4. *ϕ is a simplicial map.*

Proof. Let $\sigma = (p_0, \ldots, p_k)$ be any simplex in $\mathcal{C}_{\frac{\alpha}{1+\varepsilon}}$. Let $t = (v_0, \ldots, v_k)$ be a WST which covers σ. By the WSSD property, we have that

$$\mathrm{rad}(t) \leq \left(1 + \frac{\varepsilon}{154}\right) \mathrm{rad}(\sigma) \leq \frac{1 + \frac{\varepsilon}{154}}{1+\varepsilon} \alpha < \theta_{\Delta+1}.$$

Hence all v_i are at level at most h, using Lemma 7.2.1. Let $t' = (v_0', \ldots, v_k')$ where $v_i' = \mathrm{vcell}(v_i, h)$. We need to show that $t' \in \mathcal{A}_\alpha$. Similar to the proof of Lemma 7.2.3, if we inflate $\mathrm{meb}(t)$ by $\frac{2\varepsilon\theta_\Delta}{7}$, we cover t' and that proves the claim. Using $\alpha \leq \theta_{\Delta+1} = (1 + \frac{2\varepsilon}{5})\theta_\Delta$,

$$\begin{aligned}\mathrm{rad}(t') &\leq \mathrm{rad}(t) + \frac{2\varepsilon\theta_\Delta}{7} \leq \frac{1 + \frac{\varepsilon}{154}}{1+\varepsilon}\alpha + \frac{2\varepsilon\theta_\Delta}{7} \\ &\leq \frac{1 + \frac{533}{770}\varepsilon + \frac{111}{385}\varepsilon^2}{1+\varepsilon}\theta_\Delta \leq \theta_\Delta\end{aligned}$$

for $\varepsilon \leq 1$. It follows that $t' \in \mathcal{A}_\alpha$. ■

In the other direction, we define a map $\psi : \mathcal{A}_{\frac{\alpha}{1+\varepsilon}} \to \mathcal{C}_\alpha$ that takes a net-tree node v to its representative rep_v. For a simplex $t = (v_0, \ldots, v_k) \in \mathcal{A}_{\frac{\alpha}{1+\varepsilon}}$, we set $\psi(t) = (\mathrm{rep}_{v_0}, \ldots, \mathrm{rep}_{v_k})$. It follows that

$$\mathrm{rad}(\psi(t)) \leq (1+\varepsilon)\mathrm{rad}(t) \leq (1+\varepsilon)\frac{\alpha}{1+\varepsilon} \leq \alpha,$$

so that $\psi(t) \in \mathcal{C}_\alpha$. Therefore, ψ is a well-defined simplicial map.

$(\mathcal{C})_{\alpha\geq 0}$ and $(\mathcal{A})_{\alpha \geq 0}$ give rise to persistence modules $(H(\mathcal{C}_\alpha))_{\alpha\geq 0}$ and $(H(\mathcal{A}_\alpha))_{\alpha\geq 0}$, respectively. The respective modules are connected by linear maps f^* and g^*, where f is the inclusion map and the asterisks denote the induced linear maps on the homology groups. Let $\phi^* : H(\mathcal{C}_{\frac{\alpha}{1+\varepsilon}}) \to H(\mathcal{A}_\alpha)$ and $\psi^* : H(\mathcal{A}_{\frac{\alpha}{1+\varepsilon}}) \to H(\mathcal{C}_\alpha)$ be the linear maps corresponding to ϕ and ψ, respectively.

Lemma 7.2.5. *For all $0 \leq \alpha \leq \alpha'$, the diagrams*

$$\begin{tikzcd}
 & H(\mathcal{C}_\alpha) \arrow[r, "f^*"] & H(\mathcal{C}_{\alpha'}) \arrow[dr, "\phi^*"] & \\
H(\mathcal{A}_{\frac{\alpha}{1+\varepsilon}}) \arrow[ur, "\psi^*"] \arrow[rrr, "g^*"] & & & H(\mathcal{A}_{(1+\varepsilon)\alpha'})
\end{tikzcd}$$

$$\begin{tikzcd}
H(\mathcal{C}_\alpha) \arrow[rr, "f^*"] \arrow[dr, "\phi^*"] & & H(\mathcal{C}_{\alpha'}) \arrow[dr, "\phi^*"] & \\
 & H(\mathcal{A}_{(1+\varepsilon)\alpha}) \arrow[rr, "g^*"] & & H(\mathcal{A}_{(1+\varepsilon)\alpha'})
\end{tikzcd}$$

commute, that is, $\phi^* \circ f^* \circ \psi^* = g^*$ for the upper diagram and $\phi^* \circ f^* = g^* \circ \phi^*$ for the right diagram.

Proof. The maps commute on the simplicial level, that is, $\phi \circ f \circ \psi = g$ and $\phi \circ f = g \circ \phi$, respectively, as can be easily verified from the definition of the maps. As a result, the induced linear maps also commute. ■

Lemma 7.2.6. *For all* $0 \leq \alpha \leq \alpha'$, *the diagram*

$$\begin{tikzcd}
H(\mathcal{C}_{\frac{\alpha}{1+\varepsilon}}) \arrow[rrr, "f^*"] \arrow[dr, "\phi^*"] & & & H(\mathcal{C}_{(1+\varepsilon)\alpha'}) \\
& H(\mathcal{A}_\alpha) \arrow[r, "g^*"] & H(\mathcal{A}_{\alpha'}) \arrow[ur, "\psi^*"] &
\end{tikzcd}$$

commutes, that means, $\psi^* \circ g^* \circ \phi^* = f^*$.

Proof. In this case, the corresponding simplicial maps $\psi \circ g \circ \phi$ and f do not commute in general. We will show instead that they are contiguous (see Definition 2.4.4). We fix any simplex $\sigma = (p_0, \ldots, p_k) \in \mathcal{C}_{\frac{\alpha}{1+\varepsilon}}$. For all i, let $q'_i := g(\phi(p_i))$. By definition, q'_i is a net-tree node that covers p_i, at the level h' corresponding to α'. By definition $w_i := \psi(q'_i) = \psi(g(\phi(p_i))$ is the representative of q'_i. From the definition of $\mathcal{A}_{\alpha'}$, we have that $\operatorname{rad}(w_0, \ldots, w_k) = \operatorname{rad}(q'_0, \ldots, q'_k) \leq \theta_{\Delta'} \leq \alpha'$, where $\Delta' := \Delta_{\alpha'}$.

Since q'_i contains both p_i and w_i, inflating $\operatorname{meb}(w_0, \ldots, w_k)$ by the largest diameter of any node at the level h', will cover the simplex $\sigma = (p_0, \ldots, p_k, w_0, \ldots, w_k)$. The diameter of nodes at level h' is at most $\frac{2\varepsilon\theta_{\Delta'}}{7}$. The radius of a ball required to cover σ is at most

$$\begin{aligned}
\operatorname{rad}(p_0, \ldots, p_k, w_0, \ldots, w_k) &\leq \operatorname{rad}(w_0, \ldots, w_k) + \frac{2\varepsilon\theta_{\Delta'}}{7} \\
&\leq \alpha' + \frac{2\varepsilon}{7}\alpha' < (1+\varepsilon)\alpha'.
\end{aligned}$$

Therefore, the simplex $(p_0, \ldots, p_k, w_0, \ldots, w_k)$ is in $\mathcal{C}_{(1+\varepsilon)\alpha'}$, so $\psi \circ g \circ \phi$ and f are contiguous. As a result, the induced linear maps commute. ■

Lemma 7.2.7. *For all* $0 \leq \alpha \leq \alpha'$, *the diagram*

$$\begin{tikzcd}
& H(\mathcal{C}_{(1+\varepsilon)\alpha}) \arrow[rr, "f^*"] & & H(\mathcal{C}_{(1+\varepsilon)\alpha'}) \\
H(\mathcal{A}_\alpha) \arrow[ur, "\psi^*"] \arrow[rr, "g^*"] & & H(\mathcal{A}_{\alpha'}) \arrow[ur, "\psi^*"] &
\end{tikzcd}$$

commutes, that means, $\psi^* \circ g^* = f^* \circ \psi^*$.

Proof. Again, the corresponding simplicial maps $\psi \circ g$ and $f \circ \psi$ do not commute in general (they do only if $h_\alpha = h_{\alpha'}$). We show that the simplicial maps are contiguous, which implies the claim. Fix some $t = (q_0, \ldots, q_k) \in \mathcal{A}_\alpha$ and let v_ℓ be the representative of q_ℓ; in particular $f \circ \psi(q_\ell) = v_\ell$. Now, set $q'_\ell := g(q_\ell)$. It is clear that $P_{q_\ell} \subseteq P_{q'_\ell}$ and therefore, $v_\ell \in q'_\ell$. Set $w_\ell := \psi(g(q_\ell)) = \psi(q'_\ell)$ to be the representative of q'_ℓ. By the same argument as in Lemma 7.2.6, $\operatorname{rad}(v_0, \ldots, v_k, w_0, \ldots, w_k) \leq (1+\varepsilon)\alpha'$, which implies the claim. ■

Theorem 7.2.8. *The persistence module* $(H(\mathcal{A}_\alpha))_{\alpha \geq 0}$ *and the Čech persistence module* $(H(\mathcal{C}_\alpha))_{\alpha \geq 0}$ *are* $(1+\varepsilon)$*-approximations of each other.*

Proof. The diagrams in lemmas 7.2.5-7.2.7 commute, which implies that the two persistence modules are strongly $(1+\varepsilon)$-interleaved. The claim then follows from Theorem 2.3.6. ■

Theorem 7.2.9. *The approximation tower has size at most* $n(2/\varepsilon)^{O(\Upsilon^2)}$.

Proof. A net-tree has n leaves and at most n internal nodes, which form the vertices of the approximation tower. Therefore, the number of vertex inclusions is upper bounded by $2n$.

We bound the number of simplex inclusions in the tower. Consider any WST $t = (v_0, \ldots, v_k)$. Let Δ be the smallest integer and h the corresponding level in the net-tree such that $\mathrm{rad}(t') \leq \theta_\Delta$, where $t' := (\mathrm{vcell}(v_0, h), \ldots, \mathrm{vcell}(v_k, h))$. Then, the simplex t' exists in the complex $\mathcal{A}_{\theta_\Delta}$. For all integers Δ' lower than Δ, t does not contribute a simplex in the tower, since $\mathrm{rad}\big((\mathrm{vcell}(v_0, h'), \ldots, \mathrm{vcell}(v_k, h'))\big) > \theta_{\Delta'}$ at those scales, by our choice of Δ. This means that the simplex t' is included in the tower at the scale θ_Δ. Let Δ'' be any scale larger than Δ such that $\mathrm{rad}\big((\mathrm{vcell}(v_0, h''), \ldots, \mathrm{vcell}(v_k, h''))\big) \leq \theta_{\Delta''}$. By definition, t contributes a simplex to $\mathcal{A}_{\theta_{\Delta''}}$. However, this simplex is simply the image of t' under the map $g : \mathcal{A}_{\theta_\Delta} \to \mathcal{A}_{\theta_{\Delta''}}$. Therefore, the WST t contributes to only one simplex inclusion in the entire tower. As a result, the number of simplex inclusions in the tower is upper bounded by the size of the WSSD, which is $n(2/\varepsilon)^{O(\Upsilon^2)}$ from Corollary 7.1.12. ■

7.3 Discussion

In this chapter, we gave a generalization of WSPDs to higher dimensions and showed that they are useful in representing all simplices on a point set, in a compact manner. We considered the setup in doubling spaces; a similar result was presented in [KS13] without the assumption of doubling dimension. They used quad-trees for their scheme, while we used net-trees to arrive at an equivalent result, which is superior when Υ is low.

We also presented a tower to $(1+\varepsilon)$-approximate the Čech filtration for point sets of low doubling dimension. On the other hand, we did not explicitly give an algorithm for our method. The naive algorithm is rather expensive, since at each scale, it requires the computation of minimum enclosing balls and vcell for each WST tuple. Computing vcell can be quite expensive since requires $O(n^2)$ space. Finding an efficient algorithm to compute the approximation tower is an item for future agenda.

An alternative idea to construct the Čech-approximation would be to directly use the net-tree, bypassing the step of constructing WSSDs. An approximation complex at scale α can be derived by building a Čech complex directly on the set of nodes of $\mathcal{T}_h$; the simplicial maps can be built using the ancestor relationship. However, it remains to see if this method can lead to an efficient algorithm.

We believe that the concept of WSSDs is interesting in its own right and hopefully applicable in different contexts. It may be worthy to identify further application scenarios in the future.

Chapter 8

Local Doubling Dimension

Doubling dimension (Definition 3.3.2) is a way to capture the intrinsic structure of a metric space. It is evident from the definition that it depends on the global properties of the metric space. In some applications, information is only relevant for a fixed range of distances, and the associated metric may have a lower intrinsic dimension in this range. In such cases, the doubling dimension may not be the ideal representative for capturing the structure at those scales.

To utilize this structural property, we introduce a notion of local intrinsic dimension of metric spaces in this chapter, which follows naturally from the concept of doubling dimension. We make use of this concept in an algorithmic context by presenting a method to construct a hierarchical net-tree up to the desired scale, that avoids inspecting higher scales. This ensures that computing the tree takes time that is dominated by the local intrinsic dimension. The construction utilizes Locality-sensitive hashing for Euclidean spaces (Section 3.4). We also apply our concepts to various applications, where the range of interesting scales is restricted.

8.1 Definition and net-forests

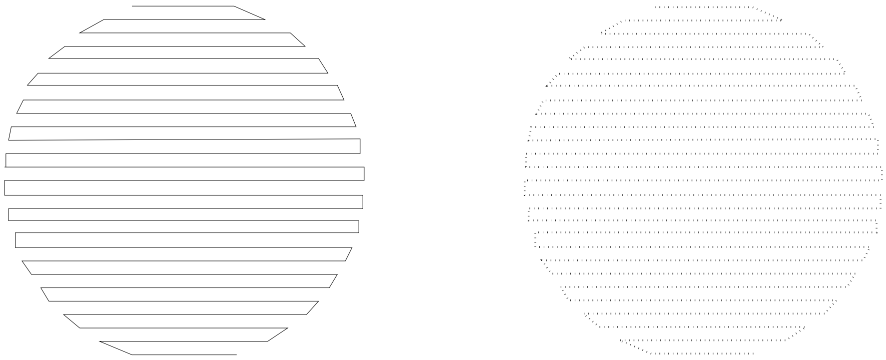
Let $P \subset \mathbb{R}^d$ be a point set with doubling dimension Υ and $n := |P|$.

Definition 8.1.1 (restricted doubling dimension). *Given $t > 0$, the* t-restricted doubling constant *of P is the smallest positive integer λ_t such that all the points of P in any ball centered at any $p \in P$ of radius r are covered by at most λ_t non-empty balls of radius $r/2$, for $0 \leq r \leq t$.*

The t-restricted doubling dimension *of P is $\Upsilon_t := \lceil \log_2 \lambda_t \rceil$.*

As is evident, this is a re-definition of the doubling dimension such that the defining balls have bounded radius, so that $\Upsilon_t \leq \Upsilon$. More precisely, Υ_t is zero for t smaller than the closest-pair distance of P, and equals Υ when t is at least the diameter of P. We sometimes call the restricted doubling dimension as the *local doubling dimension.*

We give a motivating example to illustrate the utility of the local doubling dimension. For point samples from an affine subspace of dimension k, Υ is bounded by $\Theta(k)$. This is not true in general for samples of k-manifolds, where Υ increases due to curvature. To sketch an extreme example, consider an almost space-filling curve γ in $\mathbb{R}^d$ which has distance at most ε to any point of the unit ball, for a small enough value of ε. We let P be a sufficiently dense sample of γ. For small values of t, $\Upsilon_t = 1$ since the underlying structure is 1-dimensional. On the other hand, $\Upsilon_t = \Theta(d)$ for $t = 1$; indeed, any sparser covering of the unit ball with balls of radius $1/2$ would leave some portion of the ball



(a) A curve which approximates the unit ball.

(b) A point sample: connecting vertices which are a small distance apart gives back the original curve, so $\Upsilon = 1$ at such scales. At large scales it is almost a uniform sample of the ball, so Υ depends on the ambient dimension.

Figure 8.1: An almost space filling curve.

uncovered, and by construction, γ goes through that uncovered region, so that some point in P is missed. See Figure 8.1 for an intuition. Therefore, choosing the right value of t is critical to understand the structure of the hidden manifold.

The "badness" of the previous example stems from the discrepancy between the Euclidean and geodesic distances of points lying on a lower-dimensional manifold. A common technique for approximating the geodesic distance is through the *shortest-path metric*: let $G = (P, E)$ denote the graph whose edges are defined by the pairs of points of P with Euclidean distance at most t; we call such a graph a t-intersection graph. The geodesic distance of two points p and q is then defined as the length of the shortest path from p to q in G (we assume for simplicity that G is connected). The concept of doubling dimension extends to any metric space, so let Υ' denote the doubling dimension of P equipped with the shortest path metric. While Υ_t and Υ' appear to be related, Υ' can be much larger than Υ_t: an example is shown in Figure 8.2. Moreover, using the shortest-path metric raises the question of how to compute shortest path distances efficiently, if the cost of metric queries is taken into account.

8.1.1 Net-forest

Next, we discuss a data structure for point sets, whose properties are primarily dependent on the local doubling dimension.

Definition 8.1.2 (net-forest). *For a desired parameter $t > 0$, a* net-forest *on a point set P is defined as a collection of net-trees with roots $\{v_1, \ldots, v_k\}$ such that their representatives $\{\mathrm{rep}_{v_1}, \ldots, \mathrm{rep}_{v_k}\}$ form a (t,t)-net, and the point sets $\{P_{v_1}, \ldots, P_{v_k}\}$ are a partition of P.*

We define $\mathrm{Rel}(u)$ for a node u in the net-forest in the same way as for net-trees: it is the set of net-forest nodes that are close to u and have similar diameter (see Sub-section 3.3.2). Similar to net-trees, we call a net-forest *augmented* if each node u is equipped with information about $\mathrm{Rel}(u)$. See Sub-section 3.3.2 for more details.

The natural way to build a net-forest is to simply construct a net-tree, and to truncate it up to scale t to get a forest of sub-trees. Since the nodes of this forest only interact at distances of the form $O(t)$, the properties of this forest are dependent on $\Upsilon_{O(t)}$.

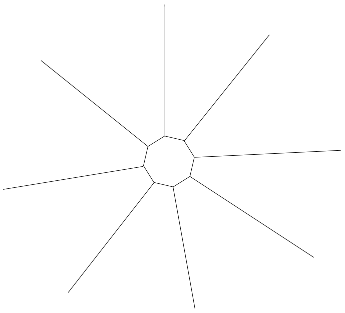


Figure 8.2: Consider a regular k-gon in the plane with all vertices on the unit circle. To each vertex, attach an "arm" of length 4 in the direction outwards from the origin. We call the endpoint of such an arm a *tip*. Let X denote the obtained shape, as depicted in the figure. Let P be a sufficiently dense point sample of X and G be the t-intersection graph on P. We choose t appropriately so that the shortest path metric on G approximates the distances on X very closely. Fixing an arbitrary vertex of the k-gon, the furthest tip has a distance of at most $4 + \pi < 8$ on X, so there is a ball (on the shortest path metric) of radius less than 8 containing all of P. However, any pair of tips has a distance of more than 8, so no metric ball of radius less than 4 can contain two tips. It follows that any covering of P with metric balls of radius less than 4 requires at least one metric ball per tip. Therefore, the doubling dimension Υ' of this metric is at least $\lceil \log_2 k \rceil$. On the other hand, for the Euclidean metric on $\mathbb{R}^2$, we can easily see that the doubling dimension $\Upsilon = O(1)$, and therefore, Υ_t is a constant as well.

Unfortunately, the net-tree construction proceeds in a top-down fashion (see [HPM06] or Sub-section 3.3.2 for more details), considering the large scales of the point set first. Thus, the algorithm has a complexity that depends on the doubling dimension of the point set. Therefore, such a construction suffers from potentially bad large-scale properties of the point set, even when they are irrelevant for the given application. To tackle this problem, we use a slightly different construction, that avoids exploring scales above $O(t)$, thereby avoiding the complexities associated with large scales.

Construction

We discuss our algorithm for constructing a net-forest, which is a simple adaptation of the net-tree algorithm. First, we construct a (t, t)-net on P by clustering the point set and assigning to each point in P its closest net-point. Each root in the net-forest represents one of the clusters. We also compute $\mathrm{Rel}(u)$ for each root by finding the close-by clusters to node u. Having this information, we can simply run the net-tree algorithm from [HPM06] individually on each cluster to construct the net-forest. For augmenting the forest, we use the top-down traversal strategy as described in [HPM06, Section 3.4], that infers the neighbors of a node from the neighbors of its parent. Since we have set up $\mathrm{Rel}(\cdot)$ for the roots of the forest, this strategy is guaranteed to detect neighboring nodes for all internal nodes, even if they belong to different trees of the forest.

Both the initial net construction and the Rel()-construction require solving the *all near neighbors problem* (Definition 3.4.3). We use the near neighbor primitive from Section 3.4 based on Locality-sensitive Hashing (LSH) to tackle the problem. Recall that LSH has three main parameters: r denotes the query radius, $0 < \rho < 1$ is a tunable parameter, and S denotes the set of points on which LSH acts. We use this notation in the rest of the section.

Net construction We construct the net using a greedy scheme: for each input point p, we store a pointer $\mathrm{N}(p)$ pointing to the net point assigned to point p. Initially, we set $\mathrm{N}(p) \leftarrow \varnothing$ for all $p \in P$. As long as there is a point p such that $\mathrm{N}(p) = \varnothing$, we set $\mathrm{N}(p) \leftarrow p$ and query the near-neighbor primitive to get a list of points with distance at most t from p. For each point q in the list we update $\mathrm{N}(q) \leftarrow p$ if either $\mathrm{N}(q) = \varnothing$ or $\|p - q\| < \|N(q) - q\|$. Then we pick the next point p with $\mathrm{N}(p) = \varnothing$. At the end, the set of points p with $\mathrm{N}(p) = p$ represents the net at scale t and the points q satisfying $\mathrm{N}(q) = p$ constitute p's cluster. The net thus constructed is a (t, t)-net. Moreover, we set the level $\ell(v) = \lfloor \log_\tau \frac{(\tau-1)t}{2\tau} \rfloor$ for each root node v, where $\tau = 11$ is the constant from the definition of net-trees (Definition 3.3.5). To find the nearby neighbors, we simply initialize the near-neighbor primitive with $r \leftarrow t$ and $S \leftarrow P$, and query the appropriate points.

Computing the Rel **set for the roots** After finding the net-points and their respective clusters, we need to augment the root nodes with neighboring information. Recall that for any node u, $\mathrm{Rel}(u)$ contains nodes in distance at most $14\tau^{\ell(u)}$ from rep_u. Since we have a (t, t)-net, the level of any root node u satisfies $14\tau^{\ell(u)} \leq 7t$. Hence we need to find neighbors of net-points within distance $7t$, where the minimum distance between any two net-points is at least t. By the doubling property and [Tal04, Proposition 1], any root net-node can have at most $\lambda_{7t}^{\log_2 \frac{7t}{t/2}}$ such neighbors which simplifies to $14^{\Upsilon_{7t}}$. We use the near-neighbor primitive to compute such neighbors. Let M denote the set of net-points and $m := |M|$. We simply have to find all pairs of points of M that have distance at most $7t$; and to do so we call the near neighbor primitive with $r \leftarrow 7t$ and $S \leftarrow M$ for all $q \in M$.

Lemma 8.1.3. *The expected time to construct the (t, t)-net using LSH is*

$$O\Big(dn^{1+\rho} \log n \Big(\log n + \Big(\frac{2}{\rho}\Big)^{\Upsilon_{t/\rho}} \Big)\Big),$$

where $0 < \rho < 1$ is the LSH parameter.

Proof. We consider the time spent on all near-neighbor queries. The resulting net consist of $m \leq n$ points. This implies that the algorithm proceeds in m rounds and queries the near-neighbors of m points. Let $\widetilde{C}_i$ denote the number of points in distance t/ρ from the i-th query point. By Lemma 3.4.4, the total complexity for all near-neighbor queries is:

$$O\left(ndkl + \sum_{i=1}^{m} dl(k + \widetilde{C}_i)\right) = O(ndkl + dl \sum_{i=1}^{m} \widetilde{C}_i),$$

where k, l are LSH parameters. We only need to bound the sum of the quantities $\widetilde{C}_i$. For that, we fix some $q \in P$ and count in how many sets $\widetilde{C}_i$ may it appear. Let p_i denote the net-point chosen in the i-th iteration. We call such a net-point *close* to q if the distance to q is at most t/ρ. By definition, the net-points close to q lie in a ball of radius t/ρ centered at q. Since any pair of net-points has a distance of more than t, any ball of radius $t/2$ can contain at most one close net point. Following the definition of the t-restricted doubling dimension, the number of such net-points can be at most $\lambda_{t/\rho}^{\log_2 \frac{t/\rho}{t/2}}$ which simplifies to $\left(\frac{2}{\rho}\right)^{\Upsilon_{t/\rho}}$. It follows that $\sum_{i=1}^{m} \widetilde{C}_i \leq n \left(\frac{2}{\rho}\right)^{\Upsilon_{t/\rho}}$. Hence, we get the claimed running time, by substituting $k = O(\log n)$ and $l = O(n^\rho \log n)$ from Lemma 3.4.4. All additional operations in the net construction besides the calls of the near-neighbor primitive are dominated by the claimed bound. ■

Lemma 8.1.4. *Computing the* Rel *sets using LSH takes expected time*

$$O\left(dn^{1+\rho}\log n\left(\log n+\left(\frac{14}{\rho}\right)^{\Upsilon_{7t/\rho}}\right)\right).$$

Proof. The proof is analogous to the proof of Lemma 8.1.3: let $\widetilde{C}_i$ (for $i = 1, \ldots, m$) denote the number of net-points in distance at most $\frac{7t}{\rho}$ to the i-th net point. A packing argument similar to the one in the previous Lemma shows that any $\widetilde{C}_i$ can be at most $\left(\frac{14}{\rho}\right)^{\Upsilon_{7t/\rho}}$, so that their sum is bounded by $m\left(\frac{14}{\rho}\right)^{\Upsilon_{7t/\rho}}$. Since $m \leq n$, the runtime follows by plugging in the respective values in Lemma 3.4.4. ■

Theorem 8.1.5. *The expected time for constructing the net-forest using LSH is*

$$O\left(dn^{1+\rho}\log n\left(\log n+\left(\frac{14}{\rho}\right)^{\Upsilon_{7t/\rho}}\right)\right).$$

Proof. Using Lemma 8.1.3 and Lemma 8.1.4, constructing the root clusters and its Rel sets are within the stated complexity bound. Constructing a single net-tree for a root node containing n_i points takes time at most $2^{14\Upsilon_{2t}}dn_i \log n_i$ (the factor of 14 in the exponent can be seen by a careful analysis of [HPM06, Section 3.4]). Constructing individual net-trees for the clusters takes time: $\sum_{i=1}^{m} 2^{14\Upsilon_{2t}}dn_i \log n_i$. Since $\sum_{i=1}^{m} n_i = n$, the above runtime is upper bounded by $2^{14\Upsilon_{2t}}dn \log n$. Augmenting the net-forest takes time $dn2^{14\Upsilon_{7t}}$ [HPM06, Section 3.4]. The runtimes for the net-tree constructions and augmentation are dominated by the Rel- and net-construction for sufficiently large values of n. ■

In comparison, constructing the full net-tree takes $n \log n 2^{O(\Upsilon)}$ time [HPM06]. Comparing the complexity bound in Theorem 8.1.5 with the naive step of constructing the full net-tree and pruning it at the appropriate level, our approach makes sense if $n^\rho \log n \ll 2^{O(\Upsilon)}$ and $\Upsilon_{O(t)} \ll \Upsilon$. We also see how the choice of ρ affects the complexity bound: for ρ very close to zero, we get an almost linear complexity in n, for the price that we have to consider larger balls in our algorithm and thus increase the restricted doubling dimension.

8.2 Applications

We now discuss a few applications of net-forest. The first deals with WSPDs (Definition 3.3.7) and SSPDs (Definition 3.3.9) from Chapter 3. The second deals with approximation schemes for Rips filtration [She13] and Čech filtrations (Chapter 7). For brevity, we only discuss the extensions of these concepts in this section.

8.2.1 Pair decompositions

Well-separated pair decomposition Using a net-tree on P, one can build an ε-WSPD of size $n\varepsilon^{-O(\Upsilon)}$ in time $d\left(2^{O(\Upsilon)}n \log n + n(1/\varepsilon)^{O(\Upsilon)}\right)$ [HPM06], for $\varepsilon > 0$. A WSPD considers pairs of points over all scales of distance, because it has to cover any pair of points of P. We only require that all pairs of points of P in distance at most t are covered; and we call the resulting structure a *t-restricted ε-WSPD.*

We construct the t-restricted ε-WSPD as follows: we start by constructing the corresponding augmented net-forest; let $\{u_1, \ldots, u_m\}$ denote the root nodes. Since we know the Rel set for any root, we can identify pairs (u_i, u_j) such that u_i is in $\mathrm{Rel}(u_j)$ and vice versa (this also includes pairs where $u_i = u_j$). For any such pair, we call `genWSPD`(u_i, u_j)

from [HPM06, Section 5], which simply traverses the sub-trees until it finds well-separated pairs. We output the union of all pairs generated in this way.

Theorem 8.2.1. *For any* $0 < \varepsilon < 1$ *and* $t > 0$*, our algorithm computes a* t*-restricted* ε*-WSPD of size* $n\varepsilon^{-O(\Upsilon_{7t})}$ *in expected time* $NF + dn\varepsilon^{-O(\Upsilon_{7t})}$*, where* NF *is the complexity for computing the net-forest from Theorem 8.1.5.*

Proof. We prove correctness first; we already know by definition that each pair of nodes which is generated is ε-well-separated. To show that each pair of points in distance t is covered, consider any pair of points $(p, q) \in P$ in distance at most t. There are roots u_1, u_2 in the net-forest with $p \in P_{u_1}$ and $q \in P_{u_2}$. Since the diameters of u_1 and u_2 both are at most $2t$, the distance of rep_{u_1} and rep_{u_2} is at most $5t < 7t$. Therefore, $u_2 \in \mathrm{Rel}(u_1)$ (and vice-versa), which ensures that the algorithm of [HPM06] ensures that there will be a generated pair that covers (p, q).

For the size bound, we use the same charging argument as in [HPM06, Section 5]. We additionally ensure by our construction that in all doubling arguments, the radii of the balls in question are at most $7t$ and therefore replace the doubling dimension by Υ_{7t} in the bound. The running time follows because the number of recursive calls of `genWSPD` is proportional to the output size, and we spend $O(d)$ time per recursion step. ■

Semi-separated pair decomposition Recall that for a given $\varepsilon > 0$, an ε-SSPD of expected weight $O(\varepsilon^{-O(\Upsilon)} n \log n)$ can be calculated in $O(\varepsilon^{-O(\Upsilon)} n \log n)$ expected time [AHP12]. We adapt the algorithm of [AHP12, Section 4] slightly to compute a t-restricted SSPD, which requires that only pairs of points in distance at most t be covered by the decomposition. First we construct the net-forest at scale $t/5$. Let $\{u_1, \ldots, u_m\}$ denote the root nodes of the net-forest. We invoke the algorithm of [AHP12, Section 4] on each of the sets $\{u_i \cup \mathrm{Rel}(u_i)\}$ and output the union of the pairs generated by each invocation. The pairs involved in the SSPD have the property that their diameter is at most $3.2t$.

Theorem 8.2.2. *For any* $0 < \varepsilon < 1$*, our algorithm computes a* t*-restricted* ε*-SSPD of* P *of expected weight* $n \log n \left(\frac{2}{\varepsilon}\right)^{O(\Upsilon_{3.2t})}$ *in* $n \log n \left(\frac{2}{\varepsilon}\right)^{O(\Upsilon_{3.2t})}$ *expected time, after computing the net-forest at scale* t*.*

Proof. To prove correctness, consider points $p, q \in P$ such that $\|p - q\| \le t$. Let u_i and u_j be the root nodes covering p and q, respectively. By the triangle inequality,

$$\begin{aligned} \|rep_{u_i} - rep_{u_j}\| &\le \|rep_{u_i} - p\| + \|p - q\| + \|q - rep_{u_j}\| \\ &\le t/5 + t + t/5 \le 7t/5. \end{aligned}$$

Hence, $u_i \in \mathrm{Rel}(u_j)$ and vice-versa. This ensures that the algorithm creates a semi-separated pair which covers the pair (p, q). For the diameter, let us say that we work with the set $P_m = \{u \cup \mathrm{Rel}(u)\}$ and wish to upper bound $\mathrm{diam}(P_m)$. For any pair of root nodes $u_i, u_j \in \mathrm{Rel}(u)$ covering points a and b respectively, it follows from the triangle inequality that $\|a - b\| \le 3.2t$.

Next we bound the weight of the SSPD. Consider the individual invocations of the algorithm of [AHP12] after constructing the net-forest. Let N_i denote the number of points in $u_i \cup \mathrm{Rel}(u_i)$. Then the total weight of the SSPD is

$$\sum_{i=1}^{m} \varepsilon^{-O(\Upsilon_{3.2t})} N_i \log N_i \le \varepsilon^{-O(\Upsilon_{3.2t})} \log n \sum_{i=1}^{m} N_i.$$

To bound the sum $S := \sum_{i=1}^{m} N_i$, consider any point p covered by u_i. This point participates in at most $2^{O(\Upsilon_{7t/5})}$ invocations of the algorithm, since that is precisely the maximum possible size of the Rel sets of the roots. This implies that the contribution of point p to S is at most $2^{O(\Upsilon_{7t/5})}$. Hence the sum S is at most $2^{O(\Upsilon_{7t/5})}n$, and bound follows. A similar argument bounds the runtime as well. ■

8.2.2 Čech and Rips filtrations

In practice, it is common to truncate Rips and Čech filtrations up to a certain scale, which is of interest in the application. For the case that the highest scale in the construction is bounded by t, we can improve the complexities of approximation schemes based on net-trees, in the sense that Υ can be replaced with Υ_{7t} in the complexity bounds. This is done by replacing the net-tree with a net-forest at scale t in the respective algorithms.

Theorem 8.2.3. *There exist $(1+\varepsilon)$-approximate towers for Rips and Čech filtrations, with k-skeletons of size at most $n\left(\frac{2}{\varepsilon}\right)^{O(k\Upsilon_{7t})}$, for the range of scales $[0,t]$, where $\epsilon \in (0,1)$.*

Proof. For Rips filtrations, this follows directly by applying the algorithm of [She13, Section 10] on a net-forest of scale t. For a root node u, $w := \{u \cup \mathrm{Rel}(u)\}$ successfully captures all edges of length at most $5t > t$ between a point $p \in P_u$ and $q \in P_w$, which is the main requirement of the algorithm of [She13, Section 10], thereby yielding a tower of size at most $n(\frac{2}{\varepsilon})^{O(k\Upsilon_{7t})}$.

To prove the same for Čech filtrations, we use our net-forests to construct well-separated simplicial decomposition (Chapter 7) first. We give the details in the rest of the section. ■

Well-separated simplicial decomposition For some $t > 0$, we define a *t-restricted (ε, k)-WSSD* to be a collection of ε-well-separated tuples such that each k-simplex on P whose minimum enclosing ball has radius at most t, is covered by some tuple.

Construction of the t-restricted WSSD

We describe the algorithm to construct the t-restricted (ε, k)-WSSD and prove its correctness and runtime. We heavily rely on the notations and results presented in Chapter 7.

The algorithm proceeds iteratively in dimension; for $k = 1$, we construct a $(2t)$-restricted $\varepsilon/2$-WSPD using the algorithm from Sub-section 8.2.1. To construct Γ_{k+1} from Γ_k, we iterate over the tuples $\gamma \in \Gamma_k$. We use the scheme of Chapter 7, computing an approximate meb of γ and then exploring ancestors of v_0 and their descendants at appropriate levels. The only complication arises when the algorithm requests for an ancestor higher than root of the tree of v_0. In such a case, our algorithm uses the root as the ancestor. In the following lemma, we will show that with this approach, we still cover all simplices with minimum enclosing radius of at most t.

Lemma 8.2.4. *The algorithm computes a t-restricted (ε, k)-WSSD.*

Proof. We show by induction that with the modified ancestor search, we still cover all simplices with meb radius at most t. For $k = 1$, the correctness of the algorithm follows from Theorem 8.2.1. Let Γ_{k-1} cover all $(k-1)$-simplices γ on P which satisfy $\mathrm{rad}(\gamma) \le t$. Consider any k-simplex $\sigma = (m_0, \ldots, m_k)$ with $\mathrm{rad}(\sigma) \le t$ and let $\sigma' := \sigma \setminus \{m_k\}$. Since σ' is a $(k-1)$-simplex and $\mathrm{rad}(\sigma') \le \mathrm{rad}(\sigma) \le t$, it is covered by some k-tuple $\gamma = (v_0, \ldots, v_{k-1}) \in \Gamma_{k-1}$. To prove correctness, when our algorithm reaches tuple γ, it should produce a $(k+1)$-tuple (γ, x) such that $m_k \in P_x$ which implies that the simplex σ is covered by the $(k+1)$-tuple (γ, x).

When handling γ, the algorithm searches for an ancestor of v_0 (say) at an appropriate scale. If this ancestor is found within the tree containing v_0 in the net-forest, the arguments from Lemma 7.1.10 carry over to ensure that a suitable x is found. So let us assume that the algorithm chooses the root of the tree of the net-forest that v_0 lies in; call that root node a_0. The algorithm considers all nodes in $\mathrm{Rel}(a_0)$ and creates new tuples with their descendants. Moreover, let a' denote the root of its tree which the point m_k. To prove our claim, it suffices to show that $a' \in \mathrm{Rel}(a_0)$. Since $\mathrm{rad}(\sigma) \leq t$, the distance of m_0 and m_k is at most $2t$. Moreover, the distance of m_0 to rep_{a_0} is at most t, because the representatives of the roots form a (t,t)-net. The same holds for m_k and a'. Using triangle inequality, the distance of rep_{a_0} and $\mathrm{rep}_{a'}$ is at most $4t$, which implies that $a' \in \mathrm{Rel}(a_0)$. ■

Lemma 8.2.5. *The size of the computed t-restricted (ε, k)-WSSD Γ_k is $n(\frac{2}{\varepsilon})^{O(\Upsilon_{7t}k)}$.*

Proof. The proof of Lemma 7.1.11 carries over directly – indeed, we can replace all occurrences of Υ by Υ_{7t}. This comes from the fact that a node u has at most $14^{\Upsilon_{7t}}$ nodes in $\mathrm{Rel}(u)$, and for any node in $\mathrm{Rel}(u)$ we search descendants of a level of at most $O(\log(2/\varepsilon))$ smaller than u (see the proof of Lemma 7.1.11 for details). Since every node in the net-forest has at most $2^{O(\Upsilon_t)}$ children, we create at most $14^{\Upsilon_{7t}} \left(\frac{2}{\varepsilon}\right)^{O(\Upsilon_t)} = \left(\frac{2}{\varepsilon}\right)^{O(\Upsilon_{7t})}$ tuples in Γ_k from a tuple in Γ_{k-1}. With that, the bound can be proved by induction. ■

Lemma 8.2.6. *Computing a t-restricted (ε, k)-WSSD takes time $nd(2/\varepsilon)^{O(\Upsilon_{7t}k)}$ after computing the net-forest at scale t.*

Proof. The proof is analogous to Lemma 7.1.13, plugging in the running time for t-restricted ε-WSPD from Theorem 8.2.1 and the size bound from Lemma 8.2.5. ■

Approximation tower for Čech filtration We use the scheme of Section 7.2 to construct the $(1+\varepsilon)$-approximate filtration on the t-restricted WSSD. The original construction works without modification. Using the notation from Section 7.2, for any WST $\sigma = (v_0, \ldots, v_k)$ with $\ell(v_i) \leq h$ for all i, we add $\sigma' = (\mathrm{vcell}(v_0, h), \ldots, \mathrm{vcell}(v_k, h))$ to $\mathcal{A}_\alpha$ if $\mathrm{rad}(\sigma') \leq \theta_\Delta$. The only potential problem with the t-restricted case is that such a vcell() might be a node higher than a root of the net-forest. This cannot happen, however, since h is chosen such that $\frac{2\tau}{\tau-1}\tau^h \leq \frac{\varepsilon}{7}\alpha$. Since $\alpha \leq t$ and $\varepsilon \leq 1$, we have that $h < \lfloor \log_\tau \frac{\tau-1}{2\tau} t \rfloor = \ell(u)$ for any root u in the net-forest, by construction. This gives rise to an approximation tower $(\mathcal{A}_\alpha)_{\alpha \in [0,t]}$ of size $n(\frac{2}{\varepsilon})^{O(\Upsilon_{7t}^2)}$ whose persistence module is an $(1+\varepsilon)$-approximation of the *truncated Čech filtration* $(\mathcal{C}_\alpha)_{\alpha \in [0,t]}$.

8.3 Discussion

We presented an algorithm to construct the relevant part of the net-tree without exploring larger scales, and avoid unfavorable properties of large scales. We also presented a few applications based on the net-forest. The main technical contribution is the adaptation of locality sensitive hashing to solve the all-near-neighbors problem. This technique is however exclusive to Euclidean spaces, which is also the domain of our results. An efficient alternative to compute near-neighbors in general metric spaces would allow us to adapt our construction to cover more scenarios and it is an agenda for future exploration.

Chapter 9

Conclusion

We briefly summarize some of the results obtained in this thesis and compare them. For simplicity, we call the approximation schemes from Chapters 4, 5, 6 and 7 as the *permutahedral*-scheme, the *barycentric*-scheme, the *digital*-scheme and the *wssd*-scheme, respectively. For a point set $P \subset \mathbb{R}^d$ with doubling dimension Υ and $n := |P|$, the qualities of the schemes can be summarized in Table 9.1. To have a uniform comparison, we have interpreted the schemes as approximations of Čech filtrations.

scheme	approximation ratio	size of k-skeleton
Permutahedral	$6\sqrt[4]{2}(d+1)$	$\min\left(n2^{O(d\log d)}, n2^{O(d\log k)}\log\Delta\right)$
Barycentric	$2^{5/4}d^{1/4}$	$n2^{O(d\log k)}$
Digital	$(1+\varepsilon)$	$n(1/\varepsilon)^{O(d)}2^{O(dk)}\log\Delta$
WSSD	$(1+\varepsilon)$	$n(1/\varepsilon)^{O(\Upsilon^2)}$

Table 9.1: Comparison of approximation schemes: we have $0 < \varepsilon \leq 1$ and Δ denotes the spread of P.

It is clear that the digital and wssd schemes have strictly better approximation factors than the permutahedral and the barycentric schemes. On the other hand, the size complexity of digital is strictly (exponentially) worse than the permutahedral and barycentric schemes. In this way, we achieve a trade-off between tower size and approximation quality. Some further observations are:

- Among the permutahedral and barycentric schemes, the latter is strictly better than the former, both in approximation quality and size. Although, the permutahedral scheme is a little simpler to describe.
- While the digital scheme de-couples the skeleton size from the exponent of $(1/\varepsilon)$, it adds a factor of spread compared to the wssd scheme. Therefore, in general neither is clearly superior than the other. On the other hand, the digital scheme is much simpler to describe and implement.
- When $\Upsilon \ll d$, the wssd scheme fares better than all other schemes.

Future work

In the discussion sections of Chapters 4–8, we discussed some possibilities for future work. Here, we mention some additional research questions which naturally arise out of this thesis.

In Chapter 4 we gave a lower bound result for approximating Čech filtrations, for a very fine approximation quality. There is no counterpart of this result for general metric spaces. This could be an avenue for future work. Better lower bounds for Euclidean point sets are also desirable, especially when the target approximation quality is a constant, and hence is more reasonable than our result.

The digital scheme is exclusive to Euclidean spaces. It is unclear what would be the counterpart in general metric spaces. Perhaps in special cases such as manifolds, it may be possible to arrive at a similar discretization scheme, which could be an interesting research question.

In all our approximation schemes, we approximate the geometry of the Rips or Čech complex in some sense. Further, for efficient computation of persistence, we have to restrict ourselves to simplicial towers. Both these constraints restrict the choices in designing approximation towers. A simple example stands out in the barycentric scheme from Chapter 5: in order to establish a simplicial tower, we use the barycentric subdivision, which leads to a lot of extra triangulation. Efficient algorithms to compute persistence for chain maps could possibly relax the constraint of sticking to geometric approximations to some extent.

We have only considered simplicial persistent homology in this thesis. There is also the concept of cubical complexes and the associated cubical persistent homology, which is more suited for some applications. Cubical complexes avoid the problem of over-triangulation and are therefore much smaller in size. Extending our techniques to cubical complexes is another attractive research question.

The theoretical results presented in this thesis are geared towards providing practical improvements in TDA. For two of our approximation schemes, the approximation factor depends on the ambient dimension, and gets progressively worse as the dimension increases. It is still possible that the approximation factor and runtimes fare much better in practice. It may be worthy to implement the schemes and to observe the amount of discrepancy between theory and practice. The digitization method needs further optimizations before a practical implementation can be realized. As of now, there is no known implementation of the net-tree, which is a roadblock in constructing WSSDs. There is an implementation of the *cover-tree* [BKL06], that has been shown to be related to the net-tree [JS16]. It may be worthy to see if our techniques adapt to the cover-tree.

Bibliography

[AHP12] M. Abam and S. Har-Peled. New Constructions of SSPDs and their Applications. *Computational Geometry Theory and Applications*, 45(5-6):200–214, July 2012.

[ARC14] A. Adcock, D. Rubin, and G. Carlsson. Classification of Hepatic Lesions using the Matching Metric. *Computer Vision and Image Understanding*, 121:36–42, 2014.

[Ass83] P. Assouad. Plongements Lipschitziens dans $\mathbb{R}^n$. *Bulletin de la Societe Mathematique de France*, 111:429–448, 1983.

[BA09] J. Baek and A.Adams. Some Useful Properties of the Permutohedral Lattice for Gaussian Filtering. Technical report, Stanford University, 2009.

[Bam54] R. Bambah. On Lattice Coverings by Spheres. *Proceedings of the National Institute of Sciences of India*, 20:25–52, 1954.

[BC03] M. Bădoiu and K. Clarkson. Smaller Core-sets for Balls. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2003.

[BC08] M. Bădoiu and K. Clarkson. Optimal Core-sets for Balls. *Computational Geometry: Theory and Applications*, 40:14–22, 2008.

[BDM15] J.-D. Boissonnat, T.K. Dey, and C. Maria. The Compressed Annotation Matrix: an Efficient Data Structure for Computing Persistent Cohomology. *Algorithmica*, 73(3):607–619, 2015.

[BdSS15] P. Bubenik, V. de Silva, and J. Scott. Metrics for Generalized Persistence Modules. *Foundations of Computational Mathematics*, 15(6):1501–1531, 2015.

[BEK10] P. Bendich, H. Edelsbrunner, and M. Kerber. Computing Robustness and Persistence for Images. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1251–1260, 2010.

[Ber] Bernoulli's Inequality. `https://goo.gl/GZXpBE`.

[Bjö95] A. Björner. Topological methods. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics (Volume 2)*, pages 1819–1872. MIT Press, 1995.

[Bjö03] A. Björner. Nerves, Fibers and Homotopy groups. *Journal of Combinatorial Theory Series A*, 102(1):88–93, 2003.

[BKL06] A. Beygelzimer, S. Kakade, and J. Langford. Cover Trees for Nearest Neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 97–104, 2006.

[BKR14] U. Bauer, M. Kerber, and J. Reininghaus. Clear and Compress: Computing Persistent Homology in Chunks. *Topological Methods in Data Analysis and Visualization III: Theory, Algorithms, and Applications*, pages 103–117, 2014.

[BKRW17] U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner. PHAT - Persistent Homology Algorithms Toolbox. *Journal of Symbolic Computation*, 78(C):76–90, January 2017.

[BOOC16] T. Bonis, M. Ovsjanikov, S. Oudot, and F. Chazal. Persistence-based Pooling for Shape Pose Recognition. In *Proceedings of the 6th International Workshop on Computational Topology in Image Context (CTIC)*, pages 19–29, 2016.

[Bor48] K. Borsuk. On the Imbedding of Systems of Compacta in Simplicial Complexes. *Fundamenta Mathematicae*, pages 217–234, 1948.

[Bou85] J. Bourgain. On Lipschitz Embedding of Finite Metric Spaces in Hilbert Space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985.

[BS14] P. Bubenik and J.A. Scott. Categorification of Persistent Homology. *Discrete & Computational Geometry*, 51(3):600–627, 2014.

[BS15] M. Botnan and G. Spreemann. Approximating Persistent Homology in Euclidean Space through Collapses. *Applied Algebra in Engineering, Communication and Computing*, 26(1-2):73–101, 2015.

[Car09] G. Carlsson. Topology and Data. *Bulletin of the American Mathematical Society*, 46:255–308, 2009.

[CBK09] M. Chung, P. Bubenik, and P. Kim. Persistence Diagrams of Cortical Surface Data. In *Proceedings of the 21st International Conference on Information Processing in Medical Imaging*, pages 386–397, 2009.

[CCR13] J. Chan, G. Carlsson, and R. Rabadan. Topology of Viral Evolution. *Proceedings of the Natural Academy of Sciences*, 110(46):18566–18571, 2013.

[CCSG$^+$09] F. Chazal, D. Cohen-Steiner, M. Glisse, L. Guibas, and S. Oudot. Proximity of Persistence Modules and their Diagrams. In *ACM Symposium on Computational Geometry (SoCG)*, pages 237–246, 2009.

[CdS10] G. Carlsson and V. de Silva. Zigzag Persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010.

[CdSGO16] F. Chazal, V. de Silva, M. Glisse, and S. Oudot. *The Structure and Stability of Persistence Modules.* Springer International Publishing, 2016.

[CdSM09] G. Carlsson, V. de Silva, and D. Morozov. Zigzag Persistent Homology and Real-valued Functions. In *Proceedings of the 25th Annual Symposium on Computational Geometry (SoCG)*, pages 247–256, 2009.

[CET15] T. Cao, H. Edelsbrunner, and T. Tan. Triangulations from Topologically Correct Digital Voronoi Diagrams. *Computational Geometry: Theory and Applications*, 48(7):507–519, 2015.

[CGOS13] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Persistence-based Clustering in Riemannian Manifolds. *Journal of ACM*, 60(6):41:1–41:38, 2013.

[CJS15] N. Cavanna, M. Jahanseir, and D. Sheehy. A Geometric Perspective on Sparse Filtrations. In *Proceedings of the 27th Canadian Conference on Computational Geometry (CCCG)*, pages 116–121, 2015.

[CK95] P. Callahan and S. Kosaraju. A Decomposition of Multidimensional Point Sets with Applications to k-Nearest Neighbors and n-body Potential Fields. *Journal of the ACM*, 42(67–90), 1995.

[CK11] C. Chen and M. Kerber. Persistent Homology Computation with a Twist. In *Proceedings of the 27th European Workshop on Computational Geometry*, 2011.

[CK15] A. Choudhary and M. Kerber. Local Doubling Dimension of Point Sets. In *Proceedings of the 27th Canadian Conference on Computational Geometry (CCCG)*, pages 156–164, 2015.

[CKRa] A. Choudhary, M. Kerber, and S. Raghavendra. Improved Topological Approximations by Sampling. Manuscript.

[CKRb] A. Choudhary, M. Kerber, and S. Raghavendra. Polynomial-Sized Topological Approximations using the Permutahedron (extended version). Accepted to Discrete and Computational Geometry.

[CKR16] A. Choudhary, M. Kerber, and S. Raghavendra. Polynomial-Sized Topological Approximations using the Permutahedron. In *Proceedings of the 32nd International Symposium on Computational Geometry (SoCG)*, pages 31:1–31:16, 2016.

[CKR17] A. Choudhary, M. Kerber, and S. Raghavendra. Improved Approximate Rips Filtrations with Shifted Integer Lattices. In *Proceedings of the 25th Annual European Symposium on Algorithms (ESA)*, pages 28:1–28:13, 2017.

[CKS] A. Choudhary, M. Kerber, and R. Sharathkumar. Approximate Čech Complexes in Low and High Dimensions. Manuscript.

[CSB87] J. H. Conway, N. J. A. Sloane, and E. Bannai. *Sphere-packings, Lattices and Groups*. Springer-Verlag, 1987.

[CSEH07] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of Persistence Diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[CSEHM10] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz Functions have L_p-stable Persistence. *Foundations of Computational Mathematics*, 10(2):127–139, 2010.

[CZ05] G. Carlsson and A. Zomorodian. Computing Persistent Homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[DFW14] T.K. Dey, F. Fan, and Y. Wang. Computing Topological Persistence for Simplicial Maps. In *Proceedings of the 30th Annual Symposium on Computational Geometry (SoCG)*, pages 345–354, 2014.

[DIIM04] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni. Locality-Sensitive Hashing Scheme based on P-stable Distributions. In *Proceedings of the 20th Annual Symposium on Computational Geometry (SoCG)*, pages 253–262, 2004.

[DKRS03] I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating CVP to Within Almost-Polynomial Factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003.

[dSG06] V. de Silva and R. Ghrist. Coordinate-free Coverage in Sensor Networks with Controlled Boundaries via Homology. *International Journal of Robotics Research*, 25(12):1205–1222, 2006.

[dSMVJ11a] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Dualities in Persistent (Co)homology. *Inverse Problems*, 27(12), 2011.

[dSMVJ11b] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Persistent Cohomology and Circular Coordinates. *Discrete & Computational Geometry*, 45(4):737–759, 2011.

[DSW16] T.K. Dey, D. Shi, and Y. Wang. SimBa: An Efficient Tool for Approximating Rips-Filtration Persistence via Simplicial Batch-Collapse. In *24th Annual European Symposium on Algorithms (ESA)*, pages 35:1–35:16, 2016.

[EH10] H. Edelsbrunner and J. Harer. *Computational Topology - An Introduction.* American Mathematical Society, 2010.

[EK12] H. Edelsbrunner and M. Kerber. Dual Complexes of Cubical Subdivisions of $\mathbb{R}^n$. *Discrete & Computational Geometry*, 47(2):393–414, 2012.

[ELZ02] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological Persistence and Simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[EM90] H. Edelsbrunner and E.P. Mücke. Simulation of Simplicity: a Technique to Cope with Degenerate Cases in Geometric Algorithms. *ACM Transactions on Graphics*, pages 66–104, 1990.

[EW17] H. Edelsbrunner and H. Wagner. Topological Data Analysis with Bregman Divergences. In *33rd International Symposium on Computational Geometry (SoCG)*, volume 77, pages 39:1–39:16, 2017.

[GKL03] A. Gupta, R. Krauthgamer, and J.R. Lee. Bounded Geometries, Fractals, and Low-Distortion Embeddings. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.

[GOT17] J.E. Goodman, J. O'Rourke, and C.D. Tóth, editors. *Handbook of Computational Geometry.* CRC Press, 2017.

[Hal05] T. Hales. A Proof of the Kepler Conjecture. *Annals of Mathematics*, 162:1065–1185, 2005.

[Hat02] A. Hatcher. *Algebraic Topology.* Cambridge University Press, 2002.

[Hel23] E. Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkte. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32:175–176, 1923.

[HMOS10] B. Hudson, G. Miller, S. Oudot, and D. Sheehy. Topological Inference via Meshing. In *Proceedings of the 26th Annual Symposium on Computational Geometry (SoCG)*, pages 277–286, 2010.

[HMR09] D. Horak, S. Maletić, and M. Rajković. Persistent Homology of Complex Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):03–34, 2009.

[HP11] S. Har-Peled. *Geometric Approximation Algorithms.* American Mathematical Society, 2011.

[HPM06] S. Har-Peled and M Mendel. Fast Construction of Nets in Low-dimensional Metrics, and their Applications. *SIAM Journal of Computing*, 35(5):150–158, 2006.

[IM98] P. Indyk and R. Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[JLS86] W.B. Johnson, J. Lindenstrauss, and G. Schechtman. Extensions of Lipschitz Maps into Banach Spaces. *Israel Journal of Mathematics*, 54(2):129–138, 1986.

[JRVM$^+$17] M. J. Jimenez, M. Rucco, P. Vicente-Munuera, P. Gómez-Gálvez, and L. M. Escudero. Topological Data Analysis for Self-organization of Biological Tissues. In *Proceedings of the Eighteenth International Workshop on Combinatorial Image Analysis (IWCIA)*, pages 229–242, 2017.

[JS16] M. Jahanseir and D. Sheehy. Transforming Hierarchical Trees on Metric Spaces. In *CG Week: Young Researchers Forum*, 2016.

[Jun01] H. Jung. Über die kleinste Kugel, die eine räumliche Figur einschliesst. *Journal für die reine und angewandte Mathematik*, 123:241–257, 1901.

[KFH16] G. Kusano, K. Fukumizu, and Y. Hiraoka. Persistence Weighted Gaussian Kernel for Topological Data Analysis. In *Proceedings of the Thirty-third International Conference on Machine Learning (ICML)*, volume 48, pages 2004–2013, 2016.

[KM95] S. Khuller and Y. Matias. A Simple Randomized Sieve Algorithm for the Closest-Pair Problem. *Information and Computation*, 118(1):34 – 37, 1995.

[KR15] M. Kerber and S. Raghvendra. Approximation and Streaming Algorithms for Projective Clustering via Random Projections. In *Proceedings of the 27th Canadian Conference on Computational Geometry (CCCG)*, 2015.

[KS13] M. Kerber and R. Sharathkumar. Approximate Čech Complex in Low and High Dimensions. In *Algorithms and Computation - 24th International Symposium (ISAAC)*, pages 666–676, 2013.

[KS17] M. Kerber and H. Schreiber. Barcodes of Towers and a Streaming Algorithm for Persistent Homology. In *Proceedings of 33rd International Symposium on Computational Geometry (SoCG)*, pages 57:1–57:15, 2017.

[LS93] L. Lovász and M. Simonovits. Random Walks in a Convex Body and an Improved Volume Algorithm. *Random Structures and Algorithms*, 4(4):359–412, 1993.

[Mat90] J. Matoušek. Bi-Lipschitz Embeddings into Low-dimensional Euclidean Spaces. *Commentationes Mathematicae Universitatis Carolinae*, 1990.

[MMv11] N. Milosavljević, D. Morozov, and P. Škraba. Zigzag Persistent Homology in Matrix Multiplication Time. In *Proceedings of the 27th Annual Symposium on Computational Geometry (SoCG)*, pages 216–225, 2011.

[MS12] R. MacPherson and B. Schweinhart. Measuring Shape with Topology. *Journal of Mathematical Physics*, 53(7):1–13, 2012.

[Mun84] J.R. Munkres. *Elements of Algebraic Topology.* Westview Press, 1984.

[NHH$^+$15] T. Nakamura, Y. Hiraoka, A. Hirata, E. Escolar, and Y. Nishiura. Persistent Homology and Many-body Atomic Structure for Medium-range Order in the Glass. *Nanotechnology*, 26(30), 2015.

[Per] The Permutahedron, Wikipedia. `https://goo.gl/bb9MRY`.

[PET$^+$14] G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. Hellyer, and F. Vaccarino. Homological Scaffolds of Brain Functional Networks. *Journal of the Royal Society Interface*, 11(101), 2014.

[PEvdW$^+$16] P. Pranav, H. Edelsbrunner, R. van de Weygaert, G. Vegter, M. Kerber, B. Jones, and M. Wintraecken. The Topology of the Cosmic Web in Terms of Persistent Betti Numbers. *Monthly Notices of the Royal Astronomical Society*, 465:4281–4310, 2016.

[PM13] E. Park and D. Mount. Output-Sensitive Well-separated Pair Decompositions for Dynamic Point Sets. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 354–363, 2013.

[PSDV13] G. Petri, M. Scolamiero, I. Donato, and F. Vaccarino. Topological Strata of Weighted Complex Networks. *PLoS One*, 2013.

[RD69] B.C. Rennie and A.J. Dobson. On Stirling Numbers of the Second Kind. *Journal of Combinatorial Theory*, 7(2):116–121, 1969.

[RNS$^+$17] M. Reimann, M. Nolte, M. Scolamiero, K. Turner, R. Perin, G. Chindemi, P. Dłotko, R. Levi, K. Hess, and H. Markram. Cliques of Neurons Bound into Cavities Provide a Missing Link between Structure and Function. *Frontiers in Computational Neuroscience*, 11:48, 2017.

[She13] D. Sheehy. Linear-size Approximations to the Vietoris-Rips Filtration. *Discrete & Computational Geometry*, 49(4):778–796, 2013.

[She14] D. Sheehy. The Persistent Homology of Distance Functions Under Random Projection. In *Proceedings of the 30th Annual Symposium on Computational Geometry (SoCG)*, pages 328:328–328:334, 2014.

[Smi07] M. H. M. Smid. The Well-Separated Pair Decomposition and its Applications. In *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC, 2007.

[Sou11] T. Sousbie. The Persistent Cosmic Web and its Filamentary Structure I: Theory and Implementation. *Monthly notices of the Royal Astronomical Society*, 414(1):350–383, 2011.

[SPK11] T. Sousbie, C. Pichon, and H. Kawahara. The Persistent Cosmic Web and its Filamentary Structure II: Illustrations. *Monthly Notices of the Royal Astronomical Society*, 414(1):384–403, 2011.

[STR$^+$17] M. Saadatfar, H. Takeuchi, V. Robins, N. Francois, and Y. Hiraoka. Pore Configuration Landscape of Granular Crystallization. *Nature Communications*, 8, 2017.

[Tal04] K. Talwar. Bypassing the Embedding: Algorithms for Low Dimensional Metrics. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 281–290, 2004.

[Var98] K.R. Varadarajan. A Divide-and-Conquer Algorithm for Min-cost Perfect Matching in the Plane. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331, 1998.

[Wal81] J.W. Walker. Homotopy Type and Euler Characteristic of Partially Ordered Sets. *European Journal of Combinatorics*, 2(4):373 – 384, 1981.

[Zie95] G.M. Ziegler. *Lectures on Polytopes*. Springer-Verlag, 1995.

Appendix A

Strong Interleaving for Barycentric Scheme

Recall that in the *barycentric scheme* (Chapter 5), we build the approximation tower over the set of scales $I := \{\alpha_s = 2^s \mid s \in \mathbb{Z}\}$. The tower $(\mathcal{X}_\alpha)_{\alpha \in I}$ connected with the simplicial map $\tilde{g}$ can be extended to the set of scales $\alpha \geq 0$ with simple modifications:

- for $\alpha \in I$, we define $\mathcal{X}_\alpha$ in the usual manner. The map $\tilde{g}$ stays the same as before for complexes at such scales.
- for all $\alpha \in [\alpha_s, \alpha_{s+1})$, we set $\mathcal{X}_\alpha = \mathcal{X}_{\alpha_s}$, for any $\alpha_s \in I$. That means, the complex stays the same in the interval between any two scales of I, so we define $\tilde{g}$ as the identity within this interval.

This gives rise to the tower $(\mathcal{X}_\alpha)_{\alpha \geq 0}$, that is connected with the simplicial map $\tilde{g}$. This modification helps in improving the interleaving with the Rips persistence module.

First, we extend the acyclic carriers C_1 and C_2 from before to the new case:

- $C_1^\alpha : \mathcal{R}_\alpha^\infty \to \mathcal{X}_{4\alpha}, \alpha > 0$: we define C_1 as before, simply changing the scales in the definition. It is straightforward to see that C_1 is still a well-defined acyclic carrier.
- $C_2^\alpha : \mathcal{X}_\alpha \to \mathcal{R}_\alpha^\infty, \alpha \geq 0$: this stays the same as before. It is simple to check that C_2 is still a well-defined acyclic carrier.

These gives rise to chain maps between the chain-complexes:

$$c_1^\alpha : \mathcal{C}_*(\mathcal{R}_\alpha^\infty) \to \mathcal{C}_*(\mathcal{X}_{4\alpha}) \qquad \text{and} \qquad c_2^\alpha : \mathcal{C}_*(\mathcal{X}_\alpha) \to \mathcal{C}_*(\mathcal{R}_\alpha^\infty),$$

using the acyclic carrier theorem as before (Theorem 2.4.3).

Lemma A.0.1. *The diagram*

$$\begin{tikzcd}
\mathcal{C}_*(\mathcal{R}_\alpha^\infty) \arrow[r, "inc"] \arrow[dr, "c_1"] & \mathcal{C}_*(\mathcal{R}_{\alpha'}^\infty) \arrow[dr, "c_1"] & \\
& \mathcal{C}_*(\mathcal{X}_{4\alpha}) \arrow[r, "\tilde{g}"] & \mathcal{C}_*(\mathcal{X}_{4\alpha'})
\end{tikzcd} \tag{A.1}$$

commutes on the homology level, for all $0 \leq \alpha \leq \alpha'$.

Proof. Consider the acyclic carrier $C_1 \circ inc : \mathcal{R}_\alpha^\infty \to \mathcal{X}_{4\alpha'}$. It is simple to verify that this carrier carries both $c_1 \circ inc$ and $\tilde{g} \circ c_1$, so the induced diagram on the homology groups commutes, from Theorem 2.4.3. ■

Lemma A.0.2. *The diagram*

$$\begin{tikzcd}
 & \mathcal{C}_*(\mathcal{R}^\infty_\alpha) \arrow[r, "inc"] & \mathcal{C}_*(\mathcal{R}^\infty_{\alpha'}) \\
\mathcal{C}_*(\mathcal{X}_\alpha) \arrow[ur, "c_2"] \arrow[r, "\tilde{g}"] & \mathcal{C}_*(\mathcal{X}_{\alpha'}) \arrow[ur, "c_2"] &
\end{tikzcd} \tag{A.2}$$

commutes on the homology level, for all $0 \leq \alpha \leq \alpha'$.

Proof. We construct an acyclic carrier $D : \mathcal{X}_\alpha \to \mathcal{R}^\infty_{\alpha'}$ which carries $inc \circ c_2$ and $c_2 \circ \tilde{g}$, thereby proving the claim (Theorem 2.4.3).

Consider any simplex $\sigma = (e_0 \subset \ldots \subset e_k) \in \mathcal{X}_\alpha$. We set $D(\sigma)$ as the simplex on the set of input points of P, which lie in the Voronoi regions of the vertices of $g(e_k)$. By triangle inequality, $D(\sigma)$ is a simplex of $\mathcal{R}^\infty_{\alpha'}$, so that D is a well-defined acyclic carrier. It is straightforward to verify that D carries both $c_2 \circ \tilde{g}$ and $inc \circ c_2$. ■

Lemma A.0.3. *The diagram*

$$\begin{tikzcd}
 & \mathcal{C}_*(\mathcal{R}^\infty_\alpha) \arrow[r, "inc"] & \mathcal{C}_*(\mathcal{R}^\infty_{\alpha'}) \arrow[dr, "c_1"] & \\
\mathcal{C}_*(\mathcal{X}_\alpha) \arrow[ur, "c_2"] \arrow[rrr, "\tilde{g}"] & & & \mathcal{C}_*(\mathcal{X}_{4\alpha'})
\end{tikzcd} \tag{A.3}$$

commutes on the homology level, for all $0 \leq \alpha \leq \alpha'$.

Proof. The diagram is essentially the same as the lower triangle of Diagram 5.1, with a change in the scales. As a result, the proof of Lemma 5.1.7 also applies for our claim directly. ■

Lemma A.0.4. *The diagram*

$$\begin{tikzcd}
\mathcal{C}_*(\mathcal{R}^\infty_\alpha) \arrow[rrr, "inc"] \arrow[dr, "c_1"] & & & \mathcal{C}_*(\mathcal{R}^\infty_{4\alpha'}) \\
 & \mathcal{C}_*(\mathcal{X}_{4\alpha}) \arrow[r, "\tilde{g}"] & \mathcal{C}_*(\mathcal{X}_{4\alpha'}) \arrow[ur, "c_2"] &
\end{tikzcd} \tag{A.4}$$

commutes on the homology level, for all $0 \leq \alpha \leq \alpha'$.

Proof. The diagram can be re-interpreted as:

$$\begin{tikzcd}
\mathcal{C}_*(\mathcal{R}^\infty_\alpha) \arrow[rr, "inc"] \arrow[dr, "\tilde{g} \circ c_1"] & & \mathcal{C}_*(\mathcal{R}^\infty_{4\alpha'}) \\
 & \mathcal{C}_*(\mathcal{X}_{4\alpha'}) \arrow[ur, "c_2"] &
\end{tikzcd} \tag{A.5}$$

The modified diagram is essentially the same as the upper triangle of Diagram 5.1, with a change in the scales and a replacement of c_1 with $\tilde{g} \circ c_1$, that is equivalent to the chain map at the scale α'. Hence, the proof of Lemma 5.1.8 also applies for our claim directly. ■

Using lemmas A.0.1, A.0.2, A.0.3, A.0.4, and the scale balancing technique for strongly interleaved persistence modules (Subsection 2.3.3), it follows that

Lemma A.0.5. *The persistence modules* $\left(H(\mathcal{X}_{2\alpha})\right)_{\alpha \geq 0}$ *and* $\left(H(\mathcal{R}^\infty_\alpha)\right)_{\alpha \geq 0}$ *are strongly* 2*-interleaved.*

Appendix B

Curriculum Vitae

Aruni Choudhary

Contact Information
Algorithms and Complexity Department,
Max-Planck-Institut für Informatik,
Campus E-1 4, 66123 Saarbrücken, Germany
Email: aruni.choudhary@mpi-inf.mpg.de
Web: http://mpi-inf.mpg.de/~achoudha/

Personal Information
Date of birth: 21 December 1989
Nationality: Indian

Educational Background

April 2014 - Present	PhD student, Max Planck Institute for Informatics and Saarland University. Advisors: Kurt Mehlhorn, Michael Kerber.
July 2007 - May 2012	Bachelor and Master of Technology in Computer Science and Engineering, Indian Institute of Technology Kharagpur. Advisors: Pallab Dasgupta, Susmita-Sur Kolay.
March 1993 - May 2007	CBSE Higher secondary, Jamshedpur Public School.

Positions

October 2013 - Present	Member, Saarbrücken Graduate School of Computer Science.
June 2012 - August 2013	Member of Technical Staff, Oracle India, Bengaluru.
May 2010 - July 2010	Intern, Processor Architecture Laboratory, École Polytechnique Fédérale de Lausanne. Host: Paolo Ienne.

Teaching

Saarland University	Tutor for Algorithms and Data Structures, 2014, 2016.
IIT Kharagpur	Tutor for Discrete Structures, 2011. Teaching assistant for Programming & Data Structures, 2012.

Publications and Articles

1. A. Choudhary and A. Ghosh. Delaunay Triangulations of Lattices obtained by Diagonally Distorting the Integer Grid. Manuscript, 2017.

2. A. Choudhary, S. Kachanovich and M. Wintraecken. Coxeter Triangulations have good Quality. In submission, 2017.

3. A. Choudhary, M. Kerber, and S. Raghvendra. Polynomial-sized Topological Approximations using the Permutahedron. *Discrete and Computational Geometry*, November 2017.

   Extended abstract in *32nd International Symposium on Computational Geometry (SoCG)*, pages 31:1–31:16, 2016.

4. A. Choudhary, M. Kerber, and S. Raghvendra. Improved Approximate Rips Filtrations with Shifted Integer Lattices. Proceedings of the *European Symposium on Algorithms (ESA)*, pages 28:1–28:13, 2017.

5. A. Choudhary, M. Kerber, and S. Raghvendra. Improved Topological Approximations by Sampling. Manuscript.

6. A. Choudhary, M. Kerber, and S. Raghvendra. Local Structures for Approximating the Rips Filtration. Preliminary version presented at *Young Researchers Forum, Symposium on Computational Geometry (SoCG)*, 2016.

7. A. Choudhary and M. Kerber. Local Doubling Dimension of Point Sets. Proceedings of the *27th Canadian Conference on Computational Geometry (CCCG)*, pages 156–164, 2015.

8. A. Choudhary, M. Kerber, and R. Sharathkumar. Approximate Čech Complexes in Low and High Dimensions. Manuscript, 2014.

9. A. Choudhary. Floorplanning of 3D Integrated Circuits. Master's thesis, Indian Institute of Technology Kharagpur, 2012.

Saarbrücken, Datum:

Aruni Choudhary
(Unterschrift)