Computational Geometry: Young Researchers Forum 2022

— book of abstracts —

This volume contains the abstracts of presentations given at “Computational Geometry: Young Researchers Forum” (CG:YRF), a satellite event of the 38rd International Symposium on Computational Geometry, held in Berlin, Germany, on June 7-10, 2022.

The CG:YRF program committee consisted of:
- Peyman Afshani, Aarhus University, Denmark
- Sergio Cabello, University of Ljubljana, Slovenia
- Hsien-Chih Chang, Dartmouth College, USA
- Anne Driemel (chair), Universität Bonn, Germany
- André Nusser, University of Cophenhagen, Denmark
- Zuzana Patáková, Charles University, Czech Republic
- Matias Korman, Mentor Graphics, USA
- Monique Teillaud, INRIA, France

There were 29 abstracts submitted to CG:YRF. Of these, 27 were accepted with revisions after a limited refereeing process to ensure some minimal standards and to check for plausibility. One abstract was withdrawn during the revision process. The abstracts have been made public for the benefit of the community and should be considered preprints rather than formally reviewed papers. Thus, these works are expected to appear in conferences with formal proceedings and/or in journals. Copyrights of the works in this booklet are maintained by their respective authors. More information about the event and about previous and future editions is available online at

http://www.computational-geometry.org/

CG:Young Researchers Forum — Program

Tuesday, 7 June 2022, 11:00–12:30, CG:YRF Session 1A

Tuesday, 7 June 2022, 11:00–12:30, CG:YRF Session 1B

Tuesday, 07. June 2022, 14:00–15:30, CG:YRF Session 2A

Tuesday, 7 June 2022, 14:00–15:30, CG:YRF Session 2B

Wednesday, 8 June 2022, 15:30–17:00, CG:YRF Session 3A

Wednesday, 8 June 2022, 15:30–17:00, CG:YRF Session 3B

Sparsity and output-sensitivity in persistence computation

Ulrich Bauer
Department of Mathematics, Technical University of Munich (TUM), Germany

Talha Bin Masood
Department of Science and Technology, Linköping University, Sweden

Barbara Giunti
Department of Mathematics, Graz University of Technology, Austria

Guillaume Houry
École Polytechnique, France

Michael Kerber
Department of Mathematics, Graz University of Technology, Austria

Abhishek Rathod
Computer Science Department, Purdue University, USA

Persistent homology has proven to be a powerful tool to extract topological information from data [3]. Therefore, understanding the complexity of its computation is an important problem in topological data analysis. Persistent homology can be computed in matrix multiplication time using a divide-and-conquer algorithm [7]. From the point of view of classical complexity, improving upon the $O(n^\omega)$ complexity would need a major theoretical advancement. In practice, the most efficient methods for persistence computation employ special variations of the Gaussian elimination method, and its worst-case complexity is cubic in the number of input simplices. In details, the Standard Persistence Algorithm (SPA) [4] performs left-to-right column additions until the **pivots** (i.e. the row indices of the lowest non-zero elements of the columns in the matrix) are pairwise distinct. It is conceivable that matrix reduction methods that maintain sparsity would be computationally advantageous. In this work, we investigate sparsity preserving methods for persistence computation with the goal of understanding how the sparsification of the matrix can improve efficiency. In general, we show that finding a sequence of column additions that maximally sparsify a matrix is NP-hard. Moreover, even if in some cases sparsification is more efficient this is not true in general. Indeed, we show with some experiments (see Figure 1) that algorithms that (heavily) prioritize sparsification do not automatically perform better than those that do not. Finally, we describe a novel variant of the persistence algorithm that keeps the matrix sparse for well-behaved outputs – leading to output-sensitive complexity bounds for this variant.

Define Sparse-Z2 as: Given a vector W and n vectors $U_1, \dots, U_n$ in $\mathbb{Z}_2^m$, find $a_1, \dots, a_n$ in $\mathbb{Z}_2$ such that $W + a_1U_1 + \cdots + a_nU_n$ has the minimum number of nonzero entries. MaxCut is a classical NP-hard problem [6], and there is a straightforward reduction from MaxCut to Sparse-Z2, which gives us the following proposition:

▶ **Proposition 1.** *Sparse-Z2 is NP-hard.*

In [5], the authors introduce the **exhaustive reduction**, based on the SPA, which attempts to remove nonzero entries in a column even after the pivot has been established. Typically, this increases the sparsity of the matrix. Another, less aggressive way to keep

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

the matrix sparse is the **swap reduction** [8, p. 77], implemented in the PHAT library [1] (see Algorithm 1). It reduces the matrix as in the SPA with the caveat that it exchanges the column it is reducing with a previous, denser one if they have the same pivot. We test both together with the retrospective algorithm (see below, Algorithm 2) over two random filtrations against the SPA in cohomology with the clearing optimisation [2]. The random filtrations are the **Erdős–Rényi filtration**, where the edges are ordered randomly and the higher dimensional simplices are added as soon as possible, and the **Vietoris–Rips filtration**, where, given a dimension d, we build the standard Vietoris-Rips over n points of $[0,1]^d$ drawn uniformly at random.

Algorithm 1 Swap reduction

```
Input: Boundary matrix $\partial$          Output: Column reduced boundary matrix $R$
$R = \partial$
for $j = 1, \dots, m$ do
    if $low\,(R[j]) \neq 0$ then                                  ▷ $low$ is the index of the pivot
        while there exists $j' < j$ with $low\,(R[j']) = low\,(R[j]) \neq 0$ do
            if $size\,(R[j]) < size\,(R[j'])$ then               ▷ $size$ is the # of nonzero entries
                swap column $R[j]$ and $R[j']$
            add column $R[j']$ to column $R[j]$
```

A **bitflip** is the operation of changing a bit from 1 to 0 or vice-versa. Here, we use the number of bitflips as a measure of complexity.

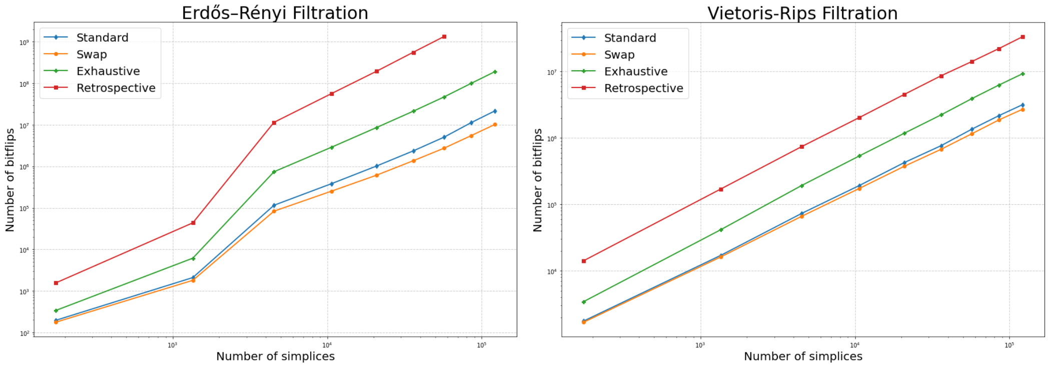


Figure 1 Empirical complexity for computing persistence for random filtrations.

Next, we introduce a new algorithm that tries to keep the matrix sparse during the reduction: the **retrospective algorithm**. To provide a brief description of the algorithm, we first introduce some notation. Let $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$ denote the simplices of a simplicial complex K. Consider a **simplexwise filtration** of K, that is, a nested sequence of subcomplexes, $\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$. In other words, $K_i = K_{i-1} \cup \{\sigma_i\}$. Let d_i denote the dimension of simplex σ_i. The symbol $\mathsf{H}_{d_i}(K_i)$ denotes the d_i-dimensional homology group of K_i and β_i its rank. Furthermore, for $\sigma_i = K_i \setminus K_{i-1}$ and $\sigma_j = K_j \setminus K_{j-1}$, we use $\beta_{i,j}$ to mean $\beta_{d_j}^{i,j}$ since $d_j = \dim(\sigma_j)$. We denote by P the collection of all pairs of indices (i,j) for which the pair (σ_i, σ_j) forms a persistence pair, and by P' the collection of all pairs (i,n) for which σ_i is an essential simplex. We set $\overline{P} = P \cup P'$. Let $R = \partial$ denote the matrix that is reduced. Let $R[k]$ be the column being reduced. An entry in $R[k]$ is **negative** (resp. **positive**) if the corresponding simplex kills (resp. gives birth to) an homological class. Moreover, a positive entry in $R[k]$ is **paired** (resp. **unpaired**) if the corresponding simplex is (not) in a persistence pair (σ_i, σ_j) with $j < k$.

The exhaustive algorithm described in [5] zeros out all the unpaired entries in $R[k]$ using left-to-right additions. The retrospective algorithm is a modification of the exhaustive algorithm, wherein right-to-left additions are used to zero out the nonzero entries in the row of the newly found pivot of $R[k]$. See Algorithm 2 for a pseudocode of the retrospective algorithm. Note that we obtain the exhaustive algorithm by removing its last two lines.

Algorithm 2 Retrospective algorithm for computing persistent homology

```
Procedure MAIN()
    for $k \leftarrow 1, \ldots, n$ do
        REDUCE($k$);

Procedure REDUCE($k$)
    Remove the negative entries from column $R[k]$;
    while $\exists\ \ell < k$ such that the entry $R_{\ell,k}$ in column $R[k]$ is paired do
        Add $R[P[\ell]]$ to $R[k]$;
    if $R[k] \neq 0$ then
        $j$ = PIVOT($R[k]$);    $P[j] \leftarrow k$;
    for columns $i$ such that the entry $R_{j,i}$ in column $R[i]$ is nonzero do
        Add $R[k]$ to $R[i]$;
```

▶ **Theorem 2.** *For Algorithm 2, we have the following bounds on the total number of bitflips:*

$$\#\textit{bitflips} \ \leq \sum_{k=1}^{n} (\beta_k)^2 \ + \ (d+1)\sum_{k=1}^{n}(\beta_k + 1)\ , \tag{1}$$

$$\#\textit{bitflips} \ \leq \sum_{(i,j)\in P}\ \sum_{\ell=i+1}^{j-1} (\beta_{\ell,j} + 1) \ + \ (d+1)\sum_{k=1}^{n}(\beta_k + 1)\ , \tag{2}$$

$$\#\textit{bitflips} \ \leq \sum_{(i,j)\in \overline{P}} (j-i)^2 \ + \ n(d+1)\,. \tag{3}$$

Sketch of proof. We call the addition of column $R[j]$ to column $R[k]$ **backward addition** if $k < j$ and **forward addition** if $k > j$. Clearly, bitflips arise from either forward additions or backward additions. The bitflips from forward additions are bounded by $(d{+}1)\sum_{k=1}^{n}(\beta_k + 1)$, giving the second summand of Eqs. (1)–(2). While the bound for (bitflips from) backward additions in Eq. (1) is obtained by counting backward additions *into* a column, the bound in Eq. (2) accounts for additions *from* a column. The third bound is established by counting the number of bitflips in row i for every positive simplex σ_i. ◀

References

1 Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. *PHAT - Persistent Homology Algorithms Toolbox.* https://bitbucket.org/phat-code/phat/src/master/, 2013–2020.

2 Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *Proceedings 27th European Workshop on Computational Geometry*, 2011.

3 Database of real-world applications of TDA. `https://www.zotero.org/groups/2425412/tda-applications`, 2020.

4 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

5 Herbert Edelsbrunner and Katharina Ölsböck. Tri-partitions and bases of an ordered complex. *Discrete & Computational Geometry*, 64(3):759–775, 2020.

6 Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

7 Nikola Milosavljević, Dmitriy Morozov, and Primož Škraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry*, SoCG '11, page 216–225, New York, NY, USA, 2011. Association for Computing Machinery.

8 Hannah Schreiber. *Algorithmic Aspects in standard and non-standard Persistent Homology.* PhD thesis, Graz University of Technology, 2019.

Reducing Multi-Parameter Flag Filtrations via Edge Collapses

Ángel Javier Alonso ✉
Graz University of Technology, Graz, Austria

Michael Kerber ✉
Graz University of Technology, Graz, Austria

Siddharth Pritam ✉
Shiv Nadar University, Delhi NCR, India

Abstract

Multi-parameter persistence homology studies the shape of data with respect to two or more parameters. A common technique is to consider a filtration on a simplicial complex. Unfortunately, the combinatorial description of such a filtration, and its underlying simplicial complex, can be large, so we are interested in removing some of its components while preserving its topological properties. A particular type of filtrations, that are useful in practice, are flag filtrations, which are entirely determined by a filtration on their 0- and 1-simplices, that is, their underlying graph. Recently, Boissonnat and Pritam [SoCG'20] reduce single-parameter flag filtrations by looking at their graph and removing certain edges. In this work, we consider flag bifiltrations (that is, flag filtrations that depend on two parameters), by introducing an algorithm to remove what we call filtration-dominated edges. We have implemented the algorithm and we report on preliminary experiments.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Topology

Keywords and phrases Computational Topology, Topological Data Analysis, Edge Collapse, Persistent Homology, Multi-parameter Persistent Homology

Funding *Ángel Javier Alonso*: Supported by Austrian Science Fund (FWF) grant P 33765-N.
Michael Kerber: Supported by Austrian Science Fund (FWF) grant P 33765-N.

1 Introduction

Our main object of study are **bifiltered graphs**, that is, we consider a finite graph G and a function f that assigns to each **grade** $p = (p_1, p_2) \in \mathbb{R}^2$ a subgraph G_p of G such that, for any other grade $q = (q_1, q_2) \in \mathbb{R}^2$ with $p_1 \leq q_1$ and $p_2 \leq q_2$ (which we write as $p \leq q$), the subgraph G_p is contained in G_q. See Figure 1 for an illustration. We will refer to both a bifiltered graph and its underlying graph by G, dropping the function f from the notation.

Given a graph G, the **flag complex** $\mathrm{Flag}(G)$ of G is the simplicial complex whose n-simplices are the n-cliques of G. A bifiltered graph G induces a filtration on $\mathrm{Flag}(G)$, that we call a **flag bifiltration**, and that is the assignment of each grade $p \in \mathbb{R}^2$ to $\mathrm{Flag}(G_p)$.

It is common in multi-persistent homology to consider flag bifiltrations, like the density-Rips and degree bifiltrations [2, 4], and study its topology. Unfortunately, the combinatorial description of a flag filtration can be large, so we are interested in removing elements while preserving its topology. Boissonnat and Pritam [1] do exactly that in the single-parameter case, via the graph-theoretical concept of edge collapses (defined below). In our work, we

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

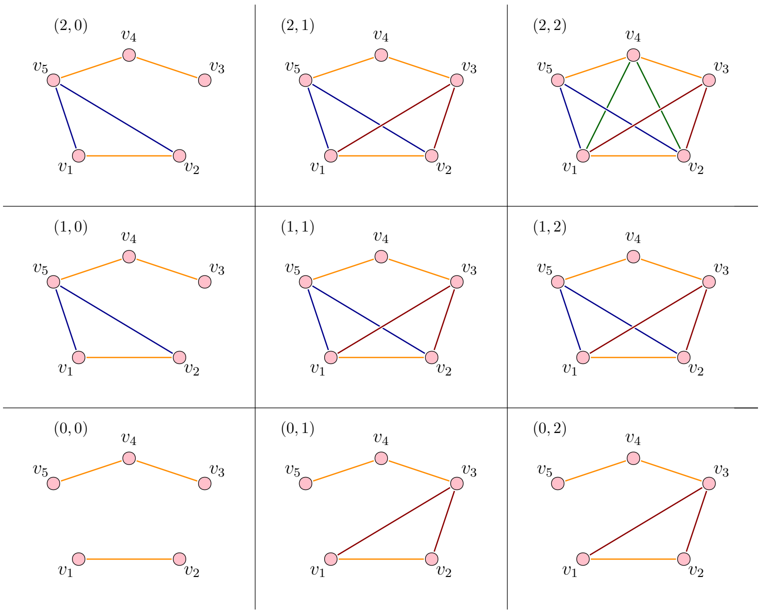


Figure 1 An example of a bifiltered graph, with underlying graph the complete graph K_5.

consider the multi-parameter case, by introducing an algorithm to remove what we call **filtration-dominated** edges (see below) from a bifiltered graph.

2 Filtration-domination and collapses

We briefly introduce edge collapses, as in [1]. Let G be a graph. For each vertex $v \in V(G)$, we denote by $N_G[v]$ the neighborhood of v: the set of vertices that are incident to v, plus v itself. For each edge $e \in E(G)$, we define the neighborhood of e as $N_G[e] := N_G[a] \cap N_G[b]$, where $a, b \in V(G)$ are the endpoints of e. An edge e is **dominated by a vertex** $v \notin e$, if $N_G[e] \subseteq N_G[v]$, and, when we do not care about the vertex that dominates, we say that e is **dominated**. We say that the removal of a dominated edge from G is an **edge collapse**.

The next definition introduces the concept of **filtration-domination**, which extends edge domination in a graph G to take into account the structure of the bifiltered graph.

▶ **Definition 1.** *Let G be a bifiltered graph. An edge e is **filtration-dominated** in G if for every $p \in \mathbb{R}^2$ the edge e is dominated in G_p, or e is not in G_p.*

The removal of a filtration-dominated edge from G does not change the multi-parameter persistence of the associated flag bifiltration (cf. [1, Theorem 4]). Thus, we can traverse the edges of G in an arbitrary order and remove edges that are filtration-dominated. In this exposition, we focus on the 1-critical case (for every edge e there is a $p \in \mathbb{R}^2$ such that, for any $q \in \mathbb{R}^2$, $e \in G_q$ if and only if $p \leq q$), but the ideas extend to the multi-critical case.

To check for filtration-domination it is enough to check for domination in the graphs G_p where the neighborhood of e changes, that is, at those $p \in \mathbb{R}^2$ such that $N_{G_p}[e] \neq N_{G_q}[e]$ for all $q < p$, since an edge e is dominated if there exists a vertex v such that $N_{G_p}[e] \subseteq N_{G_p}[v]$, and $N_{G_p}[v] \subseteq N_{G_{p'}}[v]$ if $p \leq p'$ for any two $p, p' \in \mathbb{R}^2$. Still, the number of such grades is greater than the number of neighbors of e in G. In fact, letting k be the maximum degree of a vertex in G, the neighborhood of e may change at $O(k^2)$ grades, and checking for domination at each of those grades is $O(k^2)$, resulting in a total complexity of $O(k^4)$.

As an alternative approach, we consider checking for filtration-domination by a single vertex, that is, for an edge e we check if there exists a vertex v (potentially different for each edge e) such that for every $p \in \mathbb{R}^2$ the edge e is dominated in G_p by v, or e is not in G_p. This is a relaxation of filtration-domination as in Theorem 1 leading to fewer edge removals, but we show it can be computed in $O(k^2)$ time, where k is the maximum degree of a vertex in G. In practice, we observed that the difference in the number of removals is marginal, whereas the speed-up is substantial, so we used this approach for the experimental results.

3 Experimental results

The procedure has been carefully implemented. We now describe preliminary results.

The order in which we visit each edge can be arbitrary. In our experiments, we have evaluated multiple orders, and we find that the reverse lexicographic order on the grades of the edges (visiting first the edges that appear later in the filtration) works best, and is the one we use. A paper by Glisse and Pritam [3], that deals with the single-parameter case and is being presented at this conference, also uses a "backwards" order, in contrast to [1].

We have run the procedure on density-Rips bifiltrations (Gaussian kernel with the bandwidth parameter set to the 20th percentile of all distances between distinct points) constructed on the datasets `eleg`, `senate`, `HIV`, `netw-sc`, and `dragon`, as described in [5]. For the first four datasets we remove those edges of scale parameter greater than a given threshold (the same threshold as in [1]), and we keep the `dragon` dataset as is. The results are shown in Figure 2. The red bars represent the quantity of remaining edges after running our algorithm. In all datasets, the algorithm removes more than 90% of the edges, which results in a speed-up in the subsequent algorithmic steps of the pipeline of multi-parameter persistence homology (more details on these experiments in the upcoming full paper). We have also run the single-parameter algorithm of [1] on the single-parameter filtrations constructed on the same datasets (by dropping the densities). We observe that our algorithm for bifiltrations removes a comparable number of edges, despite taking the density parameter into account and therefore being more selective for removing edges.

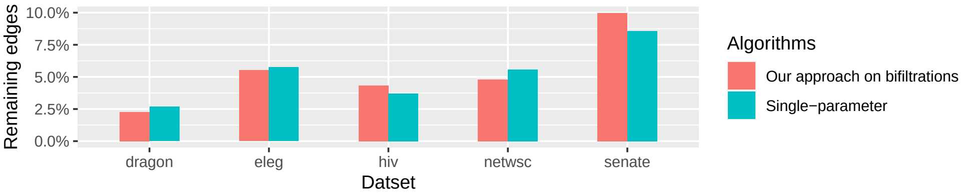


Figure 2 Bar plot of the quantity of remaining edges, shown as a percentage with respect to the initial number of edges, after running two algorithms: our algorithm on bifiltrations, and the single-parameter algorithm of [1] on single-parameter filtrations.

References

1 Jean-Daniel Boissonnat and Siddharth Pritam. Edge Collapse and Persistence of Flag Complexes. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.SoCG.2020.19`.

2 Gunnar Carlsson and Afra Zomorodian. The Theory of Multidimensional Persistence. *Discrete & Computational Geometry*, 42(1):71–93, April 2009. `doi:10.1007/s00454-009-9176-0`.

3 Marc Glisse and Siddharth Pritam. Swap, Shift and Trim to Edge Collapse a Filtration. *Accepted for the 38th International Symposium on Computational Geometry (SoCG 2022)*. `arXiv:2203.07022`.

4 Michael Lesnick and Matthew Wright. Interactive Visualization of 2-D Persistence Modules, December 2015. `arXiv:1512.00180`.

5 Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1), August 2017. `doi:10.1140/epjds/s13688-017-0109-5`.

An Approximation of Multiparameter Persistence Modules

David Loiseaux ✉
DataShape, Université Côte d'Azur, Inria, France

Andrew J. Blumberg ✉
Columbia University, New-York, USA

Mathieu Carrière ✉
DataShape, Université Côte d'Azur, Inria, France

Abstract

We present an approximation scheme based on the fibered barcode for computing an approximate decomposition of any general multiparameter persistence module, which has theorical guarantees *w.r.t.* the bottleneck distance when the module is interval decomposable. Our algorithm has a trade off between running time and precision, and works with an arbitrary number of filtrations.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Topological data analysis, Persistence modules, Multiparameter persistence

1 Introduction and Background

The main tool of Topological Data Analysis (TDA) [3, 7] is persistent homology (**PH**), which tracks the topological changes in a nested family of subspaces, called a filtration. **PH** encodes these changes in an algebraic structure called persistent module. Even though modules arising from single filtrations are well understood [5], computing and even approximating multiparameter persistent homology (**MPH**) is an important open question, which would drastically improve the use of TDA in data science. Indeed, **MPH** is known to be much richer than the single parameter case, taking into account, *e.g.*, both geometry and outliers [1] and/or several intrinsic filters [4]. Restricted to a filtration of dimension 2, this can be achieved efficiently with, *e.g.*, RIVET [10, 12]; but to our knowledge there is no implemented algorithm for all dimensions.

Multi-parameter persistence modules An n-multiparameter persistence module (or n-persistence module for short) is a family of vector spaces indexed over $\mathbb{R}^n$ with linear maps $M_x \to M_y$ if $x \leq y \in \mathbb{R}^n$, called *transition maps*; where the partial order $\leq$ in $\mathbb{R}^n$ is defined by $x \leq y \Leftrightarrow \forall i \;\; x_i \leq y_i$. In our case, these modules are obtained by applying the homology functor on a multi-filtration, *i.e.*, a family of subspaces $F_x \subseteq F$, for $x \in \mathbb{R}^n$ such that $F_x \subseteq F_y$, if $x \leq y \in \mathbb{R}^n$. Multipersistence modules can be compared with the *interleaving distance*, d_I [9], and the *bottleneck distance*, d_b [2, Section 2.3], which relies on the uniqueness of the decomposition into indecomposable modules [11, Azumaya's Theorem], which is known to be less stable but more discriminative than d_I. A common yet important class of modules are *interval modules*, defined below.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

Interval modules A subset I of $\mathbb{R}^n$ is called an *interval* if it satisfies:

- (convexity) if $p, q \in I$ and $p \leq r \leq q$, then $r \in I$,
- (connectivity) if $p, q \in I$, then there exist a finite sequence $r_1, r_2, \dots, r_m \in I$, for some $m \in \mathbb{N}$, such that $p \sim r_1 \sim r_2 \sim \cdots \sim r_m \sim q$, where $\sim$ can be either $\leq$ or $\geq$.

An n-multipersistence module M is an *n-interval module* if there exists a interval $I \subseteq \mathbb{R}^n$, called the support of M, and denoted by $\operatorname{supp}(M)$, such that:

$$\forall x \in \mathbb{R}^n,\ M_x = \begin{cases} k \text{ if } x \in I \\ \{0\} \text{ otherwise} \end{cases} \quad \text{and} \quad \forall x, y \in \mathbb{R}^n,\ \varphi_x^y = \begin{cases} \mathrm{id}_{k \to k} \text{ if } x \leq y \in I \\ 0 \text{ otherwise} \end{cases}.$$

Fibered barcodes and matching Let M be a multipersistence module. For any line l in $\mathbb{R}^n$, the multiset of bars $\mathcal{B}(M_l) := \left(\operatorname{supp}\left(M_i\big|_l\right)\right)_{i \in \mathcal{I}}$ is called the *barcode* of M along l. If L is a set of lines, the barcodes along the lines of L is called the *fibered barcode* [10] of M along L. Now, if $M = \bigoplus_{i \in \mathcal{I}} M_i$ is an interval decomposable n-module, and l^1, l^2 are two positive lines in $\mathbb{R}^n$, we say that a map $m \colon \mathcal{B}(M_{l^1}) \to \mathcal{B}(M_{l^2}) \cup \{\varnothing\}$ is an *(interval) exact matching* between l^1 and l^2 if each matched bar correspond to the same underlying summand.

2 Algorithm and guarantees

In this section, we present our general approximation scheme. The code for the following algorithm is publicly available at `https://gitlab.inria.fr/dloiseau/multipers`, and written in C++, with Python bindings.

Algorithm 1 Pseudocode to compute an approximate decomposition.

```
Preliminary calculation. Compute the barcodes along a $\frac{\delta}{2}$-grid (for the norm
  $\|\cdot\|_\infty$) of lines $L$, and match them with an exact matching. This can be done with
  e.g. the vineyard algorithm [6].
Input: For each module $I$ identified by this matching, consider its barcodes $B_L^I$
  along the lines of $L$.
Outputs: A list $B$ (resp. $D$) of birth corners in $\mathbb{R}^n$ (resp. death corners) defining
  the indicator module having support
      $$\bigcup_{b \in B} \bigcup_{d \in D} \{x \in \mathbb{R}^n : b \leq x \leq d\}.$$
  $B, D \leftarrow []$
for $l \in L$ do
    Consider the set of lines $L_l := \left\{ l + \delta \sum_{i \in \mathbf{j}} e_i \in L : \mathbf{j} \subseteq \{1, \dots, n-1\} \right\}$ and the
      bars $[b_l^I, d_l^I]$ of $I$ along it (when they exist).
    Birthpoints : Add to $B$ the birth corner
        $$\left( \min_{l' \in L_l} (b_l^I)_1, \min_{l' \in L_l} (b_l^I)_2, \dots, \min_{l' \in L_l} (b_l^I)_n \right).$$
    Deathpoints : Same as birth corner but by replacing min by max.
end
```

Note that Algorithm 1 is a simplified version of our complete algorithm. The output of Algorithm 1 can only be guaranteed in $\mathbb{R}^2$, although the extension of our algorithm to $\mathbb{R}^n$ is

based on the same idea. An example of our method is given in Figure 1.
On modules that are decomposable into a direct sum of interval modules, we have the following approximation error.

▶ **Theorem 1.** *Consider an n-persistence module M and a compact set $K \subseteq \mathbb{R}^n$. Assume that the restriction $M\big|_K$ of M to K is decomposable into interval modules. Then for any positive $\delta > 0$, there exists an interval decomposable $\tilde{M}_\delta$, computable using Algorithm 1, s.t.*

$$d_I\left(\tilde{M}_\delta, M\big|_K\right) \leq d_b\left(\tilde{M}_\delta, M\big|_K\right) \leq \delta,$$

with input set of diagonal lines L forming a $\frac{\delta}{2}$-grid over K along the canonical axes. Furthermore, under more restrictive assumptions on M (that are always true if M is defined from a finite dataset), there exists a $\delta > 0$ small enough such that $d_b\left(\tilde{M}_\delta, M\big|_K\right) = 0$.

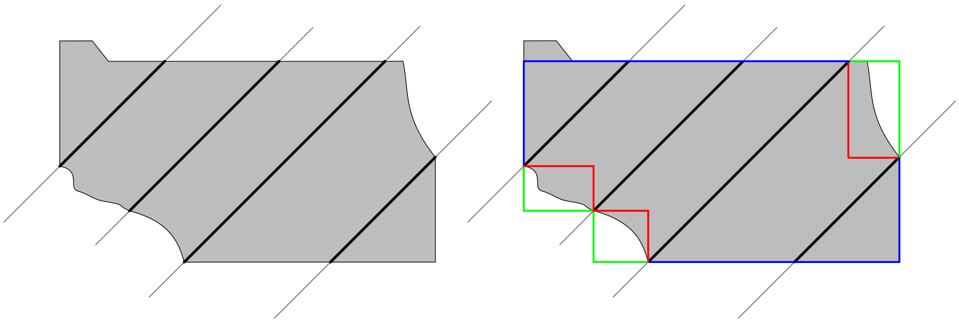


Figure 1 Example of reconstruction. **(Left)** A 2-interval module with 4 bars along 4 lines. **(Right)** An idea on how to infer the module between the bars. Ours is in green. It turns out that the structure of the boundary of an interval satisfies some kind of local stability [8]. One can thus bound the interval boundary between *neighbouring* lines, and compute a module approximation.

Our strategy of proof consists of two points. **(1)** Find a class of interval decomposable modules, constructible from the fibered (matched) barcodes, that is δ-close to the original module *w.r.t.* d_b; and **(2)** choose a candidate module in that class that is *simple* enough, to ensure that if the original module is also *simple*, then their bottleneck distance is 0.

3 Experiments

In this section, we present three experiments, displayed in Figure 2 3 and 4. We first look at two very simple, but different modules. Even though they have the same rank invariant, we can successfully recover their decomposition into interval modules with our algorithm. In our second experiment, we present the output of an indecomposable module that is not an interval. In our last experiment, we consider a noisy point cloud, with 20 000 points on an annulus with 40% outliers. The usual persistence diagram obtained from an Alpha complex cannot identify the 1-cycle of the annulus because of its sensitivity to outliers. With our reconstruction however (computed with a $\delta = 10^{-3}$ precision, i.e., ∼3k lines), we can combine both the Alpha and estimated density filtrations, in order to produce a decomposition where the cycle clearly appears (the large orange summand) in a short amount of time.

$$\begin{tikzcd} k \arrow[r, "\begin{pmatrix}1\\0\end{pmatrix}"] & k^2 \\ & k \arrow[u, "\begin{pmatrix}1\\0\end{pmatrix}"] \end{tikzcd} \qquad \text{Versus} \qquad \begin{tikzcd} k \arrow[r, "\begin{pmatrix}1\\0\end{pmatrix}"] & k^2 \\ & k \arrow[u, "\begin{pmatrix}0\\1\end{pmatrix}"] \end{tikzcd}$$

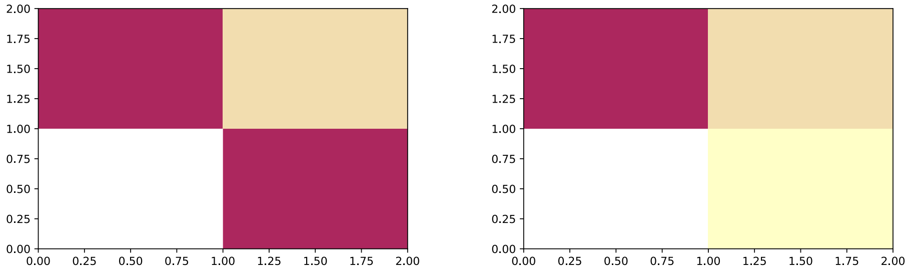
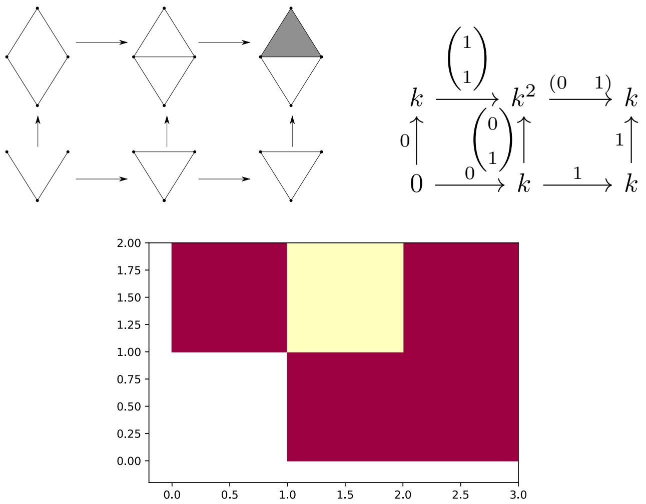


Figure 2 Bimodules with the same rank invariant. **(Top)** Modules definitions. **(Bottom)** Output of our algorithm.



$$\begin{tikzcd} k \arrow[r, "\begin{pmatrix}1\\1\end{pmatrix}"] & k^2 \arrow[r, "\begin{pmatrix}0 & 1\end{pmatrix}"] & k \\ 0 \arrow[u, "0"] \arrow[r, "0"] & k \arrow[u, "\begin{pmatrix}0\\1\end{pmatrix}"] \arrow[r, "1"] & k \arrow[u, "1"] \end{tikzcd}$$

Figure 3 A non-interval decomposable module. **(Left)** The simplicial chain complex bifiltration used as an input. **(Right)** The 1-homology of this bifiltration. **(Bottom)** The output of our algorithm.

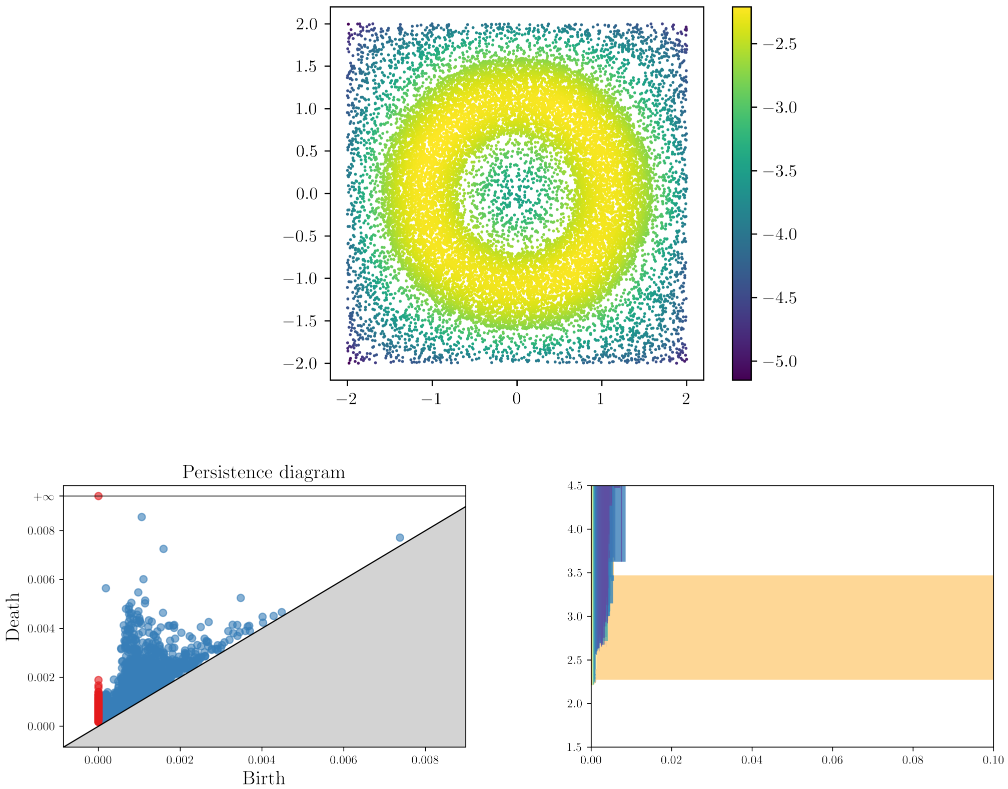


Figure 4 Noisy annulus bimodule. **(Top)** The dataset. The colours correspond to the estimated log-density values. **(Left)** The usual persistence diagram. **(Right)** The output of our algorithm.

References

1 Andrew J. Blumberg and Michael Lesnick. Stability of 2-parameter persistent homology. *arXiv:2010.09628 [cs, math]*, February 2021. URL: `http://arxiv.org/abs/2010.09628`, `arXiv:2010.09628`.

2 Magnus Bakke Botnan and Michael Lesnick. Algebraic stability of zigzag persistence modules. *Algebraic & Geometric Topology*, 18(6):3133–3204, October 2018. `arXiv:1604.00655`, `doi:10/gkw5xh`.

3 Gunnar Carlsson. Topology and data. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC*, 46:255–308, April 2009. `doi:10.1090/S0273-0979-09-01249-X`.

4 Mathieu Carrière and Andrew J. Blumberg. Multiparameter persistence image for topological machine learning. In *NeurIPS*, 2020.

5 Frédéric Chazal, Vin deSilva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. SpringerBriefs in Mathematics. Springer International Publishing, Cham, 2016. `doi:10.1007/978-3-319-42545-0`.

6 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, SCG '06, pages 119–126, New York, NY, USA, June 2006. Association for Computing Machinery. `doi:10/b2sscd`.

7 Herbert Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, Providence, R.I, 2010.

8 Claudia Landi. The rank invariant stability via interleavings. In *Research in Computational Topology*, pages 1–10. Springer, 2018.

9 Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, June 2015. `arXiv:1106.5305`, `doi:10/f7c7fc`.

10 Michael Lesnick and Matthew Wright. Interactive visualization of 2-D persistence modules. *arXiv:1512.00180 [cs, math]*, December 2015. URL: `http://arxiv.org/abs/1512.00180`, `arXiv:1512.00180`.

11 Steve Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*, volume 209 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, Rhode Island, December 2015. `doi:10.1090/surv/209`.

12 The RIVET Developers. RIVET, 2020. URL: `https://github.com/rivetTDA/rivet/`.

The Shift-Dimension of Multipersistence Modules

Wojciech Chachólski
Department of Mathematics, KTH Royal Institute of Technology
Lindstedtsvägen 25, 114 28 Stockholm, Sweden

René Corbet
Department of Mathematics, KTH Royal Institute of Technology
Lindstedtsvägen 25, 114 28 Stockholm, Sweden

Anna-Laura Sattelberger
Max-Planck-Institut für Mathematik in den Naturwissenschaften
Inselstraße 22, 04103 Leipzig, Germany

Abstract

In topological data analysis, multipersistence has a provably more difficult theory and leads to substantially bigger computational challenges than ordinary persistence does. Therefore, applications cannot be realized as easily as in the case of ordinary persistence. A main difficulty is the incompleteness of invariants in multipersistence, such as the Hilbert function and the rank invariant. Phrasing the incompleteness in a more data-scientific way, the invariants may not capture the aspects of interest of given data. As a matter of fact, a clever design of useful invariants is important. We contribute a new invariant, investigate its algebraic properties, give a fast algorithm for the computation for interval modules in two parameters, and discuss its potential use in applications.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Topological data analysis; multiparameter persistence; stable invariants; multigraded modules

Related Version https://arxiv.org/abs/2112.06509

Funding *Wojciech Chachólski*: VR, the Wallenberg AI, Autonomous System and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation, and MultipleMS funded by the European Union under the Horizon 2020 program, grant agreement 733161, and dBRAIN collaborative project at digital futures at KTH
René Corbet : Brummer & Partners MathDataLab
Anna-Laura Sattelberger: Brummer & Partners MathDataLab

1 Introduction

Establishing a useful pipeline of multipersistence, see Figure 1, is a recent task in topological data analysis. It deserves and enjoys lots of attention from many perspectives: theoretically, computationally, algorithmically, and towards implementations and applications. It is essential to provide computationally feasible constructions to make the pipeline more practical for data-scientific applications. One important aspect of the pipeline is the design of various insightful invariants. This stems from the fact that the persistence diagram from one-parameter persistence does not generalize to a discrete complete invariant in the case of multipersistence [2].

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

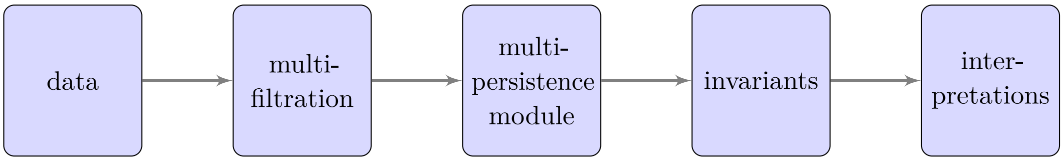


Figure 1 The pipeline of multipersistence.

2 Definition

We contribute a new invariant for multipersistence modules, the *shift-dimension*. For an r-parameter multipersistence module M, which we view as $\mathbb{R}^r_{\geq 0}$-graded module over the graded monoid ring $\mathbb{K}[\mathbb{R}^r_{\geq 0}]$, and $v \in \mathbb{R}^r_{\geq 0}$, we define the shift dimension $\dim_v(M)$ to be the smallest number of elements $m_1, \ldots, m_k$ of M such that

$$v * M \subseteq \langle m_1, \ldots, m_k \rangle,$$

where left-multiplication by v denotes the standard shift in the grading along v.

Letting the direction of the vector v fixed but its length vary, the shift-dimension induces a non-increasing map $\mathbb{R} \to \mathbb{N}$, which is suitable for machine learning algorithms when viewed as *feature map*. More generic approaches are possible as well, i.e., additionally letting the direction of v vary in ℓ spatial directions yields a feature map $\mathbb{R}^{\ell+1} \to \mathbb{N}$.

Remarkably, the algebraic definition of the shift-dimension arises as the *hierarchical stabilization* [4] of the minimal number of generators. The hierarchical stabilization adds a geometric structure to (the) minimal generators, letting us cluster them to subsets, depending on the choice of v. The clustering rule, however, is not unique and combinatorially complicated; in fact, computing the shift-dimension is algorithmic, but in general NP-hard [4].

3 Computation

To understand the behavior and the computational complexity of the shift-dimension, we introduce and investigate the notion of a *v-basis* of a module: it is a collection of elements of the module satisfying the property of the shift-dimension. We give rules for exchanging v-bases, but even for very simple examples of modules, the set of all of its v-bases do not form a matroid. Therefore, the exchange properties of v-bases partly encode the combinatorial difficulty and, hence, the computational complexity of the shift-dimension.

In the case of two-parameter interval modules, we give a *linear-time algorithm* for the computation of the shift-dimension. Note that interval modules correspond to quotients of monomial ideals, viewed as modules. We endow the minimal generators of an interval module with a total order and cluster them by an explicit rule. Our clustering rule follows a certain notion of degree-wise closeness induced by v. The algorithm computes that clustering via elementary operations in Euclidean space; see Figure 3 for both an illustration and a more explicit description of the algorithm. Even for infinitely presented interval modules, the shift-dimension is finite and gets computed by our algorithm whenever v is nonzero and neither horizontal nor vertical. Consequently, the shift-dimension may also serve as a practical tool for a certain class of infinitely presented modules that arise from data [5].

Furthermore, we give toy examples for the behavior of the shift-dimension for *direct sums of interval modules*. This class may arise directly as homology of certain multifiltrations [3], or as approximation of arbitrary finitely presented modules in two parameters [1]. The shift-dimension is not additive with respect to direct sums. In Figure 2, we give an example for this phenomenon on a direct sum of four interval modules and explain the mathematical properties that make the shift-dimension non-additive. A more rigorous study of these properties may give rise for a deeper understanding of when additivity breaks. For experiments and data-scientific applications, however, summing up the shift-dimension of the individual summands serves as an alternative to the shift-dimension of the direct sum, as illustrated in Figure 2. Using our algorithm, we can compute the additive version efficiently.

References

1 H. Asashiba, M. Buchet, E. Escolar, K. Nakashima, and M. Yoshiwaki. On interval decomposability of 2D persistence modules. Preprint arXiv:1812.05261, 2018.

2 G. Carlsson and A. Zomorodian. The theory of multidimensional persistence. *Discrete Comput. Geom.*, 42:71–93, 2009.

3 E. Escolar and Y. Hiraoka. Persistence modules on commutative ladders of finite type. *Discrete Comput. Geom.*, 55(1):100–157, 2016.

4 O. Gäfvert and W. Chachólski. Stable invariants for multidimensional persistence. Preprint arXiv:1703.03632, 2017.

5 E. Miller. Data structures for real multiparameter persistence modules, 2020. Preprint arXiv:1709.08155.

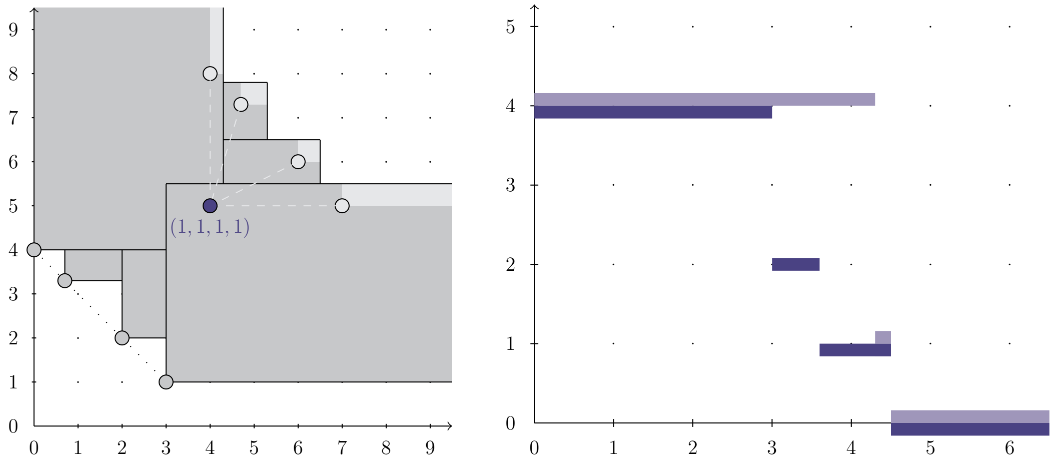


Figure 2 *Left:* Illustration of a direct sum of four interval modules M_i, all of which are supported by rectangular intervals. For $v = 4 \cdot (1,1)$, we have $\dim_v(\oplus_i M_i) = 1$, but $\sum_i \dim_v(M_i) = 4$. A v-basis of $\oplus_i M_i$ is given by the all-one vector at degree $(4,5)$. This quite extremal difference stems from the fact that (i) in the degrees of the v-shifted generators, the other summands are zero, and (ii) in the greatest common divisor of the degrees of the v-shifted generators, all summands are nonzero. *Right:* Comparison of the functions $\sum_i \dim_{\tau v}(M_i)$ (in lavender) and $\dim_{\tau v}(\oplus_i M_i)$ (in blue) for $v = (1,1)$.

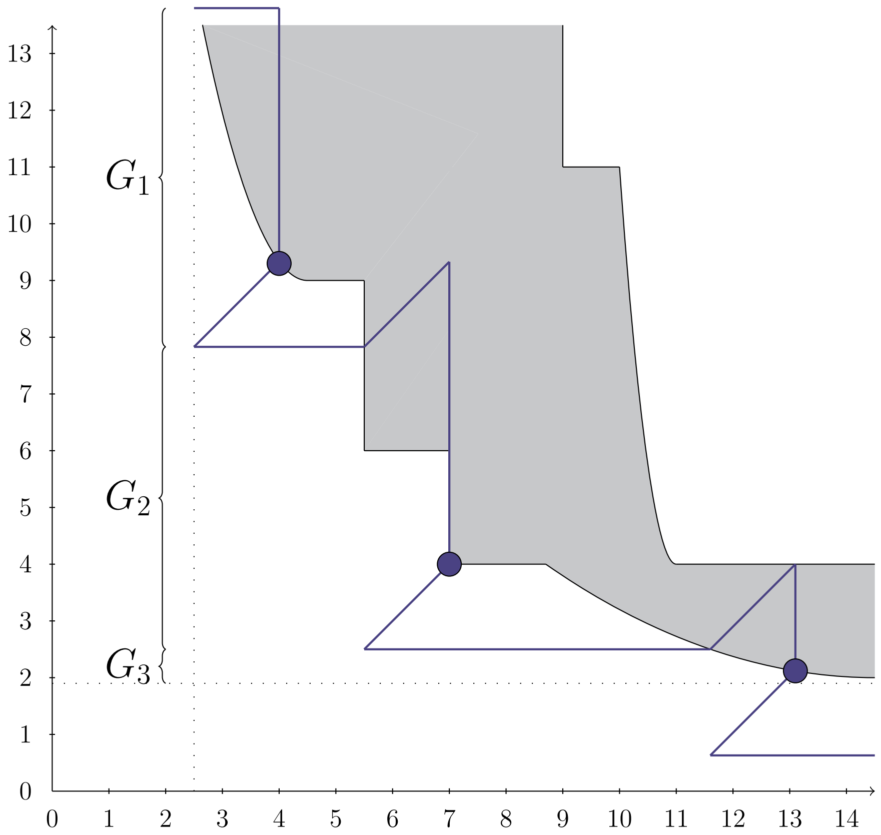


Figure 3 Using our algorithm for the construction of a v-basis of an infinitely presented interval module, with $v = (1.5, 1.5)$. The gray-shaded region denotes the interval. The algorithm starts at the degree of the top left generator (in this case, a limit), adds v, projects down to the generating curve of the module, subtracts v, and projects right to the generating curve of the module. This procedure repeats until one of the projections is not possible any more. The v-basis consists of those generators obtained by the vertical projection. In this example, the minimal generators of the module get clustered into subsets G_1, G_2, and G_3. Hence, the shift-dimension of this module is 3 with respect to v.

A Maximum Subbarcode Matching Algorithm

Oliver Anthony Chubet
North Carolina State University, United States
[0]

Abstract

We investigate the maximum subbarcode matching problem which arises from the study of persistent homology. A barcode is a set of intervals which correspond to topological features in data and is the main object of interest in a persistent homology computation. A barcode A is a subbarcode of B if each interval in A corresponds to an interval in B which contains it. We present an algorithm which takes two barcodes A and B and returns a maximum subset of A which is a subbarcode of B. Our algorithm also works on multiset input. We use a sweepline approach to yield an algorithm that runs in $O(n \log n)$ time and uses $O(n)$ space where n is the number of distinct intervals in the barcodes.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases algorithms, matching, multisets, subbarcodes

1 Introduction

In persistent homology the main object of interest is the barcode, which is a multiset of intervals encoding topological information about a given set of data. There is new interest in the implications arising from having partial knowledge or an approximation of the barcode. A barcode $\mathbb{A}$ is a subbarcode of $\mathbb{B}$ if there is a correspondence such that each interval in A maps to an interval in B which contains it. Sheehy [6] introduces the theory of subbarcodes and explores how one may falsify hypotheses about barcodes by computing subbarcodes.

A multiset $\mathbb{A} = (A, \omega_A)$ is a pair with a set A and multiplicity function $\omega_A : A \to \mathbb{Z}^+$. The weight of $\mathbb{A}$ is the sum of the multiplicity function over A, denoted $|\mathbb{A}| = \sum_{a\in A} \omega_A(a)$. A matching $\mathbb{M}$ between multisets $\mathbb{A} = (A, \omega_A)$ and $\mathbb{B} = (B, \omega_B)$ is a multiset $\mathbb{M} = (M, \omega)$ where $M \subset A \times B$ with multiplicity function $\omega : M \to \mathbb{Z}^+$ such that $\sum_{b\in B} \omega(a, b) \leq \omega_A(a)$ for all $a \in A$, and $\sum_{a\in A} \omega(a, b) \leq \omega_B(b)$ for all $b \in B$. A matching $\mathbb{M}$ is a maximum matching if it has maximum weight over all valid matchings.

A barcode $\mathbb{B} = (B, \omega_B)$ is a multiset where B is a finite set of intervals. We represent a barcode as points in $\mathbb{R}^2$ to exploit the geometry of the plane.

A subbarcode matching from $\mathbb{S}$ to $\mathbb{B}$ is a multiset matching $\mathbb{M} = (M, \omega_M)$, of $\mathbb{S}$ and $\mathbb{B}$ where $(s, b) \in M$ implies s is contained in b as intervals. If $s = (s_L, s_R)$ and $b = (b_L, b_R)$ then b contains s if $b_L \leq s_L \leq s_R \leq b_R$. See Figure 2. The maximum subbarcode matching problem is to find a subbarcode matching of maximum weight.

2 Related Work

We use a sweepline approach in our subbarcode matching algorithm. A sweepline algorithm sorts the input by one coordinate and then sweeps through the plane in order, doing updates as each element is reached [1]. Two related problems where this paradigm has been used include the maximum matching problem for intersecting intervals [2] and maximum matching in convex bipartite graphs [3, 7, 4]. The ability to reduce these matching problems to a

[0] This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

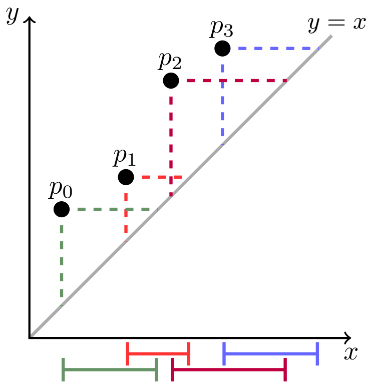


Figure 1 We may represent intervals as points in $\mathbb{R}^2$ by taking their endpoints as coordinates .

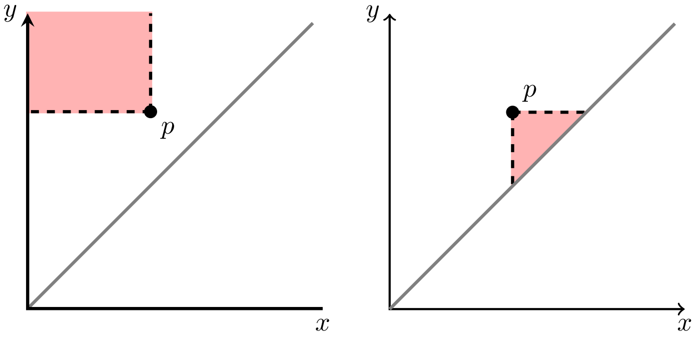


Figure 2 Any point in the shaded region on the left contains p as an interval. Any point in the shaded region on the right is contained in p as an interval.

sweepline problem yields a significant runtime benefit, most running in $O(n \log n)$ time, whereas the traditional Hopcroft-Karp algorithm for maximum matching in bipartite graphs runs in $O(n^{\frac{5}{2}})$ [5]. The strategy used in these algorithms is to avoid backtracking to keep the total operations per element small.

3 Algorithm

As input, our algorithm takes two barcodes $\mathbb{A} = (A, \omega_A), \mathbb{B} = (B, \omega_B)$ and returns a maximum subbarcode matching $\mathbb{M}$ from $\mathbb{A}$ to $\mathbb{B}$. Refer also to Figure 3.

```
**Sort** $A \cup B$ by the $x$-coordinate.
**Initialize** T to be an empty balanced binary search tree to store points from $B$ ordered
by $y$-coordinate. Initialize the residual weights $\mathtt{r}_b = \omega_B(b)$ for each $b \in B$ and $\mathtt{r}_a = \omega_A(a)$
for each $a \in A$. Initialize $(\mathtt{M}, \mathtt{W})$ to store the matching and multiplicities.
**For each** $p \in A \cup B$ in sorted order, where $p = (p_x, p_y)$, do the following:
  **If** $p \in B$, move $b$ into T.
  **Else**
    **While** $\mathtt{r}_p > 0$:
      Search for a $b \in$ T with minimum $b_y$ such that $b_y \geq p_y$.
      If there is none, then break.
```

```
            Let $\mathtt{r} = \min\{\mathtt{r}_p, \mathtt{r}_b\}$. Add $(p, b)$ to $\mathtt{M}$ and set $\mathtt{W}[(p, b)] = \mathtt{r}$, then update the
            residual weights of $p$ and $b$: $\mathtt{r}_p = \mathtt{r}_p - \mathtt{r}$ and $\mathtt{r}_b = \mathtt{r}_b - \mathtt{r}$.
            If $\mathtt{r}_b = 0$, then remove $b$ from $\mathtt{T}$.
    Return $\mathbb{M} = (\mathtt{M}, \mathtt{W})$.
```

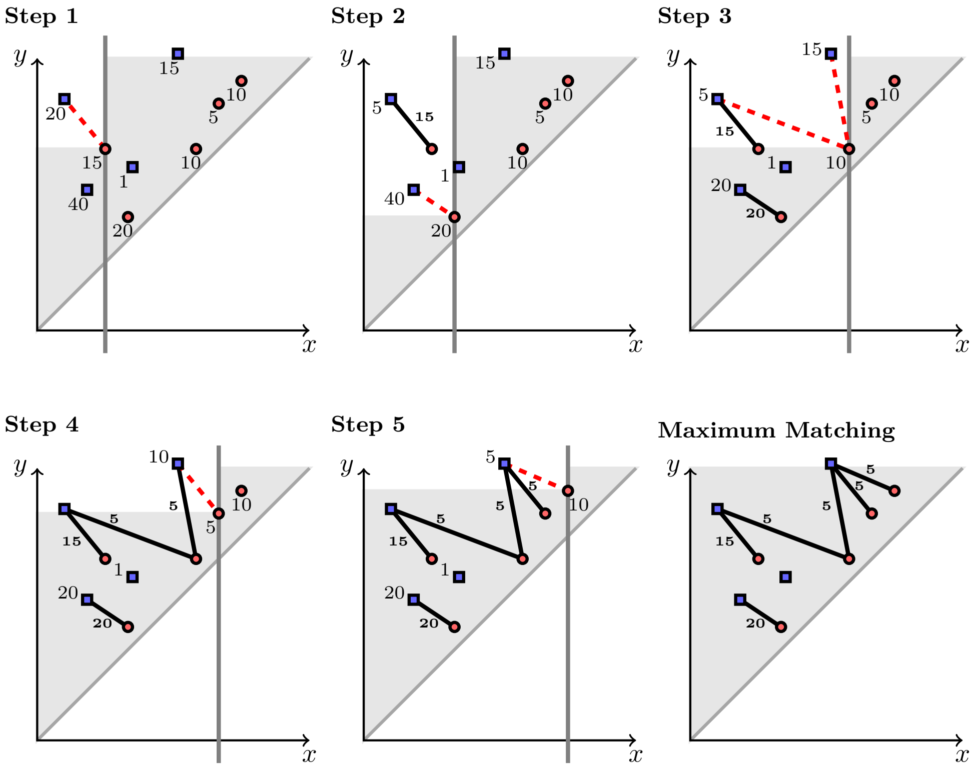


Figure 3 We find a maximum subbarcode matching from $\mathbb{A}$ to $\mathbb{B}$ (red circles and blue squares respectively) labeled by their multiplicities. We iterate through $\mathbb{A}$ in order of x-coordinate and match to the point in $\mathbb{B}$ with lowest y-coordinate. Each edge represents the match labeled with the multiplicity, and the residual multiplicities are updated for $\mathbb{A}$ and $\mathbb{B}$ accordingly.

Our algorithm finds a maximum subbarcode matching in $O(n \log n)$ time, where n is the number of distinct points. The space requirements are $O(n)$ in the number of distinct points. One may note that the maximum subbarcode matching may not be unique. In fact for certain inputs, there exists a maximum subbarcode matching of size $O(n^2)$. However, this algorithm will never consider these solutions. This algorithm will always return a matching of linear size.

4 Conclusion

The algorithm given in Section 3 gives an efficient method for computing maximum subbarcode matchings. By using a sweepline algorithm we avoid solving the maximum matching problem for bipartite graphs, allowing us to acheive $O(n \log n)$ runtime and $O(n)$ space. We present this as a computationally efficient method of comparison for persistence diagrams, which was the original problem setting. This algorithm may also apply to other problems requiring matchings of nested intervals.

In future work we will show how our algorithm can be extended to find the minimum shift needed to yield a perfect subbarcode matching. In addition we will address the relationship between subbarcodes and bottleneck distance, a metric commonly used to compare persistence diagrams.

References

1 Jon Louis Bentley and Thomas A Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on computers*, 28(09):643–647, 1979. doi:10.1109/TC.1979.1675432.

2 Danny Z Chen, Xiaobo Sharon Hu, and Xiaodong Wu. Maximum red/blue interval matching with application. In *International Computing and Combinatorics Conference*, pages 150–158. Springer, 2001. doi:10.1007/3-540-44679-6_17.

3 Gianluca Gallo. An o (n log n) algorithm for the convex bipartite matching problem. *Operations Research Letters*, 3(1):31–34, 1984. doi:10.1016/0167-6377(84)90068-3.

4 Fred Glover. Maximum matching in a convex bipartite graph. *Naval research logistics quarterly*, 14(3):313–316, 1967. doi:10.1002/nav.3800140304.

5 John E Hopcroft and Richard M Karp. An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973. doi:10.1137/0202019.

6 Don Sheehy. The persistent homology of lipschitz extensions. https://www.youtube.com/watch?v=d8dgrnZUwOU, 2022. AATRN Vietoris–Rips Online Seminar.

7 George Steiner and Julian Scott Yeomans. A linear time algorithm for maximum matchings in convex, bipartite graphs. *Computers & Mathematics with Applications*, 31(12):91–96, 1996. doi:10.1016/0898-1221(96)00079-X.

The Complexity of Geodesic Spanners

Sarita de Berg ✉
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Marc van Kreveld ✉
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Frank Staals ✉
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

We study a novel property for spanners with non-constant complexity edges: the spanner *complexity*, i.e. the total complexity of all edges. For n point sites in a simple polygon with m vertices, we show a lower bound on the complexity of any geodesic $(t-\varepsilon)$-spanner of $\Omega(mn^{1/(t-1)}+n)$, for any constant $\varepsilon > 0$ and integer constant $t \geq 2$, and provide a construction for a $4\sqrt{2}$-spanner of complexity $O((m\sqrt{n}+n)\log^2 n)$.

2012 ACM Subject Classification Theory of computation → Computational Geometry

Keywords and phrases spanner, simple polygon, geodesic distance, complexity

1 Introduction

In the design of networks on a set of nodes, we often consider two criteria: few connections between the nodes, and small distances. Spanners are geometric networks on point sites that replace the small distance criterion by a small detour criterion. Formally, a *geometric t-spanner* for a set S of n point sites in $\mathbb{R}^2$ is an edge-weighted graph $\mathcal{G} = (S, E)$ for which the distance $d_{\mathcal{G}}(p, q)$ between any two sites $p, q \in S$ is at most $t \cdot d(p, q)$, where $d(p, q)$ denotes the distance between p and q in the distance metric we consider. The smallest t for which a graph $\mathcal{G}$ is a t-spanner is called the *spanning ratio* of $\mathcal{G}$. The number of edges in the spanner is called the *size* of the spanner.

The spanning ratio and the size of spanners are not the only properties of spanners that can be optimized. Many different properties have been studied, such as total weight (or lightness), maximum degree, (hop) diameter, and fault-tolerance [4, 5].

When we consider distance metrics for which the edges in the spanner—which are shortest paths—no longer have constant complexity, another interesting property of spanners arises: the spanner *complexity*, i.e. the total complexity of all edges. We study this novel property in a setting where our sites lie in a simple polygon P with m vertices, and we measure the distance between two sites p, q by their geodesic distance: the length of the shortest path between p and q fully contained within P. The complexity of an edge between p and q is then the number of line segments in the shortest (geodesic) path between p and q. In this setting, a single edge of a spanner may have complexity $\Theta(m)$. Recently, Abam et al. [2] showed that a geodesic $(2+\varepsilon)$-spanner with $O(n \log n)$ edges exists even for n points on a polyhedral terrain. We show that any $(3-\varepsilon)$-spanner may have complexity $\Omega(nm)$, thus implying that the $(2+\varepsilon)$-spanner of Abam et al. [2] may also have complexity $\Omega(nm)$, despite having $O(n \log n)$ edges.

To improve this complexity, we show that there exists a geodesic $4\sqrt{2}$-spanner of complexity $O((m\sqrt{n}+n)\log^2 n)$. Additionally, we show a lower bound for the complexity of any $(t-\varepsilon)$-

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

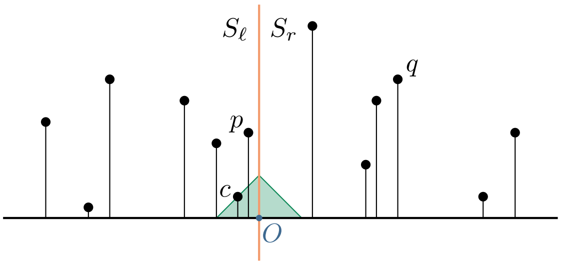


Figure 1 Construction of the additively weighted 1-dimensional spanner.

spanner of $\Omega(mn^{1/(t-1)} + n)$. Most proofs are omitted and will be included in a future full version.

2 A 1-dimensional additively weighted 2-spanner

Abam et al. [2] show that we can use a 1-dimensional spanner to construct a geodesic spanner in a simple polygon. Therefore, we first consider an additively weighted spanner $\mathcal{G}$ in 1-dimensional Euclidean space, where each site $p \in S$ has a weight $w(p) \geq 0$. The distance between two sites $p, q \in S$ is given by $d_w(p,q) = w(p) + |pq| + w(q)$, where $|pq|$ denotes the Euclidean distance. We can map $\mathbb{R}$ to the x-axis, and the weights to the y-axis, see Figure 1.

To construct our spanner $\mathcal{G}$, we partition the points into two sets S_ℓ and S_r of roughly equal size by a point O with $w(O) = 0$. S_ℓ contains all points left of O, and $S_r := S \setminus S_\ell$. We then find a point $c \in S$ for which $d_w(c, O)$ is minimal. For all $p \in S$, $p \neq c$, we add the edge (p, c) to $\mathcal{G}$. Finally, we handle the sets S_ℓ and S_r, excluding the site c, recursively.

▶ **Lemma 1.** *The graph $\mathcal{G}$ is a 2-spanner of size $O(n \log n)$.*

3 A simple geodesic spanner

Just like Abam et al. [2], we use our 1-dimensional spanner to construct a geodesic spanner for a set S of n points in a simple polygon P. We denote by $d(p,q)$ the geodesic distance between p, q, and by $\pi(p,q)$ the shortest (geodesic) path from p to q. We analyze the construction with respect to any 1-dimensional additively weighted t-spanner of size $O(n \log n)$.

As in [1, 2], we partition P into two subpolygons P_ℓ and P_r by a line segment λ, such that each subpolygon contains at most two thirds of the sites [3]. We denote by S_ℓ and S_r the sites in P_ℓ and P_r, respectively. For each $p \in S$, we then find the point p_λ on λ closest to p. As λ is a line segment, the set S_λ, containing all projected points, gives rise to a 1-dimensional Euclidean space, where $w(p_\lambda) := d(p, p_\lambda)$. We compute a t-spanner $\mathcal{G}_\lambda = (S_\lambda, E_\lambda)$ of size $O(n \log n)$ for this set. For each pair $(p_\lambda, q_\lambda) \in E_\lambda$, we add the edge (p, q) to our spanner $\mathcal{G}$. Finally, we recursively compute spanners for S_ℓ and S_r, and add their edges to $\mathcal{G}$ as well.

▶ **Lemma 2.** *The graph $\mathcal{G}$ is a geodesic $t\sqrt{2}$-spanner of size $O(n \log^2 n)$.*

Proof sketch. The main idea is to bound the difference between $d(p,q)$ and the length of the path $p \to p_\lambda \to q_\lambda \to q$ by considering the triangle $\mathcal{T} = (z, z', r)$ shown in Figure 2. ◀

4 Complexity of geodesic spanners

The construction in Figure 3 shows that any $(3 - \epsilon)$-spanner, so in particular the $2\sqrt{2}$-spanner from Section 2 and 3, has complexity $\Omega(nm)$. Additionally, the following theorem implies a

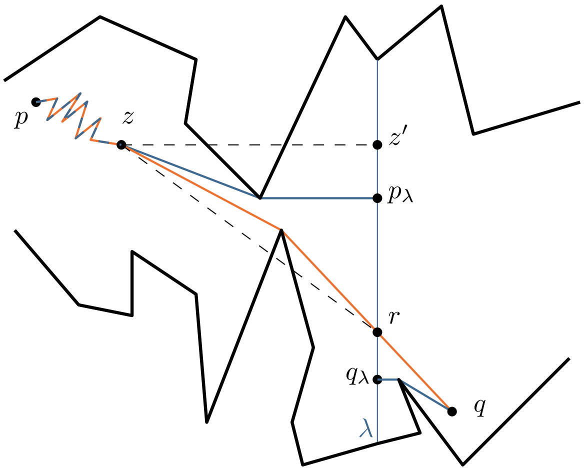


Figure 2 The shortest path $\pi(p, q)$ crosses λ at r. The difference in length between the direct path from z to r and the path through p_λ can be bounded by considering the triangle $\mathcal{T} = (z, z', r)$.

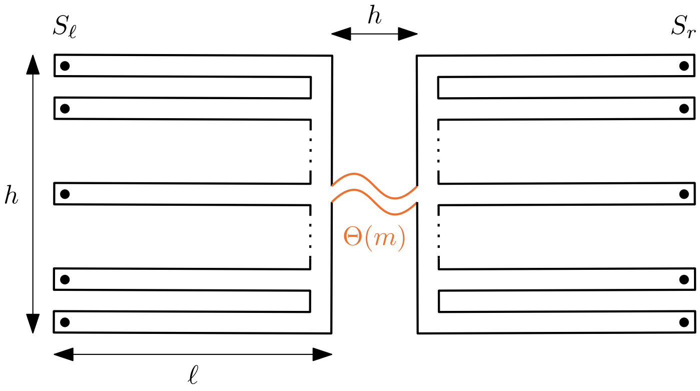


Figure 3 Any $(3 - \varepsilon)$-spanner in a simple polygon with m vertices may have complexity $\Omega(nm)$.

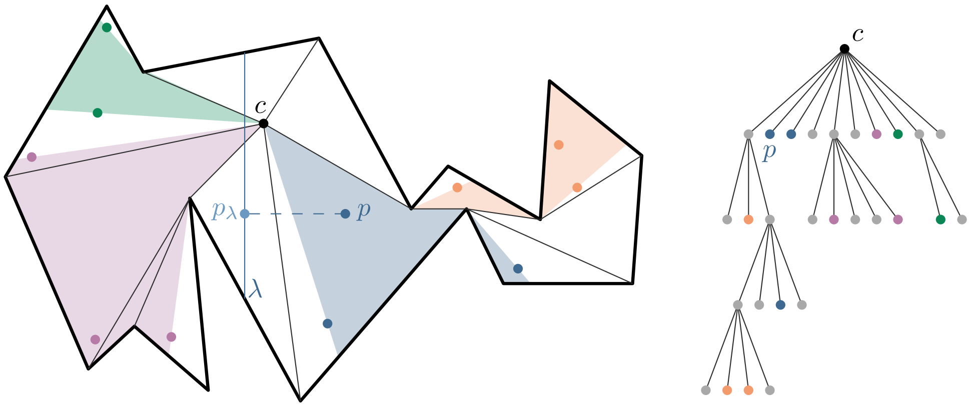
trade-off between the spanning ratio and the spanner complexity.

▶ **Theorem 3.** *For any constant $\varepsilon > 0$ and integer constant $t \geq 2$, there exists a set of n points in a simple polygon P with $m = \Omega(n)$ vertices for which any geodesic $(t - \varepsilon)$-spanner has complexity $\Omega(mn^{1/(t-1)})$.*

Next, we present a $4\sqrt{2}$-spanner of complexity $O((m\sqrt{n} + n)\log^2 n)$. We adapt our construction for the 1-dimensional spanner $\mathcal{G}_\lambda$ as follows. After finding the site $c \in S$ closest to O, we do not add all edges (p, c) to $\mathcal{G}_\lambda$. Instead, we form groups of $O(\sqrt{n})$ sites whose original points (before projection to λ) lie 'close' to each other in P. We choose these groups based on the shortest path tree of c: the union of all shortest paths from c to the vertices of P. See Figure 4. For each group S_i, we add all edges (p, c_i), $p \in S_i$, to $\mathcal{G}_\lambda$, where c_i is the site in S_i for which $d_w(c_i, O)$ is minimal. Finally, we add all edges (c_i, c) to $\mathcal{G}_\lambda$.

▶ **Theorem 4.** *Let S be a set of n point sites in a simple polygon P with m vertices. There exists a geodesic $4\sqrt{2}$-spanner of size $O(n\log^2 n)$ and complexity $O((m\sqrt{n} + n)\log^2 n)$.*

Proof sketch. There are two types of edges in the spanner: 1) edges from some c_i to c, and 2) edges from some $p \in S_i$ to c_i. There are $O(\sqrt{n})$ type 1 edges, that each have a complexity



🟨 **Figure 4** The shortest path tree SPT_c of c, where each $p \in S \setminus \{c\}$ is included as a child of the last vertex on $\pi(c, p)$. Each colored group S_i is chosen based on the in-order traversal of SPT_c. The first $\lceil \sqrt{n} \rceil$ sites are assigned to S_1, the second $\lceil \sqrt{n} \rceil$ to S_2, etc.

of $O(m)$. To bound the complexity of all type 2 edges, we show that each group defines a region in P that contains all edges within that group, see Figure 4. The interiors of these regions are disjoint, and thus the total complexity of all type 2 edges is $O(m\sqrt{n} + n)$. ◀

Let $k \geq 1$ be an integer constant. We can generalize this approach by recursively assigning sites to groups to obtain a geodesic $(2k + \varepsilon)$-spanner of complexity $O((mn^{1/k} + n) \log^2 n)$.

References

1 Mohammad Ali Abam, Marjan Adeli, Hamid Homapour, and Pooya Zafar Asadollahpoor. Geometric spanners for points inside a polygonal domain. In *31st International Symposium on Computational Geometry, SoCG*, volume 34 of *LIPIcs*, pages 186–197, 2015.

2 Mohammad Ali Abam, Mark de Berg, and Mohammad Javad Rezaei Seraji. Geodesic spanners for points on a polyhedral terrain. *SIAM J. Comput.*, 48(6):1796–1810, 2019.

3 Prosenjit Bose, Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, and Anil Maheshwari. Polygon cutting: Revisited. In *Discrete and Computational Geometry, Japanese Conference, JCDCG, Revised Papers*, volume 1763 of *LNCS*, pages 81–92, 1998.

4 Prosenjit Bose and Michiel H. M. Smid. On plane geometric spanners: A survey and open problems. *Comput. Geom.*, 46(7):818–830, 2013.

5 Giri Narasimhan and Michiel H. M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

On approximating shortest paths in weighted hexagonal tessellations

Prosenjit Bose ✉
School of Computer Science, Carleton University, Canada

Guillermo Esteban ✉
School of Computer Science, Carleton University, Canada
Departamento de Física y Matemáticas, Universidad de Alcalá, Spain

David Orden ✉
Departamento de Física y Matemáticas, Universidad de Alcalá, Spain

Rodrigo I. Silveira ✉
Departament de Matemàtiques, Universitat Politècnica de Catalunya, Spain

Abstract

A natural way to discretize continuous 2-dimensional space is by considering a weighted hexagonal grid. In this work, we study how well a shortest path between two vertices s and t in the tessellated space $SGP_w(s,t)$ approximates a shortest path $SP_w(s,t)$ from s to t in the continuous weighted space. Our main result is that the ratio $\frac{\|SGP_w(s,t)\|}{\|SP_w(s,t)\|}$ is at most 1.5, irrespective of the weight assignment.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Shortest Path, Tessellation, Weighted Region Problem

Funding P. B. is partially supported by NSERC. G. E., D. O. and R. I. S. are partially supported by H2020-MSCA-RISE project 734922 - CONNECT and project PID2019-104129GB-I00 funded by MCIN/AEI/10.13039/501100011033. G. E. and D. O. are also supported by PIUAH21/IA-062 and CM/JIN/2021-004. G. E. is funded by an FPU of the Universidad de Alcalá.

1 Introduction

Geometric shortest path problems are a class of computational geometry problems where the goal is to find an optimal path between two points s and t in a certain setting. An important shortest path problem is computing an optimal path in a geometric domain when the cost of traversing the domain varies depending on the region. The resulting metric is often called the *weighted region metric*, and the problem of computing a (weighted) shortest path between two points under this metric is known as the *weighted region problem* (WRP) [8].

In real-time applications where the WRP arises, like robotics [5, 10], gaming [7] or GIS [4], which usually require efficient and practical algorithms, the problem is simplified in two ways. First, the domain is approximated by using a (weighted) hexagonal tessellation $\mathcal{H}$. Secondly, an approximation is considered by computing a shortest path on a weighted graph associated to $\mathcal{H}$, called 3-*corner grid graph* $G_{3\text{corner}}$ [9]. In $G_{3\text{corner}}$, the vertex set is the set of corners of the tessellation, and each vertex is connected by an edge to its 3 neighboring vertices in the tessellation, see Figure 1.

The interior of a cell H_i has a weight $\omega_i \in \mathbb{R}_{\geq 0}$, so a segment π has cost $\omega_i\|\pi\|$ when traversing H_i and $\min\{\omega_i, \omega_j\}\|\pi\|$ when lying on the edge between H_i and H_j. A *shortest grid path* $SGP_w(s,t)$ between two vertices s and t is defined as the shortest path in $G_{3\text{corner}}$,

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

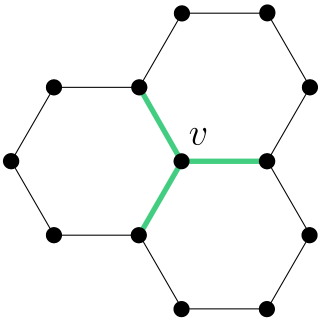


Figure 1 Vertex v is connected to its neighbors in $G_{3\text{corner}}$.

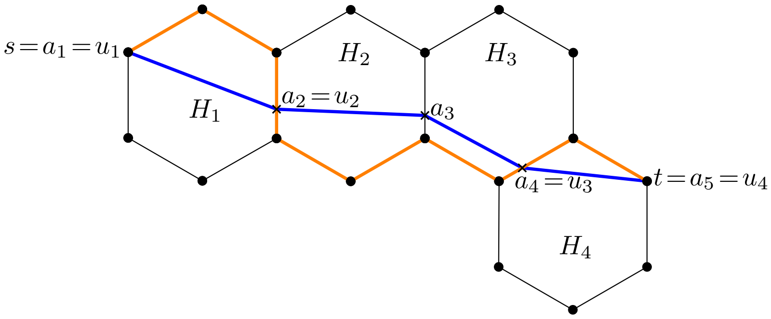


Figure 2 Shortest path $SP_w(s,t)$ (blue) and crossing path $X(s,t)$ (orange) from s to t.

which is considered as an alternative to a (standard) *shortest path* $SP_w(s,t)$ between s and t in the continuous weighted space. The aim of this work is to quantify the ratio $R = \frac{\|SGP_w(s,t)\|}{\|SP_w(s,t)\|}$.

2 Previous results

Nash [9] considered only weights in the set $\{1, \infty\}$ and obtained tight upper bounds in hexagonal, square, and triangular tessellations. When the weights of the cells are allowed to be in $\mathbb{R}_{>0}$, we are aware of some previous results. Jaklin [6] showed that $R \leq 2\sqrt{2}$ for square tessellations and another type of shortest path (with vertices at the center of the cells) . In addition, we recently proved upper bounds of $R = \frac{2}{\sqrt{3}}$ for weighted triangular cells [1], and $R = \frac{2}{\sqrt{2+\sqrt{2}}}$ for weighted square cells [2].

3 $\frac{\|SGP_w(s,t)\|}{\|SP_w(s,t)\|}$ ratio in $G_{3\text{corner}}$ for hexagonal cells

$SGP_w(s,t)$ and $SP_w(s,t)$ can be very different in both shape and length. Thus, we need to define a more convenient class of grid paths, called *crossing paths* $X(s,t)$. The key property of $X(s,t)$ is that we want it to traverse only the edges of the cells that $SP_w(s,t)$ traverses. Without loss of generality, we assume that $SP_w(s,t)$ is unique.

Let $(H_1, \ldots, H_n)$ be the ordered sequence of consecutive cells intersected by $SP_w(s,t)$ in $\mathcal{H}$. Let a_i and a_{i+1} be, respectively, the points where $SP_w(s,t)$ enters and leaves H_i, $i \in \{1, \ldots, n\}$, see Figure 2. The crossing path $X(s,t)$ is a shortest path along some edges of the cells intersected by $SP_w(s,t)$, ignoring weights. See Figure 3 for the subpath of $X(s,t)$ in H_i for different positions of a_i and a_{i+1}. The formal definition is deferred to the full

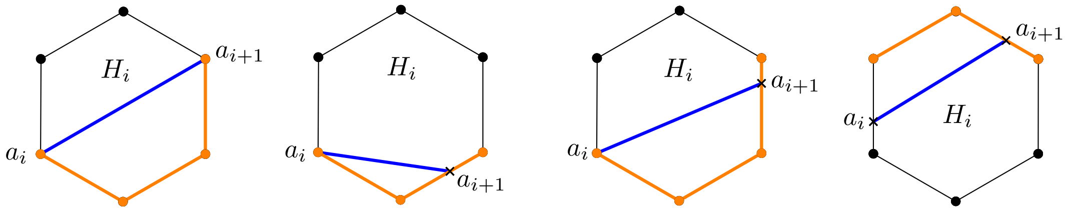


Figure 3 Subpaths of $X(s,t)$ (orange) and $SP_w(s,t)$ (blue) crossing H_i.

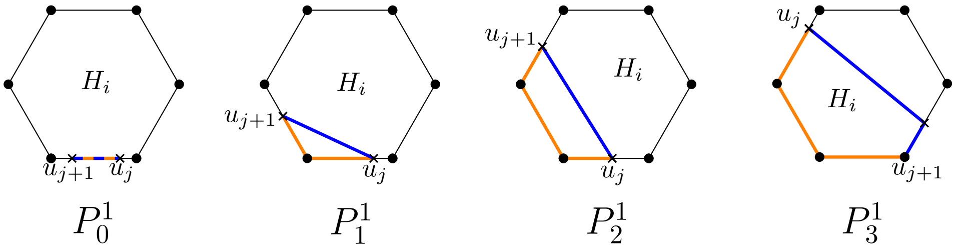


Figure 4 Weakly simple polygons P_0^1, P_1^1, P_2^1 and P_3^1, and the subpaths $SP_w(u_j, u_{j+1})$ (blue) and $X(u_j, u_{j+1})$ (orange) in a hexagonal tessellation.

version. Let $(s = u_1, \ldots, u_\ell = t)$ be a sequence of consecutive points where $X(s,t)$ and $SP_w(s,t)$ coincide. Applying the mediant inequality to $X(s,t)$ and $SP_w(s,t)$, we observe that the ratio $\frac{\|X(s,t)\|}{\|SP_w(s,t)\|}$ can be upper-bounded by the maximum among all the ratios $\frac{\|X(u_j,u_{j+1})\|}{\|SP_w(u_j,u_{j+1})\|}$, for $j \in \{1, \ldots, \ell - 1\}$.

The union of $SP_w(u_j, u_{j+1})$ and $X(u_j, u_{j+1})$ induces a weakly simple polygon [3]. We distinguish two types: in type P_k^1, $0 \le k \le 3$, the points u_j and u_{j+1} belong to the same cell, and $X(u_j, u_{j+1})$ and $SP_w(u_j, u_{j+1})$ intersect $k+1$ different edges, see Figure 4; in type P_k^b, $k \in \{1,2,3\}$, $SP_w(u_j, u_{j+1})$ intersects the interior of b cells, and $X(u_j, u_{j+1})$ and $SP_w(u_j, u_{j+1})$ intersect k different edges of the last cell whose interior is intersected by $SP_w(u_j, u_{j+1})$, see Figure 5. These are the only weakly simple polygons that can arise. So, our goal is to upper-bound the ratio $\frac{\|X(u_j,u_{j+1})\|}{\|SP_w(u_j,u_{j+1})\|}$ in each of the polygons.

We can see that $\frac{\|X(u_j,u_{j+1})\|}{\|SP_w(u_j,u_{j+1})\|} = 1$ in P_0^1. Also, P_3^b is a special case of P_1^{b+1}. Analogously, after some calculations, we are able to prove that the ratio in weakly simple polygons of type P_1^b and P_2^b, for $b \ge 2$, is maximized when $b = 2$. Thus, we get that the only relevant spanning ratios are those in the polygons of type P_1^1, P_2^1, P_1^2 and P_2^2.

$X(s,t)$ is defined based on the points where $SP_w(s,t)$ intersects the edges of the cells, so it might not be a shortest grid path. Hence, the ratio $\frac{\|X(u_j,u_{j+1})\|}{\|SP_w(u_j,u_{j+1})\|}$ could be larger than the ratio $\frac{\|SGP_w(s,t)\|}{\|SP_w(s,t)\|}$, see Figure 6. Thus, we define an additional class of grid paths, called *shortcut paths* $\Pi_i(s,t)$, see Figure 7. These paths intersect almost all the edges intersected by $X(s,t)$. So, by using $\Pi_i(s,t)$, we obtain a relation between the weights of the cells adjacent to H_i. Thus, they imply a better fit in case $SP_w(s,t)$ intersects two parallel edges, and allow us to find a tighter upper bound for the ratio $\frac{\|X(s,t)\|}{\|SP_w(s,t)\|}$.

We prove that if $\|X(s,t)\| \neq \|\Pi_i(s,t)\|$, then the weights of some cells can be modified so that the ratio between the shortest grid path and the shortest path increases. Using this fact, we proceed to calculate an upper bound on the ratio $\frac{\|X(u_j,u_{j+1})\|}{\|SP_w(u_j,u_{j+1})\|}$ in polygons of type P_1^1, P_2^1, and P_1^2 and P_2^2 when $\|X(s,t)\| = \|\Pi_i(s,t)\|$. Finally, since $\|X(s,t)\| \ge \|SGP_w(s,t)\|$, we obtain our main result. Moreover, this bound is tight, see Figure 8.

▶ **Theorem 1.** *In $G_{3corner}$, $\frac{\|SGP_w(s,t)\|}{\|SP_w(s,t)\|} \le 1.5$.*

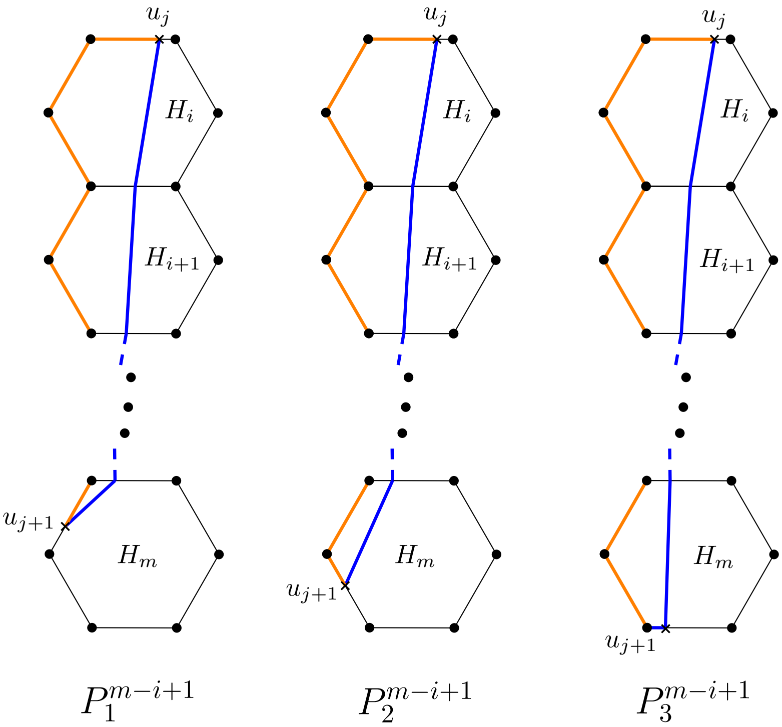


Figure 5 Weakly simple polygons P_1^{m-i+1}, P_2^{m-i+1} and P_3^{m-i+1}, and the subpaths of $SP_w(s,t)$ (blue) and $X(s,t)$ (orange) between u_j and u_{j+1} in a hexagonal tessellation.

References

1 P. Bose, G. Esteban, D. Orden, and R. I. Silveira. On approximating shortest paths in weighted triangular tessellations. *arXiv preprint arXiv:2111.13912*, 2021.

2 P. Bose, G. Esteban, D. Orden, and R. I. Silveira. Spanning ratio of shortest paths in weighted square tessellations. In *Abstracts of the 38th European Workshop on Computational Geometry (EuroCG)*, pages 65:1–8, 2022.

3 H. C. Chang, J. Erickson, and C. Xu. Detecting weakly simple polygons. In *Proceedings of the twenty-sixth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1655–1670. SIAM, 2014.

4 L. de Floriani, P. Magillo, and E. Puppo. Applications of computational geometry to geographic information systems. *Handbook of computational geometry*, 7:333–388, 2000.

5 D. Gaw and A. Meystel. Minimum-time navigation of an unmanned mobile robot in a 2-1/2D world with obstacles. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1670–1677. IEEE, 1986.

6 N. S. Jaklin. *On Weighted Regions and Social Crowds: Autonomous-agent Navigation in Virtual Worlds*. PhD thesis, Utrecht University, 2016.

7 A. Kamphuis, M. Rook, and M. H. Overmars. Tactical path finding in urban environments. In *First International Workshop on Crowd Simulation*. Citeseer, 2005.

8 J. S. Mitchell and C. Papadimitrou. The weighted region problem: Finding shortest paths through a weighted planar subdivision. *Journal of the ACM*, 38(1):18–73, 1991.

9 A. Nash. *Any-Angle Path Planning*. PhD thesis, University of Southern California, 2012.

10 N. C. Rowe and R. S. Ross. Optimal grid-free path planning across arbitrarily contoured terrain with anisotropic friction and gravity effects. *IEEE Transactions on Robotics and Automation*, 6(5):540–553, 1990.

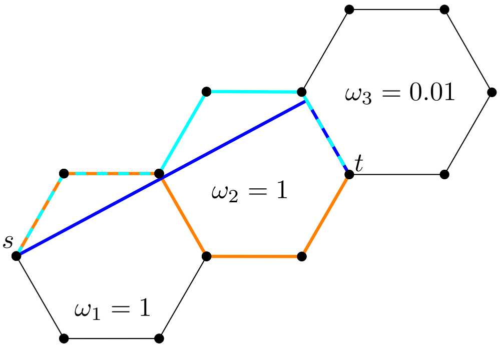


Figure 6 The ratio between the weight of $X(s,t)$ (orange) and the weight of $SP_w(s,t)$ (blue) is almost 1.44. The ratio between the weight of the cyan grid path and $SP_w(s,t)$ is ≈ 1.15.

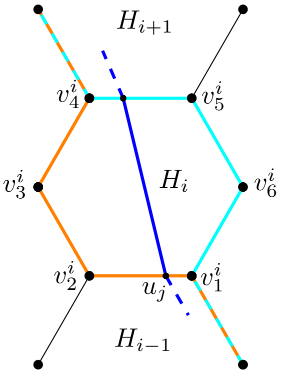


Figure 7 Subpaths of $SP_w(s,t)$ (blue), $X(s,t)$ (orange), $\Pi_i(s,t)$ (cyan). By the definition of the shortcut paths, $X(s,v_1^i) = \Pi_i(s,v_1^i)$, and $X(v_4^i,t) = \Pi_i(v_4^i,t)$.

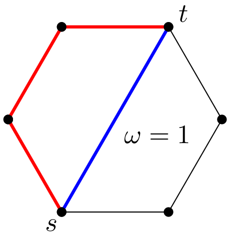


Figure 8 The ratio $\frac{\|SGP_w(s,t)\|}{\|SP_w(s,t)\|}$ is 1.5. $SP_w(s,t)$ is depicted in blue, and $SGP_w(s,t)$ in red.

Path-Connectivity of Fréchet Spaces of Graphs

Erin Chambers
St Louis University, USA

Brittany Terese Fasy
Montana State University, USA

Benjamin Holmgren
Montana State University, USA

Sushovan Majhi
University of California - Berkeley, USA

Carola Wenk
Tulane University, USA
[0]

Abstract

We examine topological properties of spaces of paths and graphs mapped to $\mathbb{R}^d$ under the Fréchet distance. We show that these spaces are path-connected if the map is either continuous or an immersion. If the map is an embedding, we show that the space of paths is path-connected, while the space of graphs only maintains this property in dimensions four or higher.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Graphs and surfaces; Mathematics of computing → Algebraic topology

Keywords and phrases Fréchet Distance, Path-Connectivity, Metric Space

Related Version A full version of the paper is available at https://benholmgren.github.io/ben-holmgren/assets/yrf2022.pdf.

Funding *Erin Chambers*: NSF grants CCF-1614562, CCF-1907612 and DBI-1759807.
Brittany Terese Fasy: NSF grants DMS-1664858, DMS-1854336, and CCF-2046730.
Benjamin Holmgren: The MSU Honors College Cameron Presidential Scholarship and NSF grants DMS-1664858 and CCF-2046730.
Carola Wenk: NSF grants CCF-1637576 and CCF-2107434.

Acknowledgements This work is an extension of [4]. We especially thank Maike Buchin, Pan Fang, Ellen Gasparovic, and Elizabeth Munch for their initial discussions on path-connectivity of spaces of graphs under the Fréchet distance.

1 Introduction

Motivated by the ubiquitous nature of one-dimensional data in a Euclidean ambient space (road networks in $\mathbb{R}^2$, for example), we investigate spaces of paths and graphs in $\mathbb{R}^d$. In particular, we examine these spaces in relation to the Fréchet distance, which is widely studied in the computational geometry literature [1–3, 5–7]. We work with three classes of paths: the set $\Pi_\mathcal{C}$ of all paths continuously mapped into $\mathbb{R}^d$, the set $\Pi_\mathcal{E}$ of paths embedded in $\mathbb{R}^d$, and the set $\Pi_\mathcal{I}$ of paths immersed in $\mathbb{R}^d$. In addition, we study three analogous spaces of graphs: the set $\mathcal{G}_\mathcal{C}$ of all graphs continuously mapped into $\mathbb{R}^d$, the set $\mathcal{G}_\mathcal{E}$ of graphs embedded in $\mathbb{R}^d$ and the set $\mathcal{G}_\mathcal{I}$ of graphs immersed in $\mathbb{R}^d$. See Figure 1 for examples of

[0] This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

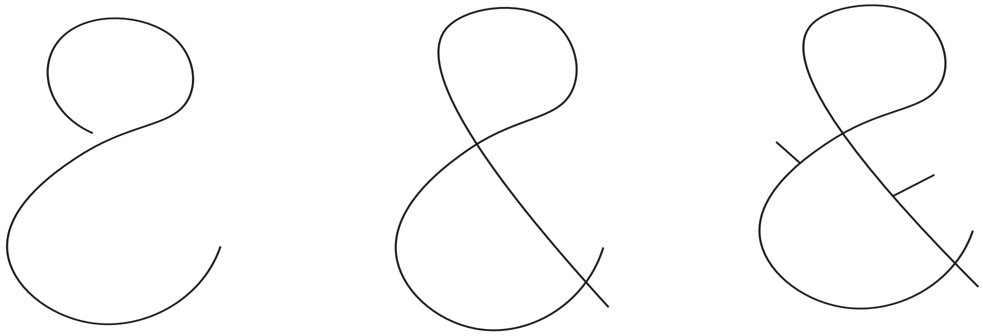


Figure 1 The images of an element in $\Pi_{\mathcal{E}}$, $\Pi_{\mathcal{I}}$, and $\Pi_{\mathcal{C}}$ respectively, mapped in $\mathbb{R}^2$.

paths in $\mathbb{R}^2$. We then topologize these sets using the open ball topology under the Fréchet distance, and study their path-connectedness property.

2 Background

We begin by defining the standard Fréchet distance for paths, adapting the definition from Alt and Godau [1]. Let $\alpha_0, \alpha_1 \in \Pi_{\mathcal{C}}$. The Fréchet distance between α_0 and α_1 is defined as:

$$d_{FP}(\alpha_0, \alpha_1) := \min_{r\colon [0,1]\to[0,1]} \max_{t\in[0,1]} |\alpha_0(t) - \alpha_1(r(t))|$$

Where r ranges over all reparameterizations of the unit interval (that is, homeomorphisms such that $r(0) = 0$ and $r(1) = 1$), and $|\cdot|$ denotes the standard Euclidean norm.

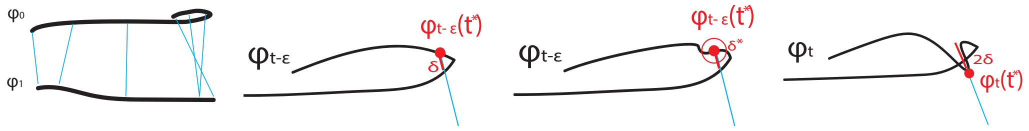
We now define the Fréchet distance for graphs, inspired by the Fréchet distance among paths. Let G be a one-dimensional simplicial complex, and let $\phi, \psi\colon G \to \mathbb{R}^d$ be continuous, rectifiable maps. Given any homeomorphism $h\colon G \to G$, we say that the *induced L_∞ distance* between the maps ϕ and $\psi \circ h$ is $||\phi - \psi \circ h||_\infty = \max_{x\in G} |\phi(x) - \psi(h(x))|$. With this distance in hand, we define the Fréchet distance between (G, ϕ) and (G, ψ) by minimizing over all homeomorphisms:[1]

$$d_{FG}\left((G,\phi),(G,\psi)\right) := \min_h ||\phi - \psi \circ h||_\infty$$

We now define and provide context for the underlying spaces that are studied in this work. Recall from above that $\Pi_{\mathcal{C}}$ denotes the set of all continuous mappings $\alpha : [0,1] \to \mathbb{R}^d$. The set $\Pi_{\mathcal{E}}$ of embedded paths in $\mathbb{R}^d$ results from further specifying that α is injective, and the set $\Pi_{\mathcal{I}}$ of immersed paths in $\mathbb{R}^d$ results from requiring only local injectivity of α. Note that $\Pi_{\mathcal{E}} \subsetneq \Pi_{\mathcal{I}} \subsetneq \Pi_{\mathcal{C}}$ and elements of $\Pi_{\mathcal{C}}, \Pi_{\mathcal{E}}$, and $\Pi_{\mathcal{I}}$ are deemed equivalent if the image of their underlying map α is equivalent, giving a path-Fréchet distance (denoted d_{FP}) of zero.

We define the analogous spaces of graphs, letting G be a one-dimensional simplicial complex and $\mathcal{G}_{\mathcal{C}}(G)$ denote the set of all continous mappings $\phi : G \to \mathbb{R}^d$. Similarly, we define the set of embeddings $\mathcal{G}_{\mathcal{E}}(G)$ with the added requirement that ϕ be injective, and the set of

[1] Other generalizations of the Fréchet distance minimize over all "orientation-preserving" homeomorphisms, which can be defined in several ways for stratified spaces. We drop this requirement in our definition.



(a) Interpolate **(b)** δ from crossing **(c)** Inflate δ^*-nbhd **(d)** Self-cross by 2δ

Figure 2 The sequence of moves to continuously conduct self crossings in $\Pi_{\mathcal{I}}$.

immersions $\mathcal{G}_{\mathcal{I}}$ with the requirement that ϕ need be only locally injective. Note that elements of $\mathcal{G}_{\mathcal{C}}, \mathcal{G}_{\mathcal{I}}$, and $\mathcal{G}_{\mathcal{E}}$ are equivalent (with graph Fréchet distance zero) if their underlying graphs belong to the same homeomorphism class, and if the image of their accompanying map ϕ is equivalent.

3 Results

▶ **Theorem 1** (Continuous Mappings). *The topological spaces of continuous mappings of paths* $(\Pi_{\mathcal{C}}, d_{FP})$ *and continuous mappings of graphs* $(\mathcal{G}_{\mathcal{C}}(G), d_{FG})$ *in* $\mathbb{R}^d$ *are path-connected.*

Proof Sketch. Let $\phi_0, \phi_1 \in \Pi_{\mathcal{C}}$. Naively, a path may be constructed from ϕ_0 to ϕ_1 by interpolating ϕ_0 to ϕ_1 along the pointwise matchings (so-called leashes) defining $d_{FP}(\phi_0, \phi_1)$. The same technique may be extended to demonstrate the path-connectivity of $\mathcal{G}_{\mathcal{C}}(G)$. ◀

▶ **Theorem 2** (Immersions). *The topological spaces of immersions of paths* $(\Pi_{\mathcal{I}}, d_{FP})$ *and immersions of graphs* $(\mathcal{G}_{\mathcal{I}}(G), d_{FG})$ *in* $\mathbb{R}^d$ *are path-connected.*

Proof Sketch. Let $\phi_0, \phi_1 \in \Pi_{\mathcal{I}}$, and construct a path $\Gamma : [0,1] \to \Pi_{\mathcal{I}}$ as in Theorem 1 by interpolating ϕ_0 to ϕ_1 along the pointwise matchings defining $d_{FP}(\phi_0, \phi_1)$. We next show that this is well defined. Suppose not, then, at some $t \in [0,1]$, $\phi_t = \Gamma(t)$ could create an intersection not present in ϕ_0. This may collapse an entire region of the image of ϕ_t, rendering ϕ_t no longer an immersion. Then, there exists $\epsilon > 0$ such that $\Gamma(t-\epsilon) = \phi_{t-\epsilon}$ has $t^* \in [0,1]$ where $\phi_{t-\epsilon}(t^*)$ is $\delta > 0$ away from a new self-intersection, and t^* comes sufficiently close to minimizing δ. At this time $t - \epsilon$, suspend interpolation along all leashes, and continuously inflate a small δ^*-neighborhood $\phi_{t-\epsilon}|_{(t^*-\delta^*, t^*+\delta^*)}$ about the point $\phi_{t-\epsilon}(t^*)$ in the image of $\phi_{t-\epsilon}$ so that the leash lengths for every point in the δ^*-neighborhood equal the leash length defined at $\phi_{t-\epsilon}(t^*)$. Then directly perturb $\phi_{t-\epsilon}(t^*)$ by 2δ along its unique leash such that the crossing at $\phi_{t-\epsilon}(t^*)$ occurs, and the crossing point defined by t^* again lies δ away from a self intersection, and 2δ away from its original position in the final image of ϕ_t. See Figure 2. Repeat the process for any subsequent crossings in the interpolation. An analogous path can be constructed for graphs. ◀

▶ **Theorem 3** (Path Embeddings). *The space* $(\Pi_{\mathcal{E}}, d_{FP})$ *is path-connected.*

Proof Sketch. Let $\phi_0, \phi_1 \in \Pi_{\mathcal{E}}$. There exists a canonical path from ϕ_0 to ϕ_1 by condensing each map toward its center until the images are "nearly straight", continuously mapping each image to a straight segment, and then interpolating as in Theorem 1. ◀

▶ **Theorem 4** (Graph Embeddings). *The topological space of graphs* $(\mathcal{G}_{\mathcal{E}}(G), d_{FG})$ *embedded in* $\mathbb{R}^d$ *is path-connected if* $d \geq 4$.

Proof Sketch. Examining the path-connectivity of $\mathcal{G}_{\mathcal{E}}$ under the Fréchet distance reduces to a knot theory problem for $d \leq 3$. For $d \geq 4$, there exists a sequence of Reidemeister moves from any tame knot to another. Hence, if $\phi_0, \phi_1 \in \mathcal{G}_{\mathcal{E}}$, we construct a path by interpolating along the pointwise matchings between ϕ_0 and ϕ_1 as in Theorem 1. If a self intersection would be created, we suspend interpolation elsewhere and conduct the corresponding Reidemeister move. Repeat the process for all intersections thereafter, until attaining the image of ϕ_1. ◀

▶ **Corollary 5** (Path-Connectivity of Metric Balls)**.** *Metric balls in the space* $\Pi_{\mathcal{C}}, \mathcal{G}_{\mathcal{C}}(G), \Pi_{\mathcal{I}}$, *and* $\mathcal{G}_{\mathcal{I}}(G)$ *are path-connected.*

Proof Sketch. Note that the techniques used in Theorem 1 and Theorem 2 never strictly increase the Frechet distance among two images of corresponding maps, so metric balls in each space are path-connected. For Theorem 2 this relies on the inflation step in Figure 2c, which assures that the Fréchet distance is fixed during a crossing event. The paths constructed in Theorem 3 and Theorem 4 do not necessarily maintain this property. ◀

References

1 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *IJCGA*, 5(1–2):75–91, 1995.

2 Kevin Buchin, Maike Buchin, and André Schulz. Fréchet distance of surfaces: Some simple hard cases. In *European Symposium on Algorithms*, pages 63–74. Springer, 2010.

3 Kevin Buchin, Tim Ophelders, and Bettina Speckmann. Computing the Fréchet distance between real-valued surfaces. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2443—-2455. ACM, 2017.

4 Maike Buchin, Erin Chambers, Pan Fang, Brittany Terese Fasy, Ellen Gasparovic, Elizabeth Munch, and Carola Wenk. Distances between immersed graphs: Metric properties, 2021.

5 Maike Buchin, Amer Krivosija, and Alexander Neuhaus. Computing the Fréchet distance of trees and graphs of bounded tree width. In *Proceedings of the 36th European Workshop on Computational Geometry*, 2020.

6 Erin Wolf Chambers, Éric Colin de Verdière, Jeff Erickson, Sylvain Lazard, Francis Lazarus, and Shripad Thite. Homotopic fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry*, 43(3):295–311, 2010. Special Issue on 24th Annual Symposium on Computational Geometry (SoCG'08). URL: `https://www.sciencedirect.com/science/article/pii/S0925772109000637`, `doi:https://doi.org/10.1016/j.comgeo.2009.02.008`.

7 Pan Fang and Carola Wenk. The Fréchet distance for plane graphs. In *Proceedings of the 37th European Workshop on Computational Geometry*, 2021.

Collapsing the Hidden-Set Convex-Cover Inequality

Reilly Browne ✉ 🏠
Stony Brook University, Stony Brook, NY, USA

Eric Chiu ✉ 🏠
Stony Brook University, Stony Brook, NY, USA

Abstract

We explore two closely related visibility problems: the maximum hidden set and the minimum convex cover. It is known that the convex cover number for any simple polygon is greater than or equal to the hidden set number. We present an explicit example where $hs(P) \neq cc(P)$ and provide subclasses where $hs(P) = cc(P)$ for all members. For histograms, we give a linear time algorithm which finds both a hidden set and a convex cover of the same size, improving from a 2008 result from Bajuelos et al. which finds the maximum hidden set when restricted to vertices and assuming general position.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Graph theory

Keywords and phrases Convex Cover, Clique Cover, Hidden set, Independent Set

Acknowledgements We want to thank Joseph Mitchell, Valentin Polishchuk, Christiane Schmidt, Kien Huynh and the rest of CG Group for discussing these problems with us.

1 Defining the problems

We consider the point visibility graphs (PVGs) of polygons. PVGs relate visibility problems to analogous graph theory problems using an infinite graph, with nodes being points in the plane and edges existing if two points see each other.

▶ **Definition 1.** *[[4]] Given a polygon P, the **point-visibility graph of P**, $PVG(P) = (V, E)$ where $V = \{p \mid p \in P\}$ and $E = \{(x, y) \mid \overline{xy} \subset P\}$.*

▶ **Definition 2.** *For a polygon P, the **hidden set number of P**, $hs(P)$, is the independence number of $PVG(P)$.*

▶ **Definition 3.** *For a polygon P, the **convex cover number of P**, $cc(P)$, is the minimum number of convex pieces needed to cover P. It is also the clique covering number of $PVG(P)$.*

▶ **Theorem 4** ([4])**.** *For all polygons P with n vertices: $1 \leq hs(P) \leq cc(P) \leq n - 2$.*

In this paper we introduce an explicit example where $hs(P) \neq cc(P)$ and show subclasses of simple polygons for which $hs(P) = cc(P)$ for all members. Polygons for which $hs(P) = cc(P)$, we refer to as homestead polygons. In personal communications with Joseph S.B. Mitchell and the Stony Brook CG group, we had conjectured that all simple polygons were homestead polygons, but our example disproves that claim.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

2 Existence of Non-Homesteads

We present a simple polygon, E (Figure 1), for which $hs(E) \neq cc(E)$. We also conjecture that the same holds for polygons resulting from the NP-hardness reduction of Shermer.

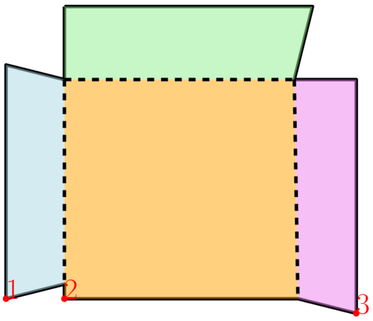


Figure 1 A convex cover of size 4 and hidden set of size 3 for the example nonhomestead.

▶ **Theorem 5.** *There exists a simple polygon that is not a homestead polygon.*

First we show that $hs(E) = 3$. We do so by finding that for every point, p, in E, the regions outside the visibility region of p can be covered with at most 2 convex pieces. Therefore, for every point in E, a hidden set including it can have at most 2 more points. We show this exhaustively, first by eliminating the intersections in convex covers of size 4 (cover with the 2 that the point is not a part of). In Figure 2, we use 2 distinct covers to eliminate points. For the remaining sections, we find the strongly visible regions and give 2 convex pieces which complete the cover, as demonstrated in Figure 3. Hence $hs(E) = 3$.

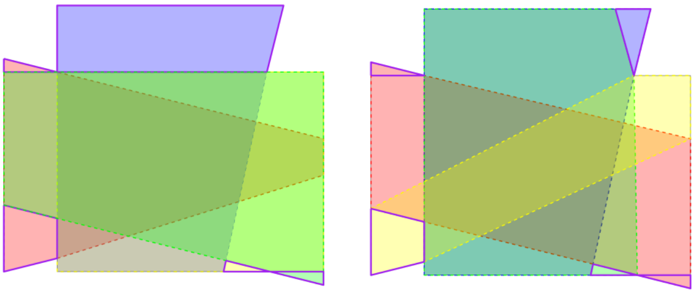


Figure 2 Two different convex covers of size 4. The regions covered only by one convex polygon across both are outlined in purple, as these are not yet eliminated.

We show that $cc(E) = 4$. Gella and Artes[2] show that the minimum clique cover of any induced subgraph of a graph G must be less than or equal to that of G. First we find an induced subgraph G of $PVG(E)$, points 1-7 in Figure 4. We then draw the complement of this, G' and determine its chromatic number, which is clearly 4 from the isomorphism. Since $\chi(G') = cc(G)$, we know that $cc(E) \geq cc(G) = 4$. Therefore $cc(E) = 4$.

3 Subclasses that are Homestead polygons

▶ **Theorem 6.** *For any spiral polygon S with r reflex vertices, $hs(S) = cc(S) = r + 1$.*

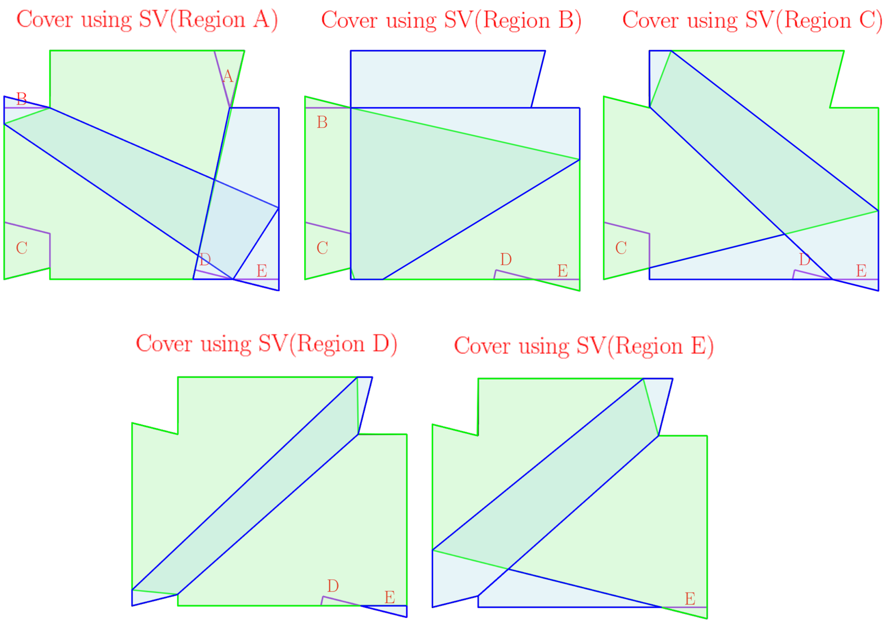


Figure 3 Strongly visible regions of subsections only need 2 convex pieces to cover E.

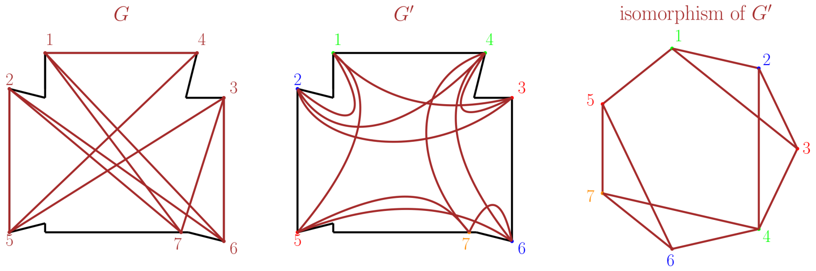


Figure 4 A subgraph G of PVG(E) with $cc(G) = 4$ ($\chi(G') = 4$, shown on right and bottom).

A spiral polygon is a simple polygon comprised of a reflex chain R and a convex chain C, with the reflex chain having r reflex vertices and 2 convex vertices that connect it to the convex chain. Shermer [4] showed $r + 1$ to be an upper bound for the convex cover and the hidden set in general. Bajuleos et al. [1] showed spirals admit a hidden set of size $r + 1$ by placing a hidden point at the midpoint of every edge in R. This means that the inequality collapses and $hs(S), cc(S)$ are both equal to $r + 1$. Shermer [5] provides a related result for generalized "j-visibility" (where points are visible from link distance j) where the size of a L_j-hidden vertex set in a spiral-like polygon is as a lowerbound on a L_j-convex cover.

▶ **Theorem 7.** *For any histogram polygon P, $hs(P) = cc(P)$.*

A histogram polygon is a simple polygon formed by two chains whose x-coordinates increase monotonically, one of which has only one edge, and where all angles between edges of the polygon are orthogonal. Bajuelos et al.[1] presented a formula for a maximum hidden *vertex* set of a histogram polygon in general position given the number of "bottom sides". Counting these bottom sides takes $O(n)$ time, implying a linear time algorithm. Our

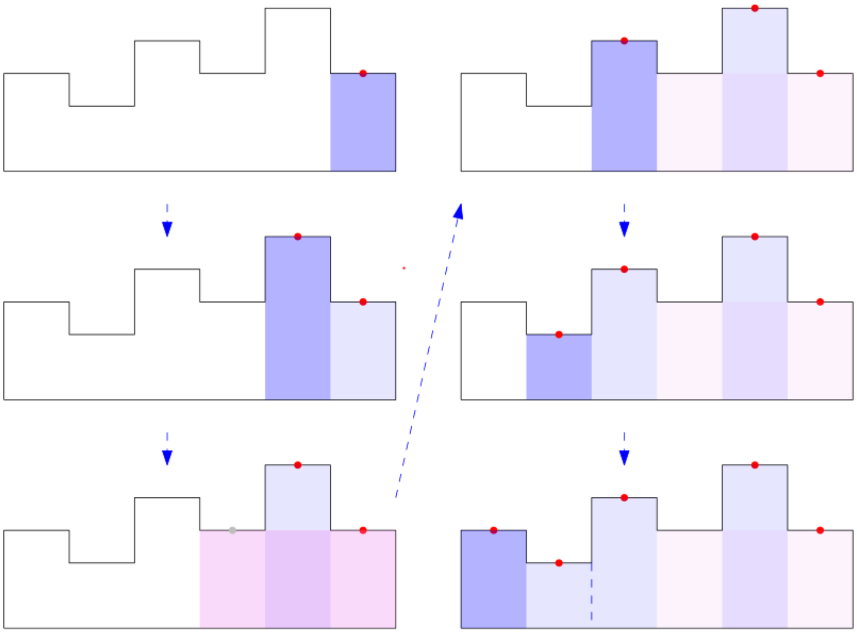


Figure 5 Demonstrating our algorithm for histogram polygons.

algorithm improves upon that result by finding a maximum hidden set (no vertex or position constraint) and a minimum convex cover in $O(n)$. Also, Hoorfar and Bagheri[3] present an $O(n)$ time algorithm for the related problem of finding minimum hidden guards in histogram polygons under orthogonal vision, meaning 3 of the 5 hiding problems presented by Shermer [4] can be solved in $O(n)$ time for histogram polygons.

We will consider the histogram polygon as an ordered list of “bars” from right to left. Each bar is the rectangle formed under each horizontal edge until the base is reached (a convex piece), paired with the midpoint of its top edge (a hidden point). We can decompose the polygon into these bars, creating an overestimate on convex cover and hidden set. To lower this, we merge bars of the same height without a boundary in between, which discards the newer hidden point and combines the rectangles.

Moving right to left, keep track of a set and a stack. The set will be our answer and the stack will keep track of all the bars which are candidates for merging. When considering a bar, we compare its height to the bars on the stack. If the bar is higher than the top of the stack or the stack is empty, we add it to the stack and the set. If the bar has equal height, we merge it with the top bar of the stack. If the bar has a lower height, then we pop the top bar off the stack and compare again. Continue until all bars have been considered and return the set of merged bars. See Figure 5 for an example run of the algorithm.

The result is a linear time algorithm (using the same stack operations argument as the Graham scan) which solves both maximum hidden set and minimum convex cover for histogram polygons. Since the algorithm always returns a hidden set and convex cover of the same size, histogram polygons must be homestead polygons.

4 Conclusion

We presented an explicit example where hidden set number and convex cover number are distinct and two classes of polygons for which they are always the same. Avenues for future work include finding more polygon subclasses that are homesteads and determining how large the gap can be. Currently, we suspect that monotone mountains in general are homesteads and that $hs(P) = O(cc(P))$ in simple polygons.

References

1 Antonio Leslie Bajuelos Domínguez, Santiago Canales Cano, Gregorio Hernández Peñalver, and Ana Mafalda Martins. Escondiendo puntos en espirales e histogramas. In *Actas de las VI Jornadas de Matemática Discreta y Algorítmica, JMDA'08*, España, 2008. Universidad de Lleida. URL: `https://oa.upm.es/4564/`.

2 Frederick S. Gella and Rosalio G. Artes. Clique cover of graphs. *Applied mathematical sciences*, 8:4301–4307, 2014.

3 Hamid Hoorfar and Alireza Bagheri. Minimum hidden guarding of histogram polygons, 2017. `arXiv:1708.05815`.

4 T. Shermer. Hiding people in polygons. *Computing*, 42(2-3):109–131, 1989. `doi:10.1007/bf02239742`.

5 Thomas C. Shermer. *Visibility Properties of Polygons*. PhD thesis, McGill University School of Computer Science, 1989.

Computing the length spectrum of combinatorial graphs on the torus

Matthijs Ebbens
Institut Fourier, CNRS, Université Grenoble Alpes, France

Francis Lazarus
G-SCOP, CNRS, Université Grenoble Alpes, France

Abstract

Let G be an unweighted and undirected graph of complexity n cellularly embedded on a torus. The computation of the length of the shortest non-trivial cycle of G is well-studied in the literature. We consider the more general problem of computing not only the length of the shortest non-trivial cycle, but also of the second shortest, the third shortest, and so on. The increasing list of lengths of the non-trivial cycles of G, where we only list the length of the shortest cycle in each free homotopy class, is called the *length spectrum* of G. In this paper, we describe an algorithm which, given the graph G and a positive integer k, computes the first k values of the length spectrum of G in time $O(n\sigma_1(k + \sqrt{kn\sigma_1}))$, where σ_1 is the length of the shortest non-trivial cycle of G.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Mathematics of computing → Graphs and surfaces; Mathematics of computing → Graph algorithms

Keywords and phrases Topological graph theory, computational topology, combinatorial surface, edge-width, systole, length spectrum

Funding This work is supported by the project TOFU of the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissement d'avenir.

1 Introduction

Combinatorial surfaces are a well-studied notion in computational topology and are usually represented as graphs that are cellularly embedded on a topological surface. Given a combinatorial surface, there exist several algorithms for computing the length of its shortest non-trivial cycle [7, 5, 2, 1]. Here, the length of a cycle is the sum of the weights of its edges if the edges are weighted, or the number of edges if not. However, relatively little is known about how to compute the *second shortest* non-trivial cycle, the *third shortest* and so on. In this paper we describe an algorithm to compute, for a given positive integer k, the lengths of the first k shortest cycles in the case where the surface is a torus and the graph is unweighted. After discussing some preliminaries in Section 2, we will state the main result in Section 3.

2 Preliminaries

Throughout this paper, $\mathbb{T}$ will denote a torus, i.e., a topological surface of genus 1 without boundary. Let G be an unweighted and undirected graph embedded on $\mathbb{T}$, where we allow G to have loop edges and multiple edges. We denote the complexity of G, i.e., the total number of its vertices and edges, by n. In this paper we assume that G is *cellularly embedded* on $\mathbb{T}$, which means that its faces are open disks. This embedding can be represented using one of the standard representations, e.g., the incidence graph of flags [4] or rotation systems [6]. To

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

each homotopy class (free or with a fixed basepoint) we associate a length, which is given by the length of the shortest cycle in that homotopy class. The *length spectrum* is defined as the list containing in increasing order the lengths of the free homotopy classes of G.

3 Main result

▶ **Theorem 1.** *Let G be a graph of complexity n cellularly embedded on a torus $\mathbb{T}$ and let k be a positive integer. The first k values of the length spectrum of G can be computed in time $O(n\sigma_1(k+\sqrt{kn\sigma_1}))$, where $\sigma_1 = O(n)$ is the length of the shortest non-trivial cycle of G.*

The idea of the algorithm is as follows. First, we compute a shortest non-trivial cycle ℓ_1 of G using one of the algorithms from the literature [1]. Cutting along ℓ_1 yields a cylinder with two copies of ℓ_1 as boundary components. For each vertex on one of the copies of ℓ_1 we compute a shortest path in the cylinder to the corresponding vertex on the other copy of ℓ_1. We denote the shortest of these shortest paths by ℓ_2 and cut along it to obtain a so-called *polygonal schema* of the torus (see Figure 1).

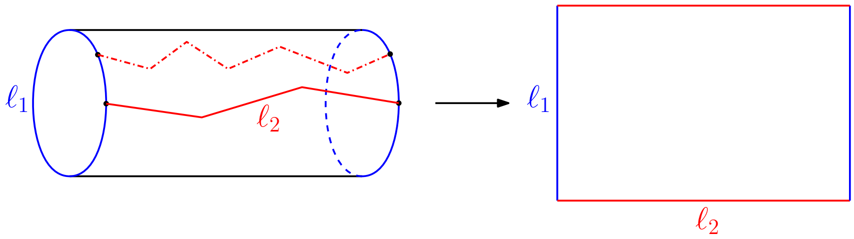


Figure 1 Constructing a polygonal schema of the torus.

By gluing infinitely many copies of this polygonal schema in a grid-like manner (see Figure 2), we obtain the infinite periodic graph $\tilde{G}$ in the universal cover $\mathbb{R}^2$ of $\mathbb{T}$. Let $\tilde{F}_0$ be a fixed copy of the polygonal schema in $\tilde{G}$. Consider the translation T_1 mapping the bottom side of $\tilde{F}_0$ to its top side and the translation T_2 mapping the left side of $\tilde{F}_0$ to its right side. Every copy of the polygonal schema in $\tilde{G}$ is the image of $\tilde{F}_0$ under a translation consisting of applying T_1 and/or T_2 (or their inverses) finitely many times.

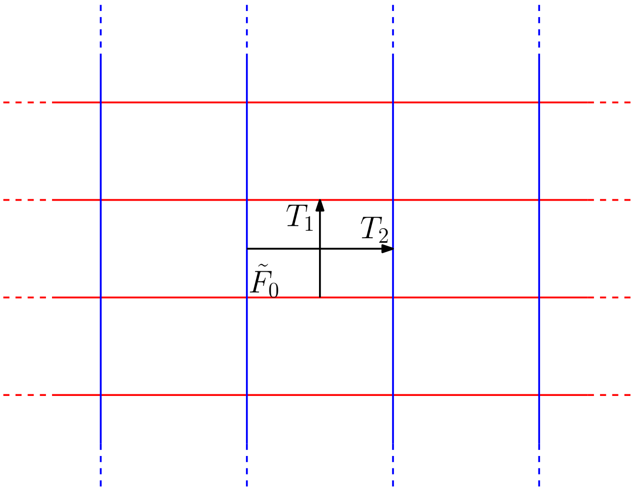


Figure 2 Tesselation of the universal cover $\mathbb{R}^2$ of $\mathbb{T}$ with copies of the polygonal schema.

A loop passing through a given vertex v of G corresponds to a path in $\tilde{G}$ from the representative $\tilde{v}_0$ of v contained in $\tilde{F}_0$ to the representative of v contained in some translate of $\tilde{F}_0$. Therefore, to obtain the length spectrum of loops passing through a given vertex v of G, we do a breadth-first search with the representative $\tilde{v}_0$ of v in $\tilde{F}_0$ as source. Note that it is not necessary (nor possible) to store the graph $\tilde{G}$ entirely; we can instead add new translates of $\tilde{F}_0$ as soon as they are discovered during the breadth-first search. In the following lemma we state an upper bound for the number of translates of $\tilde{F}_0$ that we need to visit in terms of the distance from the source and the number of translates of the source that we find.

▶ **Lemma 2.** *Let $\tilde{v} \in V(\tilde{G})$ and let r be a positive integer. Let N_{find} be the number of translates of $\tilde{v}$ within distance r of $\tilde{v}$ and let N_{search} be the number of translates of $\tilde{F}_0$ containing at least one vertex within distance r from $\tilde{v}$. Then $N_{\text{search}} \leq N_{\text{find}} + O(r)$.*

In the proof we use the notion of *horizontal*, *vertical* and *intermediate* translates (see Figure 3). In particular, we claim that if two horizontal or two vertical translates are within distance r of $\tilde{v}$, then all their intermediate translates are within distance r of $\tilde{v}$. Then the translates of $\tilde{F}_0$ that we need to visit and whose translate of $\tilde{v}$ has distance larger than r from $\tilde{v}$ are located either above a topmost translate, or left of a leftmost translate etc. and we show that the number of these translates can be upper bounded by $O(r)$.

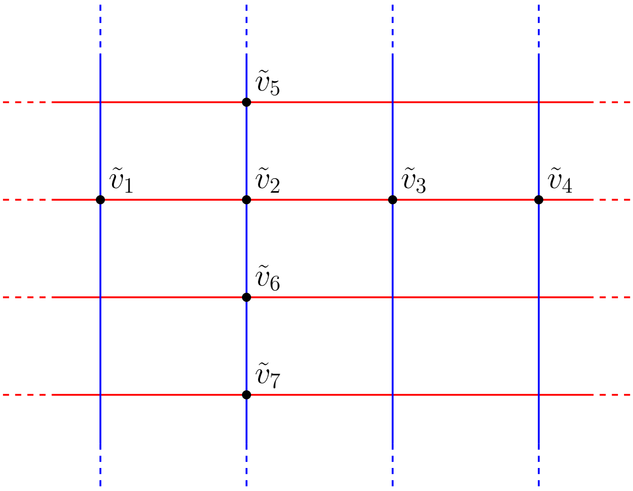


Figure 3 In the figure, $\tilde{v}_1$ and $\tilde{v}_4$ are horizontal translates with intermediate translates $\tilde{v}_2$ and $\tilde{v}_3$. Similarly, $\tilde{v}_5$ and $\tilde{v}_7$ are vertical translates with intermediate translates $\tilde{v}_2$ and $\tilde{v}_6$.

The next lemma provides an upper bound for the distance up to which we have to search.

▶ **Lemma 3.** *The $2k$ closest translates of a representative $\tilde{v} \in V(\tilde{G})$ of a vertex on ℓ_1 have distance $O(\sqrt{kn\sigma_1})$ from $\tilde{v}$.*

In the proof we show by an explicit construction that given a positive integer r, there are $\Omega(r^2\sigma_1^{-1}\sigma_2^{-1})$ translates of $\tilde{v}$ within distance r from $\tilde{v}$, where σ_2 is the length of ℓ_2. Combining Lemmas 2 and 3 we see that the number of copies of $\tilde{F}_0$ that we have to visit to find the $2k$ closest translates to a representative of a vertex on ℓ_1 is $O(k + \sqrt{kn\sigma_1})$. This yields the k shortest homotopy classes of loops, since for each loop that we find we also find the loop with the opposite orientation. Because each copy of $\tilde{F}_0$ has complexity n, this step takes time $O(n(k + \sqrt{kn\sigma_1}))$. Observe that every cycle of G that does not intersect ℓ_1 is homotopic to a multiple of $[\ell_1]$. Since ℓ_1 is the shortest cycle in its homotopy class, so are its multiples [3, Proposition 2.5]. Therefore, it is sufficient to take the vertices of ℓ_1 as sources for the breadth-first search instead of all vertices of $\tilde{F}_0$. Finally, the length spectrum of G is obtained by ordering the length spectra of loops based at the different vertices.

References

1 Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Algorithms for the edge-width of an embedded graph. *Computational Geometry*, 45(5-6):215–224, 2012.

2 Sergio Cabello and Bojan Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Discrete & Computational Geometry*, 37(2):213–235, 2007.

3 Éric Colin De Verdière and Jeff Erickson. Tightening nonsimple paths and cycles on surfaces. *SIAM Journal on Computing*, 39(8):3784–3813, 2010.

4 David Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 599–608, 2003.

5 Jeff Erickson and Sariel Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.

6 Bojan Mohar and Carsten Thomassen. *Graphs on surfaces*, volume 10. JHU press, 2001.

7 Carsten Thomassen. Embeddings of graphs with no short noncontractible cycles. *Journal of Combinatorial Theory, Series B*, 48(2):155–177, 1990.

Forbidding Edges between Points in the Plane to Disconnect the Triangulation Flip Graph

Reza Bigdeli
Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

Anna Lubiw
Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

Abstract

The ***flip graph*** for a set P of points in the plane has a vertex for every triangulation of P, and an edge when two triangulations differ by one ***flip*** that replaces one triangulation edge by another. The flip graph is known to be connected even if some triangulation edges are *constrained* to be used. We study connectivity of the flip graph when some triangulation edges are *forbidden*.

A set X of edges between points of P is a ***flip cut set*** if eliminating all triangulations that contain edges of X results in a disconnected flip graph. If X is a single edge it is called a ***flip cut edge***. The ***flip cut number*** of P is the minimum size of a flip cut set. We give an algorithm to test if an edge is a flip cut edge. For a set of n points in convex position (whose flip graph is the 1-skeleton of the associahedron) we prove that the flip cut number is $n-3$.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Combinatorial algorithms

Keywords and phrases triangulations, flip graph, reconfiguration, flip cut edge, flip cut number, associahedra, connectivity, convex polygon, algorithm, computational geometry, graph

1 Introduction

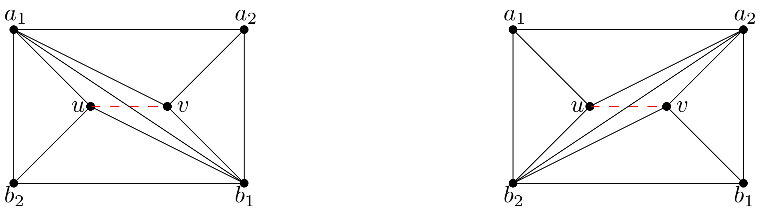
Given a set P of n points in the plane, which may include collinear points, an ***edge*** of P is a line segment pq that intersects P in exactly the two endpoints p and q. A ***triangulation*** of P is maximal set of non-crossing edges. Triangulations have important applications in graphics and mesh generation [2, 10] and are of significant mathematical interest [9].

A fundamental approach to understanding triangulations is by means of *flips*. A ***flip*** operates on a triangulation by removing one edge pq and adding another edge uv to obtain a new triangulation—of necessity, the edges pq and uv will cross and their four endpoints will form a convex quadrilateral with no other points of P inside it. For example, in Figure 1, edge a_1b_1 can be flipped to uv. In 1972, Lawson [12, 13] proved that any triangulation of point set P can be reconfigured to any other triangulation of P by a sequence of flips. This can be expressed as connectivity of the ***flip graph***, which has a vertex for every triangulation of P and an edge when two triangulations differ by a flip.

Although reconfiguring triangulations via flips is well studied [4], there are some very interesting open questions, and many properties of flip graphs remain to be discovered.

The case of points in convex position is especially interesting because there is a bijection between flips in triangulations of a convex point set and rotations in binary trees [18]. Finding the rotation distance between two binary trees is of great interest in biology for phylogenetic trees [8], and in data structures for splay trees [18]. Furthermore, the flip graph for n points in convex position is the 1-skeleton of an $(n-3)$-dimensional polytope called the *associahedron* [14], or see [6]. See Figure 2. Although there is no geometric analogue of the

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

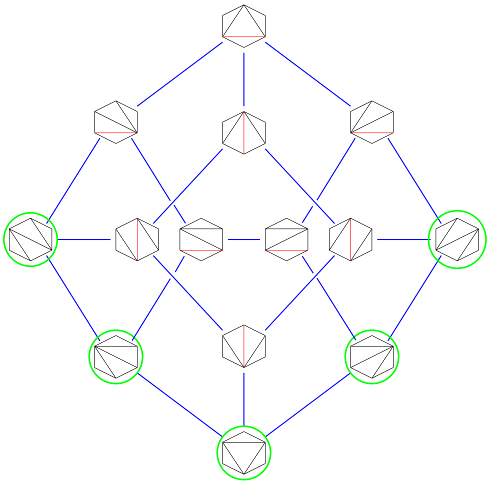


■ **Figure 1** The smallest point set that has a flip cut edge. The edge $e = uv$ is a flip cut edge since forbidding e leaves two possible triangulations (as shown) and neither one allows a flip.

associahedron for the case of triangulations of a general point set, some of its properties carry over to an abstract complex called the *flip complex*. For example, the 2-dimensional faces of the flip complex, like those of the associahedron, have size 4 or 5 [15].

An open frontier in the study of flip graphs has to do with expander properties, which would potentially lead to rapid mixing via random flips. For results on mixing in triangulations, see [5, 16, 17]. More generally, researchers study connectivity properties of flip graphs. Recently, Wagner and Welzl [19] showed that for n points in general position in the plane, the flip graph is $\lceil \frac{n}{2} - 2 \rceil$-connected. For points in convex position, the flip graph is $(n-3)$-connected, which follows from Balinski's theorem [1] applied to the 1-skeleton of the associahedron, see [19].

One intriguing thing about flip graphs of triangulations is that many properties carry over when we restrict to triangulations containing some specified non-crossing edges—so-called *constrained triangulations*. The subgraph of the flip graph consisting of triangulations that contain all the constrained edges is connected [7].



■ **Figure 2** The flip graph of points of a convex hexagon is the 1-skeleton of an associahedron. If we forbid the two red edges, the resulting flip graph (with vertices circled in green) is connected.

Our Results. We study connectivity properties of the flip graph when—instead of constraining certain edges between points to be present—we *forbid* certain edges between points. To be precise, if a set X of edges between points is forbidden, we eliminate all triangulations that contain an edge of X, and examine whether the flip graph on the remaining triangulations is connected. We say that X is a ***flip cut set*** if the resulting flip graph is disconnected; in the special case where X is a single edge, we say that the edge is a ***flip cut edge***. For example the edge uv in Figure 1 is a flip cut edge, but the two red edges in Figure 2 do not form a flip cut set. Also see Figures 3, 4. We define the ***flip cut number*** of a set of points to be the minimum size of a flip cut set. This is analogous to the *connectivity* of a graph—the minimum number of vertices whose removal disconnects the graph.

Since the structure of the flip graph depends on the edges between the points, it seems more natural to study connectivity of the flip graph after deleting some of these edges, rather than deleting some vertices of the flip graph, as standard graph connectivity does, and as the result of Wagner and Welzl [19] does.

As our main result, we characterize when an edge e is a flip cut edge in terms of connectivity (in the usual graph sense) of the edges that cross e. We then use the characterization to give an $O(n \log n)$ time algorithm to test if a given edge e in a point set of size n is a flip cut edge. With that algorithm as preprocessing, we give a linear time algorithm to test if two triangulations are still connected after we eliminate from the flip graph all triangulations containing edge e.

For the case of n points in convex position, there are no flip cut edges and we show that the flip cut number is $n - 3$. For example, in Figure 2 the leftmost and rightmost triangulations become disconnected if we forbid one more edge, which yields a flip cut set of size 3 for $n = 6$.

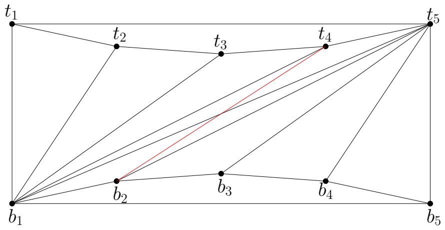


Figure 3 The "channel", and a triangulation that becomes frozen (an isolated vertex in the flip graph) if we forbid the edge b_2, t_{n-1} (in red). In fact, every edge $b_i t_j$, $i, j \notin \{1, 5\}$ is a flip cut edge.

We show that a point set of size n may have $\Theta(n^2)$ flip cut edges (see Figure 3), and we show that a flip cut edge may result in $\Theta(n)$ disconnected components in the flip graph. We also examine various special point sets whose flip graphs have been previously studied, such as points on an integer grid [5] and, more generally, point sets without empty convex pentagons [11]. Our characterization of flip cut edges becomes simpler in the absence of empty convex pentagons. Point sets without empty convex pentagons must have collinear points; our results do not assume points in general position.

For further details see the arxiv version [3].

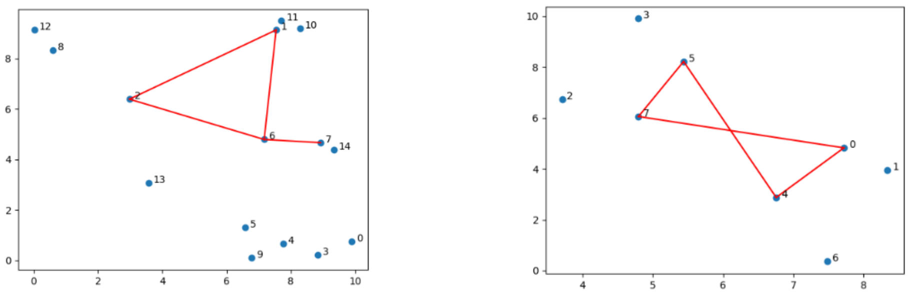


Figure 4 Some point sets and their flip cut edges (in red).

References

1 Michel L Balinski. On the graph structure of convex polyhedra in n-space. *Pacific Journal of Mathematics*, 11(2):431–434, 1961. doi:10.2140/pjm.1961.11.431.

2 Marshall Bern and David Eppstein. Mesh generation and optimal triangulation. *Computing in Euclidean Geometry*, pages 47–123, 1995. doi:10.1142/9789814355858_0002.

3 Reza Bigdeli and Anna Lubiw. Disconnecting the triangulation flip graph of points in the plane by forbidding edges, 2022. arxiv paper to appear.

4 Prosenjit Bose and Ferran Hurtado. Flips in planar graphs. *Computational Geometry*, 42(1):60–80, 2009. doi:10.1016/j.comgeo.2008.04.001.

5 Pietro Caputo, Fabio Martinelli, Alistair Sinclair, and Alexandre Stauffer. Random lattice triangulations: Structure and algorithms. *The Annals of Applied Probability*, 25(3):1650–1685, 2015. doi:10.1214/14-aap1033.

6 Cesar Ceballos, Francisco Santos, and Günter M Ziegler. Many non-equivalent realizations of the associahedron. *Combinatorica*, 35(5):513–551, 2015. doi:10.1007/s00493-014-2959-9.

7 L. Paul Chew. Constrained Delaunay triangulations. *Algorithmica*, 4(1):97–108, 1989. doi:10.1145/41958.41981.

8 Bhaskar DasGupta, Xin He, Tao Jiang, Ming Li, John Tromp, and Louxin Zhang. On distances between phylogenetic trees. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 97, pages 427–436. SIAM, 1997. doi:10.5555/314161.314338.

9 Jesús De Loera, Jörg Rambau, and Francisco Santos. *Triangulations: Structures for Algorithms and Applications*, volume 25. Springer Science & Business Media, 2010. doi:10.1007/978-3-642-12971-1.

10 Herbert Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge University Press, 2001. doi:10.1017/CBO9780511530067.

11 David Eppstein. Happy endings for flip graphs. *Journal of Computational Geometry*, 1(1):3–28, 2010. doi:10.48550/arXiv.cs/0610092.

12 C. L. Lawson. Generation of a triangular grid with application to contour plotting. Technical report, Memo 299, Jet Propulsion Laboratory, 1972.

13 C. L. Lawson. Transforming triangulations. *Discrete Mathematics*, 3(4):365–372, 1972. doi:10.1016/0012-365X(72)90093-3.

14 Carl W. Lee. The associahedron and triangulations of the n-gon. *European Journal of Combinatorics*, 10(6):551–560, 1989. doi:10.1016/S0195-6698(89)80072-1.

15 Anna Lubiw, Zuzana Masárová, and Uli Wagner. A proof of the orbit conjecture for flipping edge-labelled triangulations. *Discrete & Computational Geometry*, 61(4):880–898, 2019. doi:10.1007/s00454-018-0035-8.

16 Michael Molloy, Bruce Reed, and William Steiger. On the mixing rate of the triangulation walk. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 43, 2001. doi:10.1090/dimacs/043/11.

17 Dana Randall and Prasad Tetali. Analyzing Glauber dynamics by comparison of Markov chains. *Journal of Mathematical Physics*, 41(3):1598–1615, 2000. doi:10.1063/1.533199.

18 Daniel D Sleator, Robert E Tarjan, and William P Thurston. Rotation distance, triangulations, and hyperbolic geometry. *Journal of the American Mathematical Society*, 1(3):647–681, 1988. doi:10.1145/12130.12143.

19 Uli Wagner and Emo Welzl. Connectivity of triangulation flip graphs in the plane (Part I: Edge flips). In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2823–2841, 2020. doi:10.1137/1.9781611975994.172.

The tropical variety of antisymmetric matrices

Luis Crespo Ruiz
Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria, 39005 Santander, Spain

Francisco Santos
Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria, 39005 Santander, Spain

Abstract

The k-asociahedron is a simplicial complex whose facets correspond to k-triangulations of the n-gon, known to be homeomorphic to a sphere of dimension $k(n-2k-1)-1$. We show that it can be obtained intersecting the tropical variety of Pfaffians with the orthant of "4-point positive" weights.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Multitriangulations, tropical geometry, matchings, Pfaffians, polytopes.

Related Version A full version of the paper is available at `https://arxiv.org/abs/2203.04633`.

Funding Supported by grant PID2019-106188GB-I00/AEI/10.13039/501100011033 of the Spanish Research Agency and by project CLaPPo (21.SI03.64658) of Universidad de Cantabria and Banco Santander

1 k-triangulations

▶ **Definition 1** (*See e.g. [3]*). *Let $n > 2k$ be two positive integers. A subset $T \subseteq \binom{[n]}{2}$ of diagonals of the n-gon is called $(k+1)$-free if no $(k+1)$ diagonals in T mututally cross. The maximal $(k+1)$-free graphs are called* k-triangulations *or multi-triangulations.*

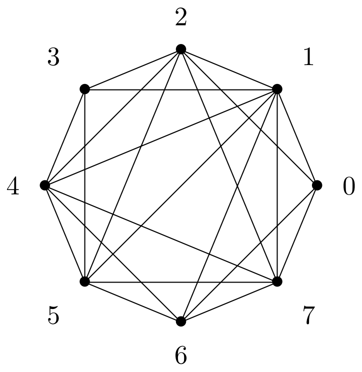


Figure 1 A 2-triangulation of the 8-gon. As expected, it has 22 edges

We are interested in the abstract simplicial complex $\mathcal{A}ss_k(n)$ on the vertex set $\binom{[n]}{2}$ whose faces are $(k+1)$-free graphs and whose facets are k-triangulations. All k-*triangulations* are known to have cardinality $k(2n-2k-1)$ [3, 5]. That is, $\mathcal{A}ss_k(n)$ is a pure simplicial complex.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

If an edge $\{i,j\}$ has $|i-j| \leq k$ (where indices are taken modulo n, and distance is measured cyclically), then it lies in every k-triangulation. We call these edges *irrelevant* and call the face of $\mathcal{A}ss_k(n)$ they span the *irrelevant face*. We can thus define the reduced complex, $\overline{\mathcal{A}ss}_k(n)$, the faces of which are the $(k+1)$-free sets of relevant edges. The exact relation between $\mathcal{A}ss_k(n)$ and $\overline{\mathcal{A}ss}_k(n)$ is that the former is the join of the latter with the irrelevant face, and hence the latter is the link of the former at the irrelevant face. Based on the fact that $\overline{\mathcal{A}ss}_1(n)$ is the face poset of the polar of the standard associahedron we define:

▶ **Definition 2.** *We call* $\overline{\mathcal{A}ss}_k(n)$ *the* k-associahedron *of parameters* n, k. *We refer to* $\mathcal{A}ss_k(n)$ *as the* extended k-associahedron.

Jonsson [5] proved that $\overline{\mathcal{A}ss}_k(n)$ is a shellable simplicial sphere of dimension $k(n-2k-1)-1$, and conjectured it to be polytopal. This conjecture is one of the motivations for this work.

2 Tropical varieties

Let $f \in \mathbb{K}[x_1, \ldots, x_N]$ be a polynomial. Each vector $d \in \mathbb{R}^N$, considered as giving weights to the variables, defines an *initial form* $\mathrm{in}_d(f)$, obtained neglecting in f the monomials of non-maximum weight. If d is "generic" then $\mathrm{in}_d(f)$ is a single monomial, but we are interested in the opposite case. The *tropical hypersurface of* f, denoted $\mathrm{trop}(f)$, is the set of d's for which at least two monomials attain the maximum weight. Put differently, $\mathrm{trop}(f) \subset \mathbb{R}^N$ equals the codimension-one skeleton of the normal fan of the Newton polytope of f [9].

▶ **Definition 3.** *A* tropical prevariety *is any finite intersection of tropical hypersurfaces. The* tropical variety *of an ideal* $I \subset \mathbb{K}[x_1, \ldots, x_N]$ *is* $\mathrm{trop}(I) := \cap_{f \in I} \mathrm{trop}(f)$.

Although a tropical variety is defined as an infinite intersection of tropical hypersurfaces, for each I a certain finite subset is enough; that is, every tropical variety is a prevariety [9, Thm. 2.6.5]. But not every generating set of I is enough, not even a universal Gröbner basis.

3 The ideal of Pfaffians of degree $k+1$

The determinant of an antisymmetric matrix of even size $2k$ with indeterminate entries is the perfect square of a homogeneous polynomial of degree k, called the *Pfaffian*. It has $2k!!$ monomials, corresponding to the (perfect) matchings among the $2k$ labels for rows and columns, all with coefficient ± 1 depending on the *parity* of each matching [1, 8, 12]. For example, for $k=2$ we get

$$\begin{vmatrix} 0 & x_{12} & x_{13} & x_{14} \\ -x_{12} & 0 & x_{23} & x_{24} \\ -x_{13} & -x_{23} & 0 & x_{34} \\ -x_{14} & -x_{24} & -x_{34} & 0 \end{vmatrix} = (x_{12}x_{34} - x_{13}x_{24} + x_{14}x_{23})^2. \tag{1}$$

The 3 terms inside the square correspond to the 3 matchings among 4 points, shown in Fig. 2.

For each $n \geq 2k+2$, let $I_k(n)$ be the ideal in $\mathbb{K}[x_{i,j}, \{i,j\} \in \binom{[n]}{2}]$ generated by all the *principal Pfaffians of degree* $k+1$ of an antisymetric matrix M of size n. That is, for each subset $U \in \binom{[n]}{2k+2}$ we consider the Pfaffian of the principal minor of M labeled by U. Since Pfaffians of degree two (see Eq. (1)) coincide with the quadratic Plücker relations, $I_1(n)$ equals the Plücker ideal defining the Grassmannian $\mathcal{G}r_2(n)$ in $\mathbb{K}^{\binom{[n]}{2}}$. See, e.g., [11, Remark 3.22]. In this $k=1$ case Speyer and Williams [13] have shown that the associahedron arises as a subfan of the tropical variety of $I_1(n)$, as we now recall.

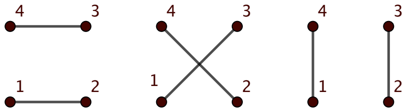


Figure 2 The three matchings among four points, corresponding to the Pfaffian in Eq. (1)

▶ **Definition 4.** *We say that a weight vector* $v \in \mathbb{R}^{\binom{[n]}{2}}$ *is* four-point positive *(abbreviated* fp-positive*) if for all* $1 \le a < a' < b < b' \le n$ *we have that*

$$v_{a,b} + v_{a',b'} \ge \max\{v_{a,a'} + v_{b,b'}, v_{a,b'} + v_{a',b}\}. \qquad (2)$$

We denote by FP_n *the subset of* $\mathbb{R}^{\binom{[n]}{2}}$ *consisting of fp-positive vectors.*

Observe that FP_n equals the space of weights that select as initial term, in each Pfaffian of four points, the matching that has a crossing. It can also be interpreted as the weights that represent *separation vectors* among sides of the n-gon, or as weights that are monotone with respect to crossing-increasing swaps among perfect matchings of each U. FP_n is a cone linearly isomorphic to an orthant plus a linear space. See details in [2, Sect. 3.1].

▶ **Theorem 5** ([13, Section 5]). *The intersection* $\mathrm{trop}(I_1(n)) \cap \mathrm{FP}_n$ *is a simplicial fan isomorphic to (the cone over) the extended associahedron* $\mathcal{A}ss_1(n)$*.*

4 Our results

Our main result generalizes Theorem 5 to arbitrary k. The starting point is to show that Pfaffians form a Gröbner basis of the ideal they generate, for any weight vector in FP_n. This generalizes the main result of [6], who prove it for a particular lexicographic weight vector:

▶ **Theorem 6.** *With respect to any weight vector* $v \in \mathrm{FP}_n$*, Pfaffians are a Gröbner basis for the ideal* $I_k(n)$*. Moreover, if* v *lies in the interior of* FP_n *then* $\mathrm{in}_v(I_k(n))$ *is the monomial ideal generated by* $(k+1)$*-crossings. That is, it is the Stanley-Reisner ideal [10] of* $\mathcal{A}ss_k(n)$*.*

This implies that k-triangulations are bases of the algebraic matroid of $\mathcal{P}f_k(n)$, which in turn coincides with the hyperconnectivity matroid [7]. See [2, Section 2.3] for more details.

We now denote $\mathcal{V}_k(n) \subset \mathbb{R}^{\binom{[n]}{2}}$ the intersection of the tropical hypersurfaces of Pfaffians of degree k. This is by definition a tropical *prevariety* containing $\mathrm{trop}(I_k(n))$, but it may not coincide with it. In the light of Theorem 6, it makes sense to look at the part of $\mathcal{V}_k(n)$ defined by fp-positive vectors. That is, we define $\mathcal{V}_k^+(n) := \mathcal{V}_k(n) \cap \mathrm{FP}_n$.

We prove that for any $v \in \mathrm{FP}_n$, being in $\mathcal{V}_k^+(n)$ is equivalent to the positivity equations (2) being satisfied with equality except in a $(k+1)$-free set [2, Theorem 4.3]. Moreover, when this happens v can be proved to be in $\mathrm{trop}(I_k(n))$ [2, Corollary 4.5]. Thus:

▶ **Theorem 7.** 1. $\mathcal{V}_k^+(n) = \mathrm{trop}(I_k(n)) \cap \mathrm{FP}_n$.
2. $\mathcal{V}_k^+(n)$ *is the union of the faces of the orthant* FP_n *corresponding to* $(k+1)$*-free graphs.*

That is, $\mathcal{V}_k^+(n)$ is embedded in FP_n as (the cone over) the k-associahedron $\mathcal{A}ss_k(n)$. In the full version of this abstract [2, Section 5] we argue that, once we have this result, in order to realize the k-associahedron as a complete (and hopefully polytopal) fan it would suffice to find a projection $\mathbb{R}^{\binom{[n]}{2}} \to \mathbb{R}^{k(2n-2k-1)}$ that is injective in $\mathcal{V}_k^+(n)$. This idea works nicely for the case $k = 1$, but we have not managed to implement it for higher k.

A projection that works for $k = 1$ consists in choosing a triangulation T_0 and forgetting the coordinates of edges that are not in T_0. The resulting fan equals the $\mathbf{g}$-vector fan in the root system of type A, a realization of the associahedron constructed before by Hohlweg, Pilaud and Stella in [4]. In this realization the vectors for the edges of T_0 are projected into basis vectors, the same edges rotated one position are projected into the negative basis vectors, and the rest of edges are projected into $\{-1, 0, 1\}$ combinations of them.

References

1 Arthur Cayley. On the theory of permutants. Cambridge and Dublin Mathematical Journal. VII (1852) 40–51. Reprinted in Collected mathematical papers, volume 2.

2 Luis Crespo Ruiz, Francisco Santos. Multitriangulations and tropical Pfaffians. Preprint, 29 pages, March 2022. arXiv:2203.04633

3 Andreas Dress, Jack H. Koolen and Vincent Moulton. On line arrangements in the hyperbolic plane. *Eur. J. Comb.*, **23**(5) (2002), 549–557.

4 Christophe Hohlweg, Vincent Pilaud, Salvatore Stella, Polytopal realizations of finite type $\mathbf{g}$-vector fans, *Adv.Math.* **328** (2018), 713-749, https://doi.org/10.1016/j.aim.2018.01.019.

5 Jakob Jonsson. Generalized triangulations of the n-gon. Unpublished manuscript (2003). An abstract was included in: Topological and Geometric Combinatorics, April 6th–April 12th, 2003, Mathematisches Forschungsinstitut Oberwolfach, Report No. 16/2003, http://www.mfo.de/programme/schedule/2003/15/Report16_2003.pdf

6 Jakob Jonsson and Wolkmar Welker, A spherical initial ideal for Pfaffians. *Illinois Journal of Mathematics* **51**:4, (2007), 1397–1407.

7 Gil Kalai, Hyperconnectivity of Graphs. *Graphs Comb.*, **1** (1985), 65–79.

8 László Lovász, M. D. Plummer, *Matching Theory.* Ann. Discr. Math., Vol. 29. North-Holland Mathematics Studies, Vol. 121. Amsterdam 1986.

9 Diane Maclagan and Bernd Sturmfels, *Introduction to tropical geometry.* Graduate Studies in Mathematics, vol. 161, American Mathematical Society, Providence, RI, 2015.

10 Ezra Miller, Bernd Sturmfels, *Combinatorial commutative algebra.* Graduate Texts in Mathematics. Vol. 227. New York, NY, 2005. ISBN 0-387-23707-0

11 Lior Pachter, Bernd Sturmfels. *Algebraic Statistics for Computational Biology*, Cambridge University Press, 2005. https://doi.org/10.1017/CBO9780511610684.

12 Günter Rote. Division-Free Algorithms for the Determinant and the Pfaffian: Algebraic and Combinatorial Approaches. In *Computational Discrete Mathematics. Advanced Lectures*, Lect. Notes in Comp. Sci. 2122, Springer, 2001. pp. 119–135.

13 David Speyer, Lauren Williams. The Tropical Totally Positive Grassmannian. *J. Algebr. Comb.* *22* (2005), 189–210. https://doi.org/10.1007/s10801-005-2513-3

Self-affine tilings, multivariate B-splines and subdivision schemes

Tatyana Zaitseva
Lomonosov Moscow State University, Russia,
http://approx-lab.math.msu.su/staff/Zaitseva_eng.html

Abstract

We consider two classification problems of self-affine tiles: the classification of polyhedral tiles and of two-digit tiles. This enables us to develop a theory of multivariate tile B-splines based on which we construct subdivision schemes for generating surfaces and establish their remarkable properties.

2012 ACM Subject Classification Mathematics of computing → Approximation; Computing methodologies → Mesh geometry models; Theory of computation → Numeric approximation algorithms

Keywords and phrases self-affine tilings, reptiles, B-splines, subdivision algorithms, wavelets

Related Version https://arxiv.org/abs/2007.11279, https://arxiv.org/abs/2107.00518

1 Introduction

Self-affine tiles are widely studied in the literature due to numerous applications in approximation theory, in construction of wavelets, crystallography, computer graphics [1, 2, 3, 4, 5].

Generally, a **tiling** of a set $G \subset \mathbb{R}^n$ is its partition of the form $G = \cup_{i=1}^{m} S_i(T)$, where T is a compact set called **tile**, and S_i are transforms from a certain class. The tiling is called **self-similar** if the tile T is similar to G by means of orthogonal transformations and homothety, and, more generally, **self-affine** if the tile is similar to G by means of some affine operator. The set T is called a self-affine tile in this case.

A popular class of self-similar tilings are reptiles. In this case, the transforms S_i consist of translations and rotations, see, for example, fig. 1 (left). Reptiles have lots of application in different fields including algorithms and data structures which use their replicating nature (such as quadtrees), modeling of the cell division, etc. Most of fractals defined as iterated function systems (IFS) are also reptiles.

2 Fundamental results

We restrict our analysis to translational self-affine tilings, i.e., all S_i are translations. Such tilings were studied in works on geometry, combinatorics, number theory, topology, numerical

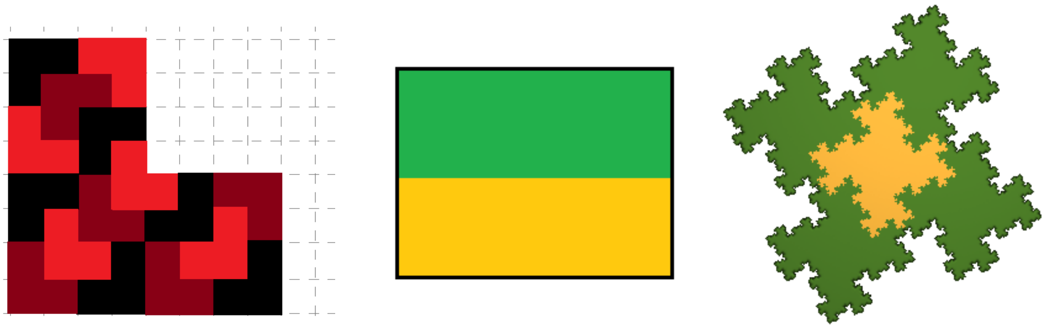


Figure 1 Left: a reptile. Right: translational tilings.

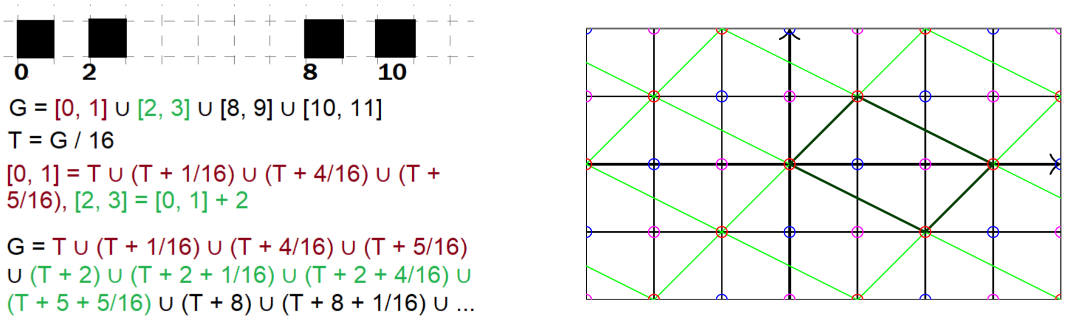

0
2
8
10
G = [0, 1] ∪ [2, 3] ∪ [8, 9] ∪ [10, 11]
T = G / 16
[0, 1] = T ∪ (T + 1/16) ∪ (T + 4/16) ∪ (T + 5/16), [2, 3] = [0, 1] + 2
G = T ∪ (T + 1/16) ∪ (T + 4/16) ∪ (T + 5/16) ∪ (T + 2) ∪ (T + 2 + 1/16) ∪ (T + 2 + 4/16) ∪ (T + 5 + 5/16) ∪ (T + 8) ∪ (T + 8 + 1/16) ∪ ...

Figure 2 Left: 1d disconnected self-similar tiling. Right: The example of $M\mathbb{Z}^2$.

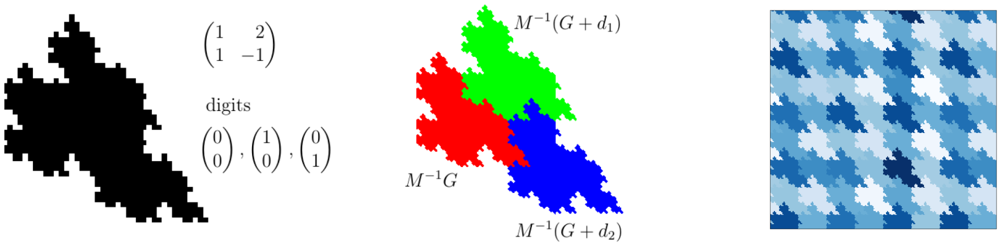

$\begin{pmatrix}1 & 2\\ 1 & -1\end{pmatrix}$
digits
$\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}$
$M^{-1}(G+d_1)$
$M^{-1}G$
$M^{-1}(G+d_2)$

Figure 3 The example of set G, its partition to three parts, and tiling of the plane.

analysis, see [1, 6, 7, 8, 9, 10, 11, 12, 13, 14] and references therein. Translational tilings can also be very different. A simple example of a self-affine tile is a parallelepiped that is the union of its binary contractions. However, this case is rather exceptional, and a typical self-affine tile has a complicated structure and fractal-like properties (see the examples on fig. 1, right). A natural question arises if there exist simple tiles other than parallelepipeds.

▶ **Problem 1.** *Classify all self-affine tiles that are polyhedral (polyhedra or finite unions of polyhedra) up to an affine similarity.*

The convex case is rather simple: only parallelepiped admits a self-affine tiling. The general case (non-convex, disconnected sets) was studied in recent works [14, 15]. The complete classification had been known only for the one-dimensional case [16], which is based on [17], see also [15]. Even in this case the problem is highly non-trivial, see fig. 2 (left).

In [18] (a brief report, the full work is currently unpublished), we obtained a complete classification for general polyhedral tilings in arbitrary dimension. Connected tiles can only be parallelepipeds, but the disconnected case produces infinitely many non-equivalent polyhedral tiles. They are unions of translates of unit cubes with a special structure. For reptiles, the problem 1 is still open; even for simple polyhedra such as simplices there are many open questions [19, 5, 20].

There is a particular case of our self-affine translational tilings which is a key ingredient in construction of Haar wavelet systems used in signal processing [1, 2, 3]. Let $M \subset \mathbb{Z}^{n\times n}$ be an integer expanding matrix ($|\lambda_i| > 1$ for all eigenvalues). The matrix M maps the lattice $\mathbb{Z}^n$ to the lattice of parallelepipeds $M\mathbb{Z}^n$ (see fig. 2 right). The volume of each parallelepiped is $m = |\det M|$, and all integer points are colored in m colours with respect to this lattice. We choose one point d_i of each colour i and form the set of digits $D \subset \mathbb{Z}^n = \{d_0, \dots d_{m-1}\}$.

The compact set $G = G(M, D) = \left\{\sum_{k=1}^{\infty} M^{-k} d_{a_k} : a_k \in \{0, 1, \dots, m-1\}\right\}$ satisfies the self-similarity equation: $G = M^{-1}(G+d_0) \cup \dots \cup M^{-1}(G + d_{m-1})$ (see fig. 3). Thus, it admits a self-affine translational tiling with a self-affine tile $T = M^{-1}G$ and $S_i(T) = T + M^{-1}d_i$.

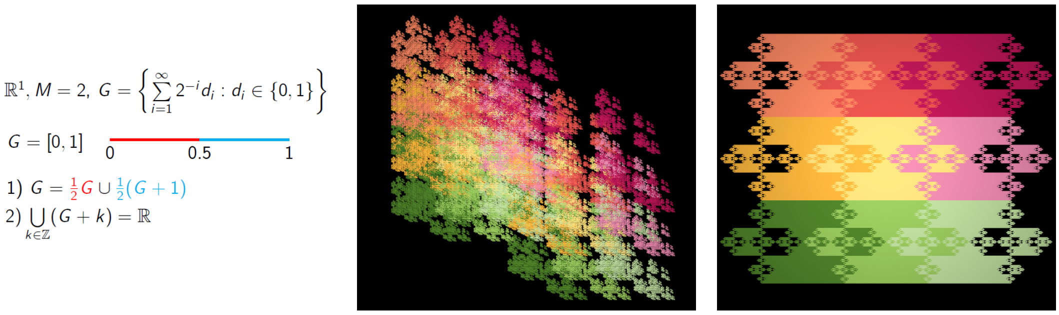
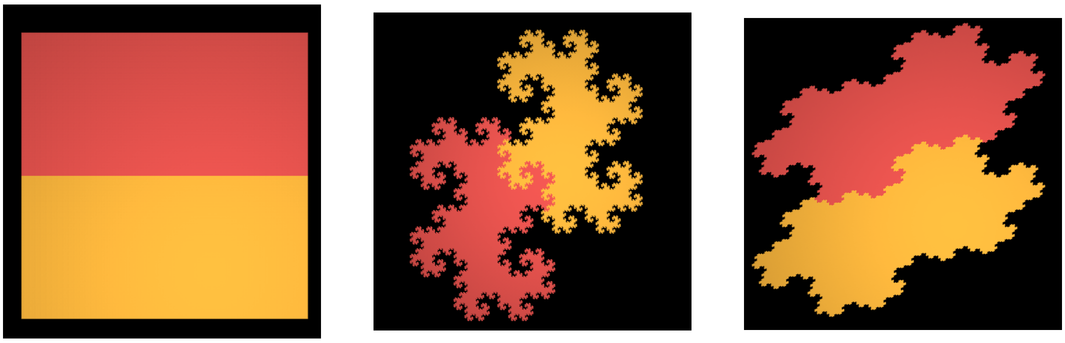

$\mathbb{R}^1$, $M = 2$, $G = \left\{\sum_{i=1}^{\infty} 2^{-i} d_i : d_i \in \{0,1\}\right\}$
$G = [0,1]$
0
0.5
1
1) $G = \frac{1}{2}G \cup \frac{1}{2}(G+1)$
2) $\bigcup_{k\in\mathbb{Z}} (G+k) = \mathbb{R}$

Figure 4 Example of different self-affine tilings.

The set G can also be considered as a self-affine tile of the set MG: $MG = (G + d_0) \cup \ldots \cup (G + d_{m-1})$, $S_i(G) = G + d_i$. In [1, 9] it is proved that the Lebesgue measure $|G|$ is a positive integer and the translates $\{G + k\}_{k\in\mathbb{Z}^n}$ cover the entire space $\mathbb{R}^n$ in $|G|$ layers. If $|G| = 1$, then $G + k_1$ and $G + k_2$ with $k_1 \neq k_2$ have intersection only of measure zero. The translations of set G are said to form a tiling of $\mathbb{R}^n$ (fig. 3) that explains another meaning why the set G is called a tile. The simplest example in $\mathbb{R}^1$ is a unit segment $[0, 1]$ for $M = 2$, $D = \{0, 1\}$ (see fig. 4 left). On the plane, tilings can have different properties (see fig. 4). The two-digital case, i.e., when $m = 2$, is the most convenient in applications. We call such sets G 2-tiles.

▶ **Problem 2.** *Classify all 2-tiles up to affine similarity.*

It is known that in the two-dimensional case there are three 2-tiles, we call them a square, a dragon, and a bear (in the literature they are also known as rectangle, twindragon and tame twindragon), see fig. 5. 2-tiles were studied, for example, in [6, 7, 9, 11, 12, 21].



Square

Dragon

Bear

Figure 5 The partitions of the plane 2-tiles into two affinely-similar parts

Using algebraic tools, we obtained [21] the full classification of 2-tiles when the matrix is isotropic (similar to an orthogonal matrix multiplied by a number). It turns out that in odd dimensions they can be only parallelepipeds, and in every even dimension $n = 2k$, there exist precisely three 2-tiles: the parallelepiped, the direct product of k (two-dimensional) dragons, and the direct product of k (two-dimensional) bears. We reduced the non-isotropic case to the problems on polynomials. In 3D, there are 7 non-similar types and only one of them is isotropic (parallelepiped) [6]. In 4D there are 21 types, etc [21].

Surprisingly, on the plane there are six types of 2-reptiles (with two parts) with rational angles of rotations [22]. Three of them coincide with 2-tiles and three are another dragons.

3 Tile B-splines and their applications to surface modelling

Tiles allow to generalize the classical concept of B-splines to the multivariate setting. Namely, we consider the tile B-spline $B_n(G) = \underbrace{\chi_G * \ldots * \chi_G}_{n+1}$, where G is a tile, $\chi_G(x) = 1$ if $x \in G$, $\chi_G(x) = 0$ if $x \notin G$, and $f * g(x) = \int f(y)g(x-y)dy$ is a convolution. In the univariate case $G = [0, 1]$ (see fig. 6, left, see [23]).

Surprisingly enough, the B_3 with G chosen as a Bear tile is three times continuously differentiable while the classical B-spline of the same order is not (see B_3 on fig. 6, right).

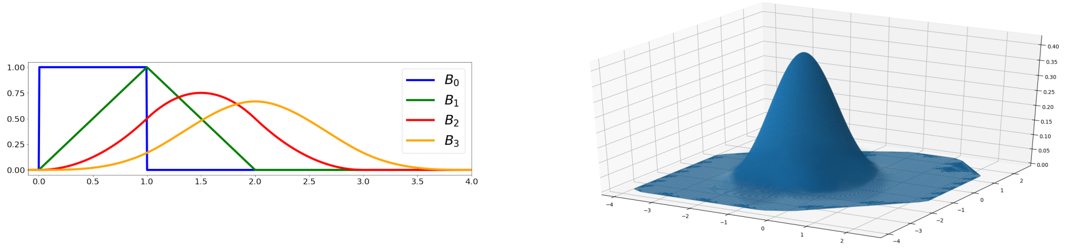


Figure 6 Left: 1d B-splines. Right: Tile B-spline B_3 for Bear tile.

Tile B-splines generate efficient subdivision schemes which are linear iterative algorithms extrapolating functions from their values on a regular grid (fig. 7). From our results it follows that the B_3-subdivision scheme produces C^3 surfaces, which improves many well-known schemes such as four-point scheme, butterfly scheme, etc. I currently work on this topic investigating the behavior of B_k for $k > 3$ and how these subdivision schemes could be generalized for irregular meshes.

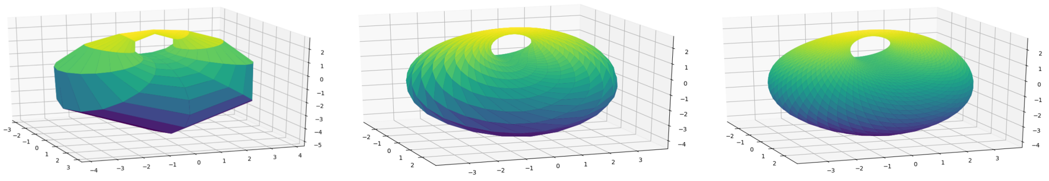


Figure 7 After 0, 2 and 4 iterations of the subdivision scheme B_3.

References

1 J. Lagarias, Y. Wang, Integral self-affine tiles in $\mathbb{R}^n$. II. Lattice tilings, J. Fourier Anal. Appl. **3**(1):83-102, 1997.

2 N. Dyn, D. Levin, Subdivision schemes in geometric modelling. Acta Numerica, **11**(0):73-144, 2002.

3 C.A. Cabrelli, C. Heil, U.M. Molter, Self-similarity and multiwavelets in higher dimensions, Memoirs Amer. Math. Soc. **170**(807), 2004.

4 P. Wojtaszczyk, sA Mathematical Introduction to Wavelets, London Math. Soc. Stud. Texts, vol. 37, Cambridge Univ. Press, Cambridge, New York, Melbourne, Madrid (1997).

5 G. Gelbrich, Crystallographic reptiles. Geom. Dedic. **51**(3):235–256, 1994.

6 C. Bandt, Combinatorial topology of three-dimensional self-affine tiles, (2010) arXiv:1002.0710

7 W.J. Gilbert, Radix representations of quadratic fields, J. Math. Anal. Appl. **83**(1):264-274, 1981.

8 K. Gröchenig, A. Haas, Self-similar lattice tilings, J. Fourier Anal. Appl., **1**(2):131-170, 1994.

9 C. Gröchenig, W.R. Madych, Multiresolution analysis, Haar bases, and self-similar tilings of $\mathbb{R}^n$, IEEE Trans. Inform. Theory **38**(2):556-568, 1992.

10 P. Kirschenhofer, J.M. Thuswaldner, Shift radix systems – a survey, In: Numeration and Substitution 2012, Research Institute for Mathematical Sciences (RIMS), Kyoto, 1-59, 2014.

11 X.G. He, K.S. Lau, Characterization of tile digit sets with prime determinants, Applied and Computational Harmonic Analysis, **16**(3):159-173, 2004.

12 R. Gundy, A. Jonsson, Scaling functions on $\mathbb{R}^2$ for dilations of determinant ±2, Applied and Computational Harmonic Analysis **29**(1):49-62, 2010.

13 J.M. Thuswaldner, S.-Q. Zhang, On self-affine tiles whose boundary is a sphere, Trans. Amer. Math. Soc. **373**(1): 491-527, 2020.

14 Y.-M. Yang and Y. Zhang. Tilings of convex polyhedral cones and topological properties of self-affine tiles. Discrete & Computational Geometry, **66**:876–901, 2020.

15 T. Zaitseva. Simple tiles and attractors. Sb. Math. **211**(9):24–59, 2020.

16 C. Long. Addition theorems for sets of integers. Pacific Journal of Mathematics, **23**(1): 107–112, 1967.

17 N. G. de Bruijn. On number systems. Nieuw Arch. Wisk., **4**: 15–17, 1956.

18 V. Protasov, T. Zaitseva. Self-affine tilings of polyhedra. Doklady Mathematics, **500**:55–61, 2021, arxiv: 2107.00518.

19 J. Matoušek, Z. Safernová, On the nonexistence of k-reptile tetrahedra. Discrete & Computational Geometry, **46**(3):599-609, 2011.

20 J. Kynčl, Z. Patakova, On the nonexistence of k reptile simplices in $\mathbb{R}^3$ and $\mathbb{R}^4$. The Electronic Journal of Combinatorics, **24**(3), 2017.

21 V. Protasov, T. Zaitseva. Self-affine 2-attractors and tiles. arXiv: 2007.11279.

22 S.-M. Ngai, V.F. Sirvent, J.J.P. Veerman, Y. Wang, On 2-Reptiles in the Plane, Geom. Dedicata **82**(1):325-344, 2000.

23 T. Zaitseva, https://github.com/TZZZZ/Tile_Bsplines.

On Asymptotic Packing of Geometric Graphs

Daniel W. Cranston
Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA

Jiaxi Nie
Department of Mathematics, UCSD, San Diego, CA, USA

Jacques Verstraëte
Department of Mathematics, UCSD, San Diego, CA, USA

Alexandra Wesolek
Department of Mathematics, Simon Fraser University, Burnaby, BC Canada

Abstract

A set of geometric graphs is *geometric-packable* if it can be asymptotically packed into every sequence of geometric drawings $(H_n)_{n\geq 1}$ of complete graphs $(K_n)_{n\geq 1}$. When G is a triangle, 4-cycle or 4-cycle with a chord, we show that the set of plane drawings of G is geometric-packable. In contrast, the analogous statement is false when G is nearly any other planar Hamiltonian graph (with at most 3 possible exceptions). Further, for each planar Hamiltonian graph G, we determine whether or not plane drawings of G can be asymptotically packed into the sequence of *convex* drawings of complete graphs.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory

Keywords and phrases Geometric graph drawings, convex drawings, graph packing.

Related Version A full version of the paper is available at `https://arxiv.org/abs/2111.03933`.

Funding *Jacques Verstraëte*: Research supported by NSF award DMS-1800332.
Alexandra Wesolek: Supported by the Vanier Canada Graduate Scholarships program.

1 Introduction

A *geometric graph* G is a graph drawn in the Euclidean plane such that its vertices are points in general position (no three points on the same line) and its edges are drawn as straight line segments. In this paper, H_n is a geometric complete graph on n vertices. Generally, a packing of H_n is a collection of geometric subgraphs of H_n which are edge-disjoint. An important conjecture asks whether it is possible to pack $\lfloor \frac{n}{2} \rfloor$ plane spanning trees into H_n. This is possible if the drawing of H_n is convex [1, 3], which means the vertices are in convex position, but it was very recently shown that the general conjecture is false [5]. The current best general construction shows that it is possible to pack $\lfloor \frac{n}{3} \rfloor$ plane spanning trees into H_n [2]. In this paper, we relax the requirement that the graphs that are packed be large, and we focus on packing small graphs into H_n.

For any abstract planar graph G, let $\mathcal{P}(G)$ denote the set of all plane geometric drawings of G. A $\mathcal{P}(G)$*-packing* of H_n is a collection of edge-disjoint subgraphs of H_n that are each a member of $\mathcal{P}(G)$. Figure 1 shows examples of two distinct geometric graphs H_6 and $\mathcal{P}(C_4)$-packings, where C_4 is the 4-cycle. The size of a $\mathcal{P}(G)$-packing is the number of subgraphs of H_n that are matched with a member in $\mathcal{P}(G)$. Let $p(H_n, \mathcal{P}(G))$ be the maximum size of

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

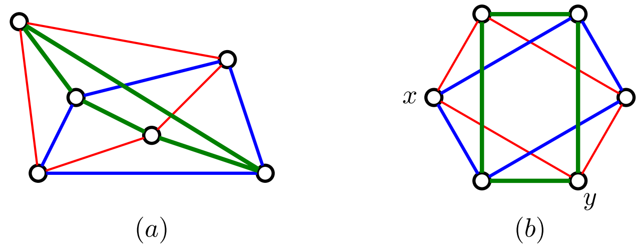


Figure 1 Packing of plane 4-cycles into geometric graphs H_6. The drawing (b) is convex.

a $\mathcal{P}(G)$-packing in H_n and $e(G)$ the number of edges of G. Let $\{H_n\}_{n\geq 1}$ be a sequence of complete geometric graphs. Now $\{H_n\}_{n\geq 1}$ can be *asymptotically packed by* $\mathcal{P}(G)$ if

$$\lim_{n\to\infty} \frac{p(H_n, \mathcal{P}(G))e(G)}{e(H_n)} = 1.$$

Further, we say G is *geometric-packable* if any sequence $\{H_n\}_{n\geq 1}$ can be asymptotically packed by $\mathcal{P}(G)$.

2 Main results

▶ **Theorem 1.** *If G is a planar Hamiltonian graph, then $\mathcal{P}(G)$ is not geometric-packable unless G is the 3-cycle C_3, the 4-cycle C_4, or one of the four graphs $\Theta_1, \Theta_2, \Theta_3, \Theta_4$ shown in Figure 2. Further, $\mathcal{P}(G)$ is geometric-packable if G is one of C_3, C_4, and Θ_1.*

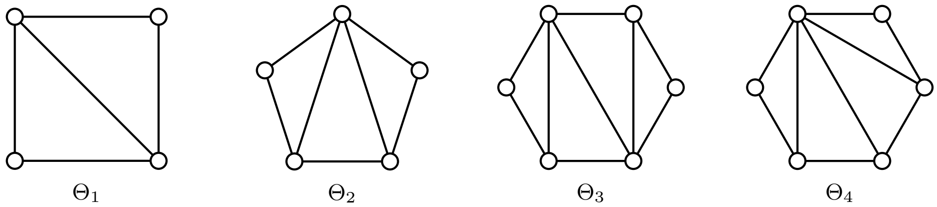


Figure 2 Four plane triangulated cycles. The first, Θ_1, is geometric-packable. For each of the remaining three, the question of geometric-packability remains open.

In Section 3 we provide a construction which shows that Θ_1 is geometric-packable. To show that a graph is not geometric-packable, we show that it does not asymptotically pack into some sequence $\{H_n\}_{n\geq 1}$. The sequence we consider is the sequence of convex H_i's.

A *convex geometric graph* (CGG for short) G is a geometric graph whose vertices are in strictly convex position as in Figure 1 (b). We denote the vertices of G by $v_0, v_1, \ldots, v_{n-1}$ and assume these vertices appear in clockwise (cyclic) order on the boundary of their convex hull (indexing is modulo n). Informally, two CGGs are convex-isomorphic if some graph isomorphism between them preserves the cyclic order of all vertices. Let $\mathcal{K}_n$ be the complete CGG on n vertices. Given a CGG G, a *G-packing* of $\mathcal{K}_n$ is a collection of edge-disjoint subgraphs of $\mathcal{K}_n$ that are convex-isomorphic to G. For example, Figure 1 shows that there are 3 edge disjoint copies of C_4 in $\mathcal{K}_6$ (which is best possible since $\lfloor\binom{6}{2}/4\rfloor = 3$.)

G is *convex-packable* if $\{\mathcal{K}_n\}_{n\geq 1}$ can be asymptotically packed by G. Note that the plane C_3 and C_4 are convex-packable by Theorem 1; $\{\mathcal{K}_n\}_{n\geq 1}$ can be asymptotically packed by $\mathcal{P}(C_3), \mathcal{P}(C_4)$ and each plane subgraph C_i in K_n is the unique CGG of C_i. We naturally ask:

Are all plane cycles convex-packable? The answer is No and it follows from an edge length argument. The length of an edge in $\mathcal{K}_n$ is the distance between its vertices with respect to the boundary cycle (of the convex hull). For example, the length of the edge xy in Figure 1 is two. In fact, for all $k \geq 5$, the average length of the edges in a plane copy of C_k in $\mathcal{K}_n$ is at most n/k. Hence, the average length of all edges covered by a plane C_k-packing of $\mathcal{K}_n$ is also at most n/k. In contrast, the average length of all edges in $\mathcal{K}_n$ is $(1 + o(1))n/4$. So when $k \geq 5$, no plane C_k-packing can cover all but $o(n^2)$ edges of $\mathcal{K}_n$. By extending this average length argument, we find a necessary condition for a CGG to be convex-packable [4, Lemma 5].

▶ **Theorem 2.** *All plane Hamiltonian CGGs are not convex-packable, except for the two plane cycles C_3 and C_4 and the four CGGs $\Theta_1, \Theta_2, \Theta_3$ and Θ_4 shown in Figure 2, which are all convex-packable.*

The advantage of studying Hamiltonian graphs is that they have a unique embedding as plane CGG's (consecutive vertices along the Hamiltonian cycle are consecutive in the clockwise order of the CGG). Hence, if a graph has a unique embedding as a plane CGG, convex-packability follows directly from geometric-packability. While in this paper we consider plane CGG's G, it can be further asked which CGG's that have a crossing are convex-packable.

3 Proof that Θ_1 is geometric-packable.

We consider the set of plane 4-cycles with a chord, $\mathcal{P}(\Theta_1)$. Let D_n be an arbitrary geometric drawing in the plane of K_n. Let $f(n) = 2n \log_2 n$. We prove by induction on n that there exists a Θ_1-packing of D_n that covers all but at most $f(n)$ edges.

Let $m = \lfloor n/4 \rfloor$. By the Ham Sandwich Theorem, there exist two straight lines partitioning the plane into 4 parts, where each part contains at least m vertices. Ignoring up to 3 vertices, we pick m vertices in each part and denote the resulting vertex sets in clockwise order by P_1, P_2, P_3, P_4 and the vertices by $v_{i,j}$, where $i \in \{1, \ldots, 4\}$ is the vertex set and $j \in \{1, \ldots, m\}$.

Let D'_n be the spanning subgraph of D_n whose edge set consists of all edges with endpoints in distinct parts, except for those with one endpoint in each of P_2 and P_4. Let F_n be a collection of copies of plane Θ_1 whose vertex set is $\{v_{1,j}, v_{2,k}, v_{3,j+k}, v_{4,k}\}$ and whose chord is $\{v_{1,j}, v_{3,j+k}\}$, with $j, k \in \{1, \ldots, m\}$; here each second index is modulo m. Figure 3 shows an example for $n = 8$ (and $m = 2$). It is easy to check that F_n is a Θ_1-packing of D'_n that covers all but at most $3n$ edges. Note that $D_n \setminus D'_n$ consists of three complete components, one induced by $P_2 \cup P_4$, and the others induced by P_1 and P_3. Thus, by

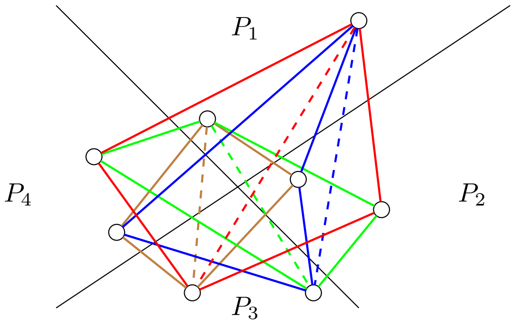


Figure 3 The vertices are partitioned into four equal parts P_1, P_2, P_3, P_4. The figure shows the set F_8 of edge disjoint Θ_1's, where chords are drawn as dashed lines.

induction, there exists a Θ_1-packing of D_n such that the number of uncovered edges is at most $f(n/2) + 2f(n/4) + 3n \leq f(n)$, as desired.

References

1 Frank Bernhart and Paul C Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979.

2 Ahmad Biniaz and Alfredo García. Packing plane spanning trees into a point set. *Computational Geometry*, 90:101653, 2020.

3 Prosenjit Bose, Ferran Hurtado, Eduardo Rivera-Campo, and David R Wood. Partitions of complete geometric graphs into plane trees. *Computational Geometry*, 34(2):116–125, 2006.

4 Daniel W Cranston, Jiaxi Nie, Jacques Verstraëte, and Alexandra Wesolek. On asymptotic packing of geometric graphs. *arXiv preprint arXiv:2111.03933*, 2021.

5 Johannes Obenaus and Joachim Orthaber. Edge partitions of complete geometric graphs (part 1). *arXiv preprint arXiv:2108.05159*, 2021.

A Lattice-Theoretic Perspective on the Persistence Map*

Brendan Mallery[1]
Tufts University, Medford, Massachusetts, USA

Adélie Garin
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Justin Curry
University at Albany, State University of New York, USA

Abstract

We provide a naturally isomorphic description of the persistence map from merge trees to barcodes [3] in terms of a monotone map from the partition lattice to the subset lattice. Our description is local, which offers the potential to speed up inverse computations, and brings classical tools in combinatorics to bear on an active area of research in topological data analysis (TDA).

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Theory of computation → Computational geometry; Mathematics of computing → Combinatoric problems; Mathematics of computing → Permutations and combinations; Mathematics of computing → Trees

Keywords and phrases inverse problems, lattices, persistent homology, merge trees, barcodes, persistence map

Funding
Adélie Garin: SNSF, CRSII5 177237
Justin Curry: Supported by NSF CCF-1850052 and NASA 80GRC020C0016

1 Background on the Inverse Problem

Merge trees play a central role in topological data analysis (TDA). One can apply persistent homology to a merge tree to obtain an "adjacency free" description of a merge tree in terms of its barcode, we call this association of a barcode to a merge tree the *persistence map*. Characterizing precisely how many merge trees map to the same barcode was studied in [2,3,5] and has yielded significant connections to geometric group theory, combinatorics, and statistics. Understanding the fiber of the persistence map is crucial for understanding how noise in data propagates to noise in persistent homology.

In [3,5] a combinatorial version of this inverse problem was considered; see Figure 1. A *combinatorial merge tree* is a binary, rooted, combinatorial tree with birth-ordered labels on the leaves $\{0, 1, \ldots, n\}$ and death-ordered labels on the internal nodes. Every barcode with n finite-length bars whose left (birth) endpoints are distinct and whose right (death) endpoints are distinct can be encoded by a *combinatorial barcode* $B = \{(i, j)\}$ if the i^{th} birth endpoint is matched with the j^{th} death endpoint. Equivalently, a combinatorial barcode is the graph of a permutation σ of $\{1, \ldots, n\}$.

In this abstract, we characterize the persistence map from combinatorial merge trees to combinatorial barcodes in terms of monotone maps between two lattices: the subset lattice and the partition lattice. We show that a maximal chain in the subset and partition lattices

[*] This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

[1] Corresponding Author

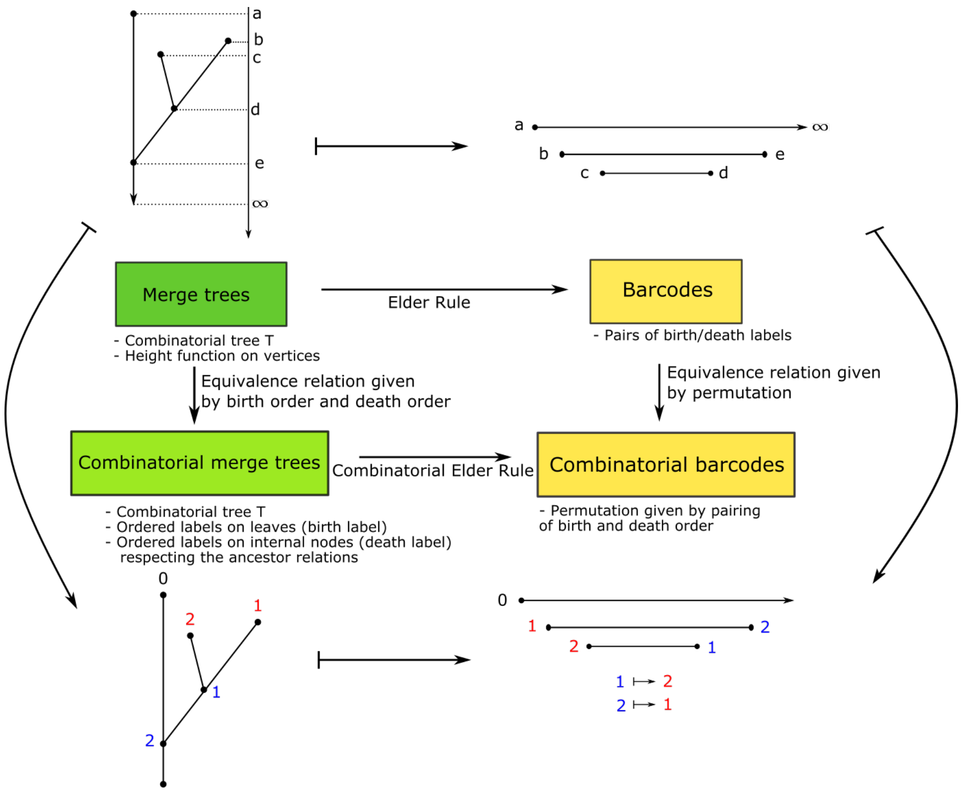

a
b
c
d
e
∞
a
∞
b
e
c
d
Merge trees
Elder Rule
Barcodes
- Combinatorial tree T
- Height function on vertices
- Pairs of birth/death labels
Equivalence relation given by birth order and death order
Equivalence relation given by permutation
Combinatorial merge trees
Combinatorial Elder Rule
Combinatorial barcodes
- Combinatorial tree T
- Ordered labels on leaves (birth label)
- Ordered labels on internal nodes (death label) respecting the ancestor relations
- Permutation given by pairing of birth and death order
0
2
1
1
2
0
1
2
2
1
1 ↦ 2
2 ↦ 1

Figure 1 Figure from [3], expressing the combinatorial inverse problem.

corresponds to a combinatorial barcode and combinatorial merge tree respectively, and that one may incrementally construct solutions to the inverse problem using this correspondence.

2 A Lattice Version of the Inverse Problem

Let $(P, \preccurlyeq)$ be a poset. Recall that a *lattice* is a poset equipped with meets and joins. A totally ordered subset $\mathcal{C} \subseteq P$ is called a *chain*. A chain is *maximal* if it is not a proper subset of any other chain in P. A *path* γ is a chain C such that for any $e \in P$ that lies between two elements of C, then $C \cup \{e\}$ is not a chain. A path is *based* at $x_0 \in P$ if the lowest element in γ is x_0. If P has a unique lowest element $\hat{0}$ (e.g. a lattice), we write $\tilde{P}$ as the poset of paths based at $\hat{0}$, which is a poset via containment of paths. There is a unique surjective map $\pi_P : \tilde{P} \to P$ sending a path to its endpoint. Furthermore, if $f : P \to Q$ is a monotone map of posets, there is a unique map $\tilde{f} : \tilde{P} \to \tilde{Q}$ such that $f \circ \pi_P = \pi_Q \circ \tilde{f}$. We call $\tilde{f}$ the *lift* of f.

▶ **Definition 1** (Subset Lattice). *Let $[n] = \{1, \dots, n\}$ and consider $P = \mathcal{P}([n])$, the set of all subsets of $[n]$, including the empty set $\emptyset$, equipped with the partial order $\subseteq$ of "being a subset of". This forms the **subset lattice** Π_n of $[n]$, with $A \cap B$ and $A \cup B$ being the meet and join of $A, B \in \Pi_n$, respectively. The poset of paths in Π_n based at $\emptyset$ is $\tilde{\Pi}_n$.*

▶ **Definition 2** (Partition Lattice). *A* partition *of the set $\mathbf{n} := \{0, 1, \dots, n\}$ is a collection of disjoint subsets $\mathcal{U} = \{U_1, \dots, U_k\}$ of $\mathbf{n}$ whose union is $\mathbf{n}$. A partition $\mathcal{U}$* refines *a partition $\mathcal{U}'$, written $\mathcal{U} \preceq \mathcal{U}'$, if every subset of $\mathcal{U}'$ is equal to a union of elements of $\mathcal{U}$. We denote the **lattice of partitions** of $\mathbf{n}$ by $\mathcal{P}_n$. The poset of paths based at $\{\{0\}, \dots, \{n\}\}$ is $\tilde{\mathcal{P}}_n$.*

We can filter a combinatorial barcode B with n bars into sets $B_1 \subset \cdots \subset B_n := B$ where B_k is the set of pairs $\{(i,j)\}_{j \leq k}$. We refer to B_k as a *partial (combinatorial) barcode.* The set of all partial barcodes with at most n bars forms a poset by containment, which we denote by $\mathcal{PCB}_n$. Similarly, a *partial (combinatorial) merge tree* is a filtration of a combinatorial merge tree T with $n+1$ leaves by subgraphs $T_0 \subset T_1 \subset \cdots \subset T_n := T$ where T_k is the full subgraph supported on the set of leaf nodes and all internal nodes with label less than or equal to k. Partial merge trees also forms a poset by subgraph containment, denoted $\mathcal{PCT}_n$; see Figure 2. The persistence map between combinatorial merge trees and barcodes extends to a map from $\mathcal{PCT}_n$ to $\mathcal{PCB}_n$, which we also call the persistence map.

▶ **Theorem 3.** *The poset of partial merge trees $\mathcal{PCT}_n$ and barcodes $\mathcal{PCB}_n$ are isomorphic to $\tilde{\mathcal{P}}_n$ and $\tilde{\Pi}_n$, respectively. Furthermore, there is a monotone map $H : \mathcal{P}_n \to \Pi_n$ whose lift $\tilde{H} : \tilde{\mathcal{P}}_n \to \tilde{\Pi}_n$ is naturally isomorphic to the persistence map from $\mathcal{PCT}_n \to \mathcal{PCB}_n$.*

Proof. Every partial merge tree $T_0 \subset \cdots \subset T_k$ defines a path $\mathcal{U}_0 < \cdots < \mathcal{U}_k$, where $\mathcal{U}_i$ is the partition of the leaf node labels induced by connected components in the graph T_i. Each partition of leaf labels can be identified with a partition of $\mathbf{n}$. Hence we obtain an equivalence between partial merge trees and paths of partitions $\mathcal{U}_0 < \cdots < \mathcal{U}_k$ which give a sequence of connected components indexed by the filtration parameter, see the bottom left graphic in Figure 2. The maximal element (endpoint) of a path $\gamma \in \tilde{\mathcal{P}}_n$ of length k corresponds to a partition with k parts, indexing the leaf labels of the connected components of the k'th level set of a patial merge tree T_k.

Similarly, every partial barcode $B_1 \subset \cdots \subset B_k$ defines a path in the subset lattice $\emptyset := A_0 \subset \cdots \subset A_k$. Each set A_k is the set of birth labels whose deaths occur by time k. These specify the isomorphisms.

Define $H : \mathcal{P}_n \to \Pi_n$ as follows: Let $(U_1, U_2, ..., U_k)$ be a partition of $\mathbf{n}$. For each U_i, let $U_i' := U_i \setminus \{\min\{x \in U_i\}\}$. Let $H((U_1, U_2, ..., U_k)) = \cup_{i \in [k]} U_i' \in \Pi_n$. This map is monotone, since if $(U_1, U_2, ..., U_k) \leq (V_1, V_2, ..., V_l)$, then the latter partition is obtained by collapsing parts of the first, which can only add elements to $H((U_1, U_2, ..., U_k))$. It is easy to see that this map is also surjective. This lifts to a natural map $\tilde{H}$, defined on paths.

The map H maps each partition to the subset of $\mathbf{n}$ obtained by removing the eldest leaf node in each component from $\mathbf{n}$. This is in accordance with the Elder Rule [2] of persistent homology, which asserts that when the rank of the persistence module decreases by e.g. one, the youngest bar has been killed by the persistence algorithm. Hence we see that H indicates which bars in the filtration persist upon a "merge event"; the image is the union $B = \cup_{i \in [k]} B_i$ of leaf node labels that have been killed by stage k. Applying this computation to each element in γ produces a path in $B_0 \subset B_1 ... \subset B_k$ in $\mathcal{B}_n$. The combinatorial barcode is encoded by the successive differences between B_i and B_{i+1}.

◀

3 Future Work

Theorem 3 is still in need of a full geometric description that accounts for actual positions and lengths of bars in a barcode and edges in a merge tree. In [1] a novel coordinatization of barcode space was given based on the relation with the symmetric group. However, a similar picture for merge tree space that uses the connection with the partition lattice is unknown. Additionally, the lattice structure on these "skeletonizations" of barcode and merge tree space has not been fully explored. As noted in [4, 6, 7], Möbius inversion provides another way of summarizing topological changes in a filtration, which suggests that inverse problems, lattice theory, and Möbius inversion may occupy a rich intersection of ideas.

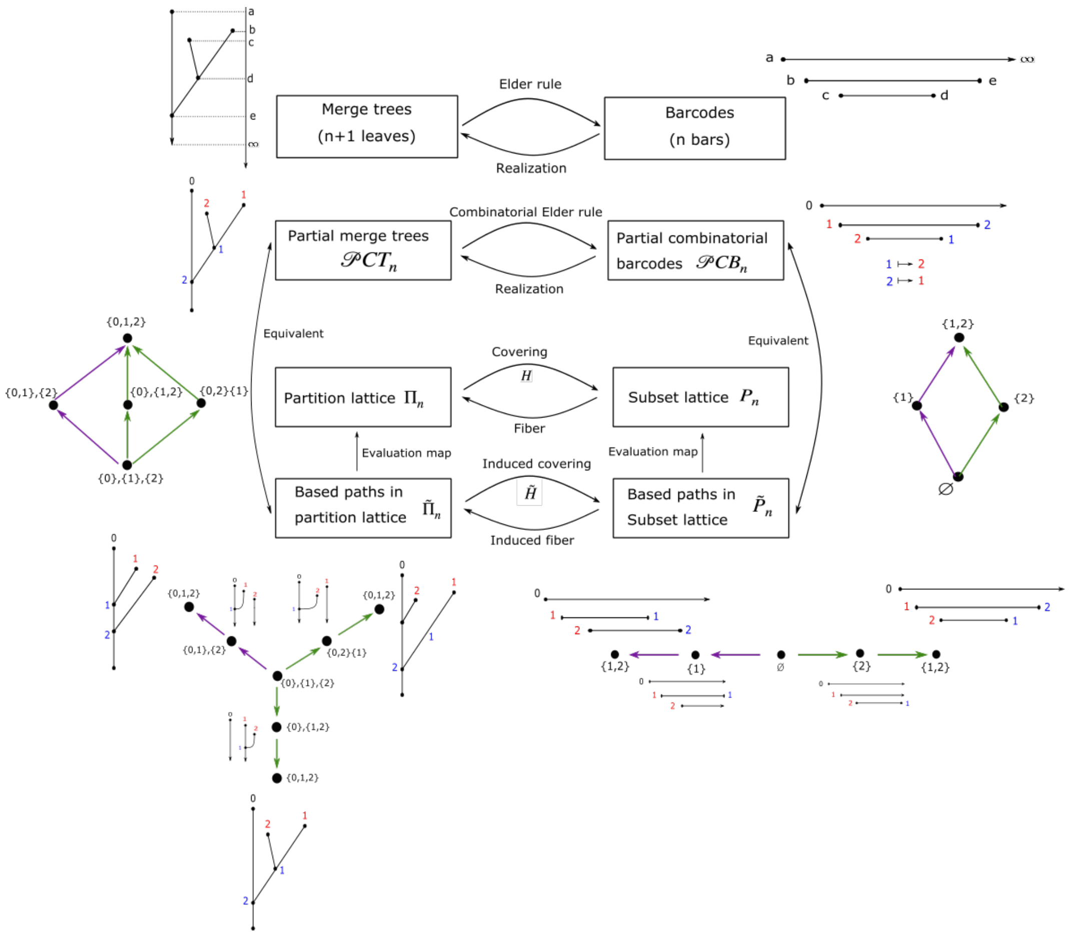


Figure 2 Illustration of Theorem 3.

References

1 Benjamin Brück and Adélie Garin. Stratifying the space of barcodes using Coxeter complexes. *ArXiv*, 2112.10571, 2021.

2 Justin Curry. The fiber of the persistence map for functions on the interval. *Journal of Applied and Computational Topology*, 2(3):301–321, 2018.

3 Justin Curry, Jordan DeSha, Adélie Garin, Kathryn Hess, Lida Kanari, and Brendan Mallery. From trees to barcodes and back again II: Combinatorial and probabilistic aspects of a topological inverse problem. *ArXiv*, 2107.11212v2, 2021.

4 Aziz Burak Gulen and Alexander McCleary. Diagrams of persistence modules over finite posets. *arXiv preprint arXiv:2201.06650*, 2022.

5 Lida Kanari, Adélie Garin, and Kathryn Hess. From trees to barcodes and back again: theoretical and statistical perspectives. *Algorithms*, 13, 2020. doi:10.3390/a13120335.

6 Alexander McCleary and Amit Patel. Edit distance and persistence diagrams over lattices. *arXiv preprint arXiv:2010.07337*, 2020.

7 Amit Patel. Generalized persistence diagrams. *Journal of Applied and Computational Topology*, 1(3):397–419, 2018.

Persistent sheaf cohomology

Florian Russold

Institute of Geometry, Graz University of Technology, Austria

Abstract

We expand the toolbox of (co)homological methods in computational topology by applying the concept of persistence to sheaf cohomology. Since sheaves (of vector spaces) combine topological information with algebraic information, they allow for variation along an algebraic dimension and along a topological dimension. Consequently, we introduce two different constructions of sheaf cohomology (co)persistence modules. One of them can be viewed as a natural generalization of the construction of simplicial or singular cohomology copersistence modules. We discuss how both constructions relate to each other and show that some classical results from persistence theory can be generalized to sheaves.

2012 ACM Subject Classification Mathematical Foundations

Keywords and phrases Computational topology, Persistent homology, Sheaf theory

Related Version A full version of the paper is available at `https://arxiv.org/abs/2204.13446`.

Funding *Florian Russold*: The author is supported by the Austrian Science Fund (FWF): W1230.

Acknowledgements The author thanks Michael Kerber, Barbara Giunti and Jan Jendrysiak for useful discussions.

1 Introduction

In recent years, applied sheaf theory started to gain momentum in the computational topology community. Sheaves [1, 11] are used, to describe information flows in networks [9, 10, 12], for sensor integration and data fusion [14, 15], for stratification learning [2] and in various other situations [4, 8, 13]. One of the main tools of sheaf theory and, in particular, applied sheaf theory is sheaf cohomology. Sheaf cohomology can be used to investigate local to global inference problems. For example, sheaf cohomology (or cosheaf homology) is used to compute global persistent (co)homology from local persistent (co)homology [4, 5, 17, 19]. The goal of this work is to apply the concept of persistence [6, 7, 18], one of the most prominent tools of computational topology, to sheaf cohomology. Instead of tracking the evolution of simplicial homology under the variation of simplicial complexes, we track the evolution of sheaf cohomology under the variation of sheaves. Since sheaves (of vector spaces) combine topological and algebraic information, they allow for variation along the algebraic and the topological dimension. Consequently, we introduce the following two constructions of sheaf cohomology (co)persistence modules.

2 Sheaf persistence modules of algebraic type

For our first construction, we use the well-known fact [11] that (k-dimensional) sheaf cohomology assigns to a sheaf of $\mathbb{F}$-vector spaces F on a topological space X a vector space $H^k(X, F)$. Moreover, this assignment is functorial, that is, given a morphism of sheaves $\phi\colon F \to G$, we obtain a linear map of the cohomology vector spaces $H^k(X, \phi)\colon H^k(X, F) \to$

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

$H^k(X,G)$. Hence, as depicted in Figure 1, a linear diagram $\vec{F}$ of sheaves and sheaf morphisms gives rise to a one-dimensional persistence module $H^k(X,\vec{F})$. Note that the algebraic

$$\begin{tikzcd}
\vec{F}\colon & F_0 \arrow[r, "\phi_0"] & F_1 \arrow[r, "\phi_1"] & F_2 \arrow[r, "\phi_2"] & \cdots \\
H^k(X,\vec{F})\colon & H^k(X,F_0) \arrow[r, "H^k(X{,}\phi_0)"] & H^k(X,F_1) \arrow[r, "H^k(X{,}\phi_1)"] & H^k(X,F_2) \arrow[r, "H^k(X{,}\phi_2)"] & \cdots
\end{tikzcd}$$

Figure 1 Construction of a sheaf persistence module of algebraic type.

information provided by the sheaves F_i is variable along $\vec{F}$, whereas the topological space X is fixed. Therefore, we call the obtained persistence module a *sheaf persistence module of algebraic type*. If $H^k(X,F_i)$ is finite-dimensional for all $i \in \mathbb{N}_0$, then $H^k(X,\vec{F})$ can be decomposed into interval modules corresponding to persistent sheaf cohomology classes in $\vec{F}$.

A classical result from persistence theory states that persistence modules (of finite type) correspond to (finitely generated) graded $\mathbb{F}[t]$-modules [3, 18]. For example, $H^k(X,\vec{F})$ corresponds to the graded $\mathbb{F}[t]$-module $\bigoplus_{n\in\mathbb{N}_0} H^k(X,F_n)$ where $t \cdot x := H^k(X,\phi_n)(x)$ for all $x \in H^k(X,F_n)$. In analogy to persistence modules of finite type, we define a diagram $\vec{F}$ to be of finite type if it becomes "constant" at some point and F_i is a sheaf of finite-dimensional vector spaces for all $i \in \mathbb{N}_0$. The following theorem extends the results above to linear diagrams of sheaves and contains them as the special case of a one-point space.

▶ **Theorem 1.** *A linear diagram of sheaves $\vec{F}$ (of finite type) on X corresponds to a sheaf $M_{\vec{F}}$ of (finitely generated) graded $\mathbb{F}[t]$-modules on X.*

The next theorem states that we can relate the persistent cohomology of $\vec{F}$ with the ordinary cohomology of the corresponding sheaf of graded modules $M_{\vec{F}}$.

▶ **Theorem 2.** *There is an isomorphism of graded $\mathbb{F}[t]$-modules*

$$H^k(X,M_{\vec{F}}) \cong \bigoplus_{n\in\mathbb{N}_0} H^k(X,F_n) \quad .$$

The significance of this result is that it allows us to compute the interval decomposition of $H^k(X,\vec{F})$ by computing the cohomology of the corresponding sheaf of graded modules. Under certain conditions, the cohomology of $M_{\vec{F}}$ can be computed by (graded) matrix reduction, a method familiar from standard persistent homology.

3 Sheaf copersistence modules of topological type

For our second construction we use the following: Given a continuous map $f\colon X \to Y$ and a sheaf of vector spaces F on Y, we can pull back the sheaf F on Y along the map f to obtain a sheaf f^*F on X called the inverse image sheaf with respect to f. Moreover, there exists a linear map $H^k(f)\colon H^k(Y,F) \to H^k(X,f^*F)$ induced by f on cohomology vector spaces [11]. We now consider a linear diagram of topological spaces and continuous maps $\vec{X}$ indexed by $\mathbb{N}_0 \cup \{\infty\}$ as depicted by the commutative diagram in the upper part of Figure 2 and a sheaf F on X_∞. We define, for every $i \in \mathbb{N}_0$, the sheaf $F^i := g_i^*F$ on X_i. Note that $F^i = g_i^*F = (g_{i+1} \circ f_i)^*F = f_i^*g_{i+1}^*F = f_i^*F^{i+1}$. Hence, as depicted in the middle row of Figure 2, we can visualize the construction as iteratively pulling back the sheaves F^i along the maps f_{i-1}. We now connect the sheaf cohomology vector spaces $H^k(X_i,F^i)$ by the linear maps $H^k(f_i)\colon H^k(X_{i+1},F^{i+1}) \to H^k(X_i,f_i^*F^{i+1}) = H^k(X_i,F^i)$ induced by the continuous

$$\begin{tikzcd}
 & & X_\infty & & \\
\vec{X}\colon & X_0 \arrow[ur, "g_0"] \arrow[r, "f_0"] & X_1 \arrow[u, "g_1"] \arrow[r, "f_1"] & X_2 \arrow[ul, "g_2"] \arrow[r, "f_2"] & \cdots \arrow[ull] \\
 & F^0 & F^1 \arrow[l, mapsto, "f_0^*"'] & F^2 \arrow[l, mapsto, "f_1^*"'] & \cdots \arrow[l, mapsto, "f_2^*"'] \\
D^k(\vec{X},F)\colon & H^k(X_0,F^0) & H^k(X_1,F^1) \arrow[l, "H^k(f_0)"'] & H^k(X_2,F^2) \arrow[l, "H^k(f_1)"'] & \cdots \arrow[l, "H^k(f_2)"']
\end{tikzcd}$$

Figure 2 Construction of a sheaf copersistence module of topological type.

maps f_i to obtain the copersistence module $D^k(\vec{X},F)$ depicted in the bottom row of Figure 2. In this case, the algebraic information is provided by a fixed sheaf F, whereas the topological information is variable along the diagram $\vec{X}$. Therefore, we call this copersistence module a *sheaf copersistence module of topological type*. If $H^k(X_i,F^i)$ is finite-dimensional for all $i \in \mathbb{N}_0$, then $D^k(\vec{X},F)$ can be decomposed into interval modules corresponding to persistent sheaf cohomology classes of the sheaves F^i over $\vec{X}$.

If the involved topological spaces are semi-locally contractible [16] or abstract simplicial complexes equipped with the Alexandrov topology [4] and if F is the constant $\mathbb{F}$-valued sheaf, then this construction yields the singular or simplicial cohomology copersistence module, respectively. Hence, our second construction can be viewed as a generalization of the construction of singular or simplicial cohomology copersistence modules.

Under some conditions on the maps f_i and g_i, we can associate to the pair $(\vec{X},F)$ a linear diagram $G(\vec{X},F)$ of sheaves and sheaf morphisms on X_∞ such that there is an isomorphism of copersistence modules $H^k\big(X_\infty, G(\vec{X},F)\big) \cong D^k(\vec{X},F)$. Hence, in some cases, we can reduce the second construction to the first one. This is possible, for example, if $\vec{X}$ is a filtration of abstract simplicial complexes equipped with the Alexandrov topology.

It is also possible to consider a diagram of topological spaces $\vec{X}$ and a diagram of sheaves $\vec{F}$ on X_∞ and combine both constructions to obtain a two-dimensional (co)persistence module with a topological and an algebraic dimension.

References

1 Glen E. Bredon. *Sheaf Theory*. Graduate Texts in Mathematics. Springer New York, 1997. URL: https://books.google.at/books?id=zGdqWepiT1QC.

2 Adam Brown and Bei Wang. Sheaf-theoretic stratification learning from geometric and topological perspectives. *Discrete and Computational Geometry*, 65, 2021. doi:10.1007/s00454-020-00206-y.

3 René Corbet and Michael Kerber. The representation theorem of persistence revisited and generalized. *Journal of Applied and Computational Topology*, 2(1-2):1–31, 2018. URL: http://dx.doi.org/10.1007/s41468-018-0015-3, doi:10.1007/s41468-018-0015-3.

4 Justin Curry. *Sheaves, Cosheaves and Applications*. PhD thesis, University of Pennsylvania, 2014. arXiv:1303.3255.

5 Justin Curry, Robert Ghrist, and Vidit Nanda. Discrete morse theory for computing cellular sheaf cohomology. *Foundations of Computational Mathematics*, 16, 2013. doi:10.1007/s10208-015-9266-8.

6 Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete and Computational Geometry*, 45:737–759, 2009. doi:10.1007/s00454-011-9344-x.

7 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533, 2002. doi:10.1007/s00454-002-2885-2.

8 Robert Ghrist. *Elementary Applied Topology.* CreateSpace Independent Publishing Platform, 2014. URL: https://books.google.at/books?id=Z5ATogEACAAJ.

9 Robert Ghrist and Yasuaki Hiraoka. Applications of sheaf cohomology and exact sequences on network codings. 2011. https://www2.math.upenn.edu/ ghrist/preprints/networkcoding-short.pdf.

10 Jakob Hansen and Robert Ghrist. Opinion dynamics on discourse sheaves. *SIAM Journal on Applied Mathematics*, 81:2033–2060, 2021. doi:10.1137/20M1341088.

11 Birger Iversen. *Cohomology of Sheaves.* Universitext. Springer Berlin Heidelberg, 1986. URL: https://books.google.at/books?id=kKfDQgAACAAJ.

12 Michael Robinson. Understanding networks and their behaviors using sheaf theory. *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*, 2013. doi:10.1109/GlobalSIP.2013.6737040.

13 Michael Robinson. *Topological Signal Processing.* Mathematical Engineering. Springer Berlin Heidelberg, 2014. URL: https://books.google.at/books?id=ieiGAgAAQBAJ.

14 Michael Robinson. Sheaves are the canonical data structure for sensor integration. *Information Fusion*, 36:208–224, 2017. URL: https://www.sciencedirect.com/science/article/pii/S156625351630207X, doi:https://doi.org/10.1016/j.inffus.2016.12.002.

15 Michael Robinson, C. Joslyn, Emilie Hogan, and C. Capraro. Conglomeration of heterogeneous content using local topology. 2015. https://www.drmichaelrobinson.net/simplex_techrep.pdf.

16 Yehonatan Sella. Comparison of sheaf cohomology and singular cohomology, 2016. arXiv:1602.06674. arXiv:1602.06674.

17 Hee Rhang Yoon and Robert Ghrist. Persistence by parts: Multiscale feature detection via distributed persistent homology, 2020. arXiv:2001.01623. arXiv:2001.01623.

18 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33:249–274, 2005. doi:10.1007/s00454-004-1146-y.

19 Álvaro Torras Casas. Distributing persistent homology via spectral sequences, 2020. arXiv:1907.05228. arXiv:1907.05228.

Stratifying the space of barcodes using Coxeter complexes*

Benjamin Brück
ETH Zürich, Switzerland

Adélie Garin[1]
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Abstract

We use tools from geometric group theory to produce a stratification of the space $\mathcal{B}_n$ of barcodes with n bars. The top-dimensional strata are indexed by permutations associated to barcodes as defined by Kanari, Garin and Hess. More generally, the strata correspond to marked double cosets of parabolic subgroups of the symmetric group Sym_n. This subdivides $\mathcal{B}_n$ into regions that consist of barcodes with the same averages and standard deviations of birth and death times and the same permutation type.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Combinatoric problems; Mathematics of computing → Permutations and combinations

Keywords and phrases Barcodes, Coxeter complex, stratification

Related Version A full version of the paper is available at `https://arxiv.org/abs/2112.10571`.

Funding *Benjamin Brück*: ETHZ, FIM
Adélie Garin: SNSF, CRSII5 177237

1 Introduction

Barcodes [7] are topological summaries of the persistent homology of a filtered space. The barcode B associated to a *filtration* $\{X_t\}_{t\in\mathbb{R}}$ is a multiset of points $(b, d) \in \mathbb{R}^2$. It summarises the creation and destruction of homology classes while varying the parameter t, which is often interpreted as "time". A bar $(b, d) \in B$ corresponds to a homology cycle appearing in X_b and becoming a boundary in X_d. The first element of the pair (b, d) is called the *birth* and the second one the *death*. A barcode is called strict if the births and deaths all take different values [8, 6]. A strict barcode B can be associated with a permutation $\sigma_B \in \mathrm{Sym}_n$ that tracks the order of the deaths with respect to the order of the births [8, 6].

Understanding the space of barcodes is crucial to be able to do statistics or comparison between data using their barcodes. We use Coxeter complexes to develop a new description of the set $\mathcal{B}_n$ of barcodes with n bars. We give coordinates for this set that have natural interpretations when doing statistics with barcodes. These coordinates define a stratification of $\mathcal{B}_n$ where the top-dimensional strata are indexed by the symmetric group Sym_n.

The advantages of these new coordinates are two-fold: Firstly, using points in Coxeter complexes, one obtains coordinates that uniquely specify barcodes and are yet compatible with the combinatorial structure of $\mathcal{B}_n$ given by permutation equivalence classes in [8, 6]. Secondly, one resolves the problem that permutation equivalence classes themselves carry no notion of "size": The decomposition of $\mathcal{B}_n$ into regions subdivides these equivalence classes

[*] This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

[1] Corresponding Author

by also taking into account the averages and standard deviations of births and deaths. This makes these regions a finer invariant than the permutation type used in [8, 6] to compare neurons' barcodes and random barcodes. This work lays the theoretical basics to extend these applications.

2 Background

The *Coxeter complex* $\Sigma(\mathrm{Sym}_n)$ [1, 3, 2] associated with Sym_n consists of a simplicial complex where the top-dimensional simplices are in one-to-one correspondence with the elements of Sym_n. Its geometric realization is a simplicial decomposition of the $(n-2)$-sphere, see Figure 1, and is the dual of the permutohedron [9]. The set of k-simplices in $\Sigma(\mathrm{Sym}_n)$ is in one-to-one correspondence with the cosets of rank-$(n-1-k)$ parabolic subgroups P_T of Sym_n, i.e., subgroups that are generated by sets of k adjacent transpositions of the type $(i, i+1)$. More formally, let S be the set of all adjacent transpositions, and for $T \subset S$, let P_T be the subgroup generated by T. Then

$$\Sigma(\mathrm{Sym}_n) = \bigcup_{T \subseteq S} \mathrm{Sym}_n / P_T = \{\tau P_T \mid \tau \in \mathrm{Sym}_n, T \subseteq S\},$$

where each simplex τP_T has dimension $\dim(\tau P_T) = |S \setminus T| - 1$ and the face relation is defined by the partial order $\tau P_T \leq \tau' P_{T'} \Leftrightarrow \tau P_T \supseteq \tau' P_{T'}$. The group Sym_n acts simplicially on $\Sigma(\mathrm{Sym}_n)$ by left multiplication on the cosets, $\gamma \cdot (\tau P) = \gamma\tau P$, and the action on the geometric realisation of $\Sigma(\mathrm{Sym}_n)$ corresponds to hyperplanes reflections in $\mathbb{R}^n$ (see Figure 1).

A stratification [4] is a decomposition of a space into disjoint subsets (strata) with nice properties. For this abstract, it is enough to know that a simplicial decomposition of a sphere gives a stratification, where the strata are given by the simplices.

3 Contributions

Our main contributions can be summarised as follows (see [5] for more details).

▶ **Theorem 1.** *Let $\mathcal{B}_n$ denote the set of barcodes with n bars.*

1. *$\mathcal{B}_n$ can in a natural way be seen as a subset of a quotient $\mathrm{Sym}_n \backslash \mathbb{R}^{2n}$.*
2. *$\mathcal{B}_n$ is stratified over a certain poset of double cosets of parabolic subgroups of Sym_n, see [5] for a full statement.*
3. *Using this description, one obtains a decomposition of $\mathcal{B}_n$ into different regions. Each region is characterised as the set of all barcodes having the same average birth and death, the same standard deviation of births and deaths and the same permutation type $\sigma_B \in \mathrm{Sym}_n$, as defined in [8, 6].*
4. *This description gives rise to metrics on $\mathcal{B}_n$ that coincide with modified versions of the bottleneck and Wasserstein metrics, see [5].*

To obtain this description of $\mathcal{B}_n$ we proceed as follows. A barcode is an (unordered) multiset of n pairs of real numbers (births and deaths). It can hence be seen as a point in the quotient space $\mathrm{Sym}_n \backslash (\mathbb{R}^n \times \mathbb{R}^n)$, where the action of Sym_n permutes the coordinate pairs. Since the birth is smaller than the death for every barcode, $\mathcal{B}_n$ is a proper subset of this quotient of $\mathbb{R}^{2n}$.

The Coxeter complex $\Sigma(\mathrm{Sym}_n)$ associated to Sym_n is a simplicial complex whose geometric realisation is homeomorphic to an $(n-2)$-sphere (Figure 1). Hence, we can decompose $\mathbb{R}^n$ as $\mathbb{R}^n \cong \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R}$, where $\mathrm{cone}(\Sigma(\mathrm{Sym}_n)) = \big(\Sigma(\mathrm{Sym}_n) \times [0, \infty)\big)/(x, 0) \sim (y, 0) \cong$

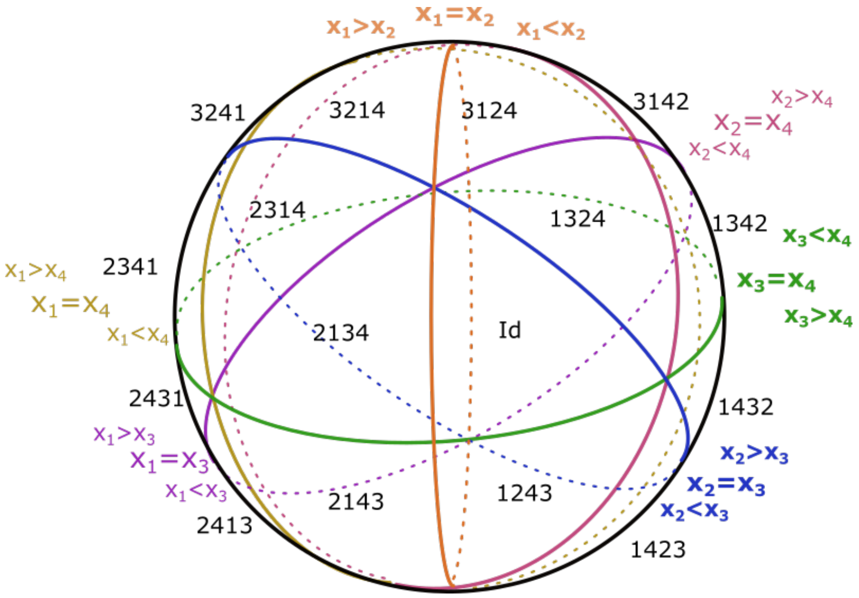


Figure 1 The geometric realisation of the Coxeter complex $\Sigma(\mathrm{Sym}_4)$. The permutation corresponding to each triangle of the front of the sphere is indicated in black. The hyperplanes $x_i = x_j$ depicted in colours correspond to the transpositions $(i, j) \in \mathrm{Sym}_4$.

$\mathbb{R}^{n-1}$. This decomposition allows to describe each point $x \in \mathbb{R}^n$ via coordinates $x_\theta, \bar{x}, \|v_x\|$, where x_θ specifies a point on the Coxeter complex, $\|v_x\|$ is the "cone parameter" and $\bar{x}$ parametrises the remaining $\mathbb{R}$.

In summary, this describes $\mathcal{B}_n$ as a subset of

$$\mathcal{B}_n \subset \mathrm{Sym}_n \backslash \big(\mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R} \times \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R}\big).$$

We call the coordinates that we obtain from this description *Coxeter coordinates*. It turns out that for each barcode, these coordinates are $b_\theta, \bar{b}, \|v_b\|$ and $d_\theta, \bar{d}, \|v_d\|$, where $\bar{b}$ and $\bar{d}$ are the averages of the births and deaths, $\|v_b\|$ and $\|v_d\|$ are their standard deviations and the coordinates b_θ and d_θ describe the permutation equivalence class of the barcode of [8, 6]. The stratification one obtains is induced by the simplicial structure of $\Sigma(\mathrm{Sym}_n)$.

References

1 Peter Abramenko and Kenneth S. Brown. *Buildings*, volume 248 of *Graduate Texts in Mathematics*. Springer, New York, 2008. doi:10.1007/978-0-387-78835-7.

2 Anders Björner. Some combinatorial and algebraic properties of Coxeter complexes and Tits buildings. *Advances in Mathematics*, 52(3):173–212, 1984. doi:10.1016/0001-8708(84)90021-5.

3 Anders Björner and Francesco Brenti. *Combinatorics of Coxeter groups*, volume 231 of *Graduate Texts in Mathematics*. Springer, New York, 2005.

4 Martin R. Bridson and André Haefliger. *Metric spaces of non-positive curvature*, volume 319 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1999. doi:10.1007/978-3-662-12494-9.

5 Benjamin Brück and Adélie Garin. Stratifying the space of barcodes using Coxeter complexes. *ArXiv*, 2112.10571, 2021.

6 Justin Curry, Jordan DeSha, Adélie Garin, Kathryn Hess, Lida Kanari, and Brendan Mallery. From trees to barcodes and back again: theoretical and statistical perspectives, 2021. arXiv:https://arxiv.org/abs/2010.11620.

7 Herbert Edelsbrunner and John Harer. Persistent homology—a survey. In *Surveys on discrete and computational geometry*, volume 453 of *Contemp. Math.*, pages 257–282. Amer. Math. Soc., Providence, RI, 2008. doi:10.1090/conm/453/08802.

8 Lida Kanari, Adélie Garin, and Kathryn Hess. From trees to barcodes and back again: theoretical and statistical perspectives. *Algorithms*, 13, 2020.

9 Alexander Postnikov. Permutohedra, associahedra, and beyond. *Int. Math. Res. Not. IMRN*, (6):1026–1106, 2009. doi:10.1093/imrn/rnn153.

Improved Search of Relevant Points for Nearest-Neighbor Classification

Alejandro Flores-Velazco
Department of Computer Science
University of Maryland, College Park, MD, USA

Abstract

Given a training set $P \subset \mathbb{R}^d$, the *nearest-neighbor classifier* assigns any query point $q \in \mathbb{R}^d$ to the class of its closest point in P. The set of *border points* of P are those that define the boundaries that separate points of different classes, and thus are relevant to correctly classify new query points. Improving over a decades-long result by Clarkson (FOCS'94), a recent paper by Eppstein (SOSA'22) proposes an output-sensitive algorithm to find the set of border points of P in $\mathcal{O}(n^2 + nk^2)$ time, where k is the size of such set. In this paper, we further improve this algorithm to run in $\mathcal{O}(nk^2)$ time by proving that the first steps of the original algorithm, which require $\mathcal{O}(n^2)$ time, are unnecessary.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases nearest-neighbor classification, nearest-neighbor rule, decision boundaries, border points, relevant points

Related Version A full version of the paper is available at `https://arxiv.org/abs/2203.03567`.

1 Introduction

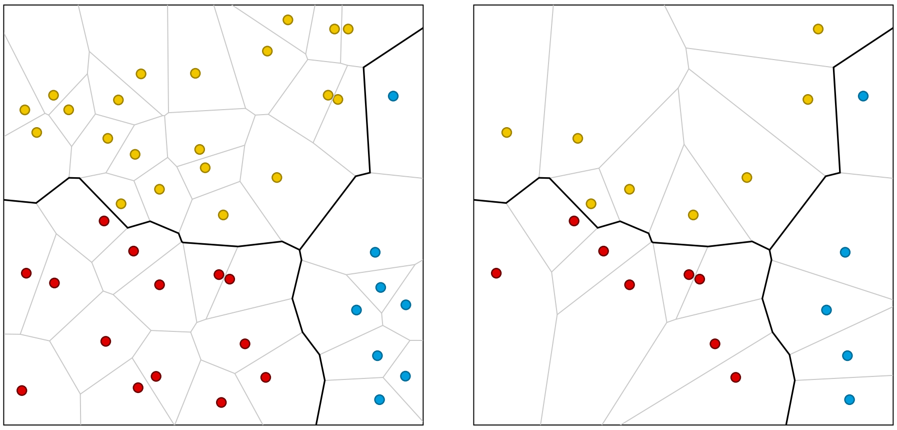
In the context of non-parametric classification, we are given a training set $P \subset \mathbb{R}^d$ consisting of n *labeled* points in d-dimensional Euclidean space, where the label of every point in P indicates the *class* (or *color*) that the point belongs to. The *nearest-neighbor classifier* [3] is a well-known classification technique that *predicts* the class of any *unlabeled* query point $q \in \mathbb{R}^d$ with the class of its closest point in P. That is, it *classifies* q.

The set of *border points* (or *relevant points*[1]) of the training set P are those that define the boundaries between points of different classes, and whose omission from the training set would imply the misclassification of some query points in $\mathbb{R}^d$. Formally, two points $p, \hat{p} \in P$ are border points of P if they belong to different classes, and there exist some point $q \in \mathbb{R}^d$ such that q is equidistant to both p and $\hat{p}$, and no other point of P is closer to q than these two points (*i.e.*, the empty ball property of Voronoi Diagrams). See Figure 1 for an example of a training set P in $\mathbb{R}^2$ and its set of border points. Throughout, we let k denote the total number of border points in the training set. By definition, if instead of building the nearest-neighbor classifier with the entire training set P we use the set of border points of P, its dependency is reduced from n to k, while still obtaining the same classification for any query point in $\mathbb{R}^d$. This becomes particularly relevant for applications where $k \ll n$.

Improving over a decades-long result by Clarkson [1], a recent paper by Eppstein [2] proposes an *output-sensitive* algorithm to find the set of border points of P in $\mathcal{O}(n^2 + nk^2)$ worst-case time, where k is the size of such set. In this paper, we further improve Eppstein's algorithm to have time complexity equal to $\mathcal{O}(nk^2)$. We achieve this by proving that the first steps of the original algorithm, which require $\mathcal{O}(n^2)$ time, are unnecessary.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

[1] While [2] uses the term *relevant points*, the term *border points* has been the standard in the literature of this and other related problems [4, 6–8]. For this reason, we stick to the term *border points*.



(a) Original training set P **(b)** Border/Relevant points of P

Figure 1 On the left, a training set P with points of three classes: *red*, *blue* and *yellow*. There the black lines highlight the *boundaries* of P between points of different classes. On the right, a subset of these points corresponding to the set of border points of P. Note that by definition, the boundaries between points of different classes remain the same for P and for its set of border points.

2 Sketching Our Approach

Eppstein's algorithm. Eppstein's approach is strikingly simple, yet full of interesting ideas. The algorithm can be naturally split into two main phases or steps. The initialization step, where it selects an initial set of border points, which is then used during the search step in order to find all the remaining border points of P.

The initialization step involves finding an initial subset of all border points. In particular, it finds at least one point for every class boundary of P. Eppstein observes that this can be achieved by computing the Minimum Spanning Tree (MST) of P, identifying the edges of the MST that connect points of different classes (denoted as *bichromatic* edges), and selecting the endpoints of all such edges. This step takes $\mathcal{O}(n^2)$ time, but we prove it is unnecessary.

The search step then finds every remaining border point of P. It iterates over every selected border point p, and uses a series of subroutines that we jointly call the "inversion method". Basically, this method identifies a subset of border points that are "visible" from p. After running this method on every selected point, the algorithm terminates with the guarantee of having selected every border point of P.

Our Approach. We propose a simple modification to Eppstein's algorithm, which avoids the initialization step altogether. That is, avoiding to compute the MST of P, along with the subsequent selection of bichromatic edges to produce the initial subset of border points.

Instead, we simply start the search process with any arbitrary point of P, while the rest of the algorithm remains virtually unchanged. We show that this new algorithm is not only correct, meaning that it only finds border points of P, but also complete, as all border points of P are eventually found by our algorithm. Additionally, by avoiding the main bottleneck of the original algorithm, our algorithm computes the same set of points in $\mathcal{O}(nk^2)$ time.

To understand why this can be done, it is useful to explore why Eppstein's algorithm computes the MST of P in the first place. The reasons are twofold, as both the correctness and completeness proofs of Eppstein's paper rely on it. First, note that the original algorithm only applies the inversion method on border points of P, as Eppstein's correctness proof (*i.e.*, their Lemma 6) guarantees that the resulting points of the inversion method are border points, but only when the point p that it was applied to is also a border point. However, we prove that this is the case whether p is a border point or not, generalizing their statement.

Moreover, these initial border points selected from the MST provide Eppstein's algorithm at least one starting point on every class boundary of P from where to start the search of the remaining border points of P. Additionally, Eppstein's completeness proof shows that the search step can "move along" any given boundary and eventually select all its defining border points. Therefore, from the perspective of their completeness proof, it becomes essential to select at least one starting point from every boundary. However, we are able to prove that the search process is far more powerful, and can even "jump" between nearby boundaries.

Altogether, this implies that regardless of where the search process starts, our algorithm will eventually discover every boundary of P and select all their defining border points. Thus, rendering the computation of the MST of P unnecessary.

Details. The contributions of this work have been sketched due to space constraints. See the full paper [5] for further details on our algorithm, and its correctness and completeness proofs.

References

1 Kenneth L Clarkson. More output-sensitive geometric algorithms. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 695–702. IEEE, 1994.

2 David Eppstein. Finding relevant points for nearest-neighbor classification. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 68–78. SIAM, 2022.

3 E. Fix and J. L. Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. *US Air Force School of Aviation Medicine*, Technical Report 4(3):477+, January 1951.

4 Alejandro Flores-Velazco. Social distancing is good for points too! In *Proceedings of the 32st Canadian Conference on Computational Geometry, CCCG 2020, August 5-7, 2020, University of Saskatchewan, Saskatoon, Saskatchewan, Canada*, 2020.

5 Alejandro Flores-Velazco. Improved search of relevant points for nearest-neighbor classification. *arXiv preprint arXiv:2203.03567*, 2022.

6 Alejandro Flores-Velazco and David M. Mount. Guarantees on nearest-neighbor condensation heuristics. In *Proceedings of the 31st Canadian Conference on Computational Geometry, CCCG 2019, August 8-10, 2019, University of Alberta, Edmonton, Alberta, Canada*, 2019.

7 Norbert Jankowski and Marek Grochowski. Comparison of instances selection algorithms I. Algorithms survey. In *Artificial Intelligence and Soft Computing-ICAISC 2004*, pages 598–603. Springer, 2004.

8 D. Randall Wilson and Tony R. Martinez. Instance pruning techniques. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 403–411, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. URL: `http://dl.acm.org/citation.cfm?id=645526.657143`.

Chromatic k-Nearest Neighbor Queries

Thijs van der Horst
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Maarten Löffler
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Frank Staals
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

We study k-Nearest Neighbor classifiers from a geometric viewpoint. We give data structures with worst-case sublinear query times, even when k is large (i.e., $k = \Theta(n)$).

2012 ACM Subject Classification Theory of computation → Computational Geometry

Keywords and phrases data structure, nearest neighbor, classification

Funding *Maarten Löffler*: Partially supported by the Dutch Research Council (NWO) under the project numbers 614.001.504 and 628.011.005.

Acknowledgements The authors would like to thank anonymous reviewers of an earlier draft of this work for their helpful comments.

1 Introduction

One of the most popular approaches for classification problems is to use a k-Nearest-Neighbor (k-NN) classifier [2, 4, 6, 7]. In a k-NN classifier the predicted class of a query item q is taken to be the most frequently appearing class among the k items most similar to q. One can model this as a geometric problem in which the input items are represented by a set P of n colored points in $\mathbb{R}^d$: the color of the points represents their class, and the distance between points measures their similarity. The goal is then to store P so that one can efficiently find the color (class) c^* most frequently occurring among the k points in P closest to a query point q. See Figure 1(left). We refer to such queries as *chromatic k-NN queries*. To answer such queries, k-NN classifiers often store P in, e.g., a kd-tree and answer queries by explicitly

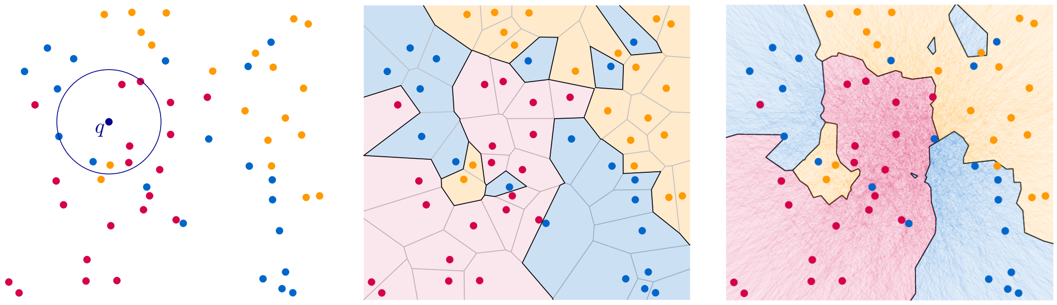


Figure 1 (left) A set of input points from three different classes (colors). The class of a query point q is determined by the labels of its k nearest neighbors (with $k = 7$ as shown here q is classified as red). (center) The color partition for $k = 1$. (right) The color partition for $k = 3$.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

reporting the k points closest to q, scanning through this set to compute the most frequently occurring color [2]. Unfortunately, for many distance measures (including the Euclidean distance) such an approach has no guarantees on the query time other than the trivial $O(n)$ time bound. Even assuming that the dependency on n during the query time is small (e.g. when the points are nicely distributed [5]), the approach requires $\Theta(k)$ time to explicitly process all k points closest to q, whereas the desired output is only a single value: the most frequently appearing color. Hence, our main goal is to design a data structure to store P that has sublinear query time in terms of both n and k, while still using only small space.

The only result on the theory of chromatic k-NN queries that we are aware of is that of Mount *et al.* [8]. They study the problem in the case that we measure distance using the Euclidean metric and that the number of colors c, as well as the parameter k, are small constants. Mount *et al.* state that it is unclear how to obtain a query time independent of k, and instead analyze the query times in terms of the *chromatic density* ρ of a query q. Intuitively, this term models the idea that if many points near q have the same color, queries should be easier to answer than when there are multiple colors with roughly the same number of points. We aim for bounds only in terms of combinatorial properties (i.e. n, c, and k) and allow the number of colors, as well as the parameter k, to depend on n. Our results are particularly relevant when k and c are large compared to n.

2 Our approach

Our main idea is to answer a query in two steps. (1) We identify a region $\mathcal{D}_m^k(q)$ that contains exactly the set k-NN$_m(q)$ of the k sites closest to q according to distance metric L_m. In fact, we search for the smallest disk under metric L_m, that is centered around q, such that k-NN$_m(q)$ is contained in it. This disk uniquely identifies $\mathcal{D}_m^k(q)$. (2) We then find the *mode color* c^*; that is, the most frequently occurring color among the points in the region $\mathcal{D}_m^k(q)$. This way, we never have to explicitly enumerate the set k-NN$_m(q)$. We will design separate data structures for these two steps, for $m \in \{1, 2, \infty\}$, and in two-dimensional space.

2.1 Finding $\mathcal{D}_m^k(q)$

The main idea for finding $\mathcal{D}_m^k(q)$ is to identify an ordered set of candidate distances, such that the distance r^* between q and its k^{th} nearest neighbor is contained in this set. We then perform binary search over these candidate distances, constructing the metric disk $D_m(q,r) = \{p \in \mathbb{R}^2 \mid L_m(p,q) \leq r\}$ for each considered distance r. By then counting the number of points inside this disk, we can find out whether r^* is greater than, smaller than, or exactly equal to r. Once r^* is found, we can report $\mathcal{D}_m^k(q)$ as $D_m(q, r^*)$.

Creating the candidate distances is simple when $m = \infty$ (and by affine transformations on $\mathbb{R}^2$, also when $m = 1$). Because the distance between q and a point $p \in P$ is entirely governed by either the x- or y-coordinates of the points, we can take the set of candidate distances to be $\{|p_x - q_x| \mid p \in P\} \cup \{|p_y - q_y| \mid p \in P\}$. Using balanced binary search trees to store the x- and y-coordinates of P, built during preprocessing, we can search over these candidate distances while only computing a small number ($O(\log n)$) of them explicitly. This approach gives the following result.

▶ **Theorem 1.** *With $O(n \log n)$ preprocessing time and $O(n)$ space, we can find $\mathcal{D}_\infty^k(q)$ in $O(n^\delta)$ time, for an arbitrarily small constant $\delta > 0$.*

When $m = 2$, we use the partition tree $\mathcal{T}$ of Agarwal *et al.* [1], built for circular range searching, to generate candidate distances. Given a set Ω of open, connected regions in $\mathbb{R}^2$, called *cells*, such that there is a cell $\omega \in \Omega$ containing the k^{th} nearest neighbor of q, we let the set of candidate distances be equal to the distances between q and arbitrary points inside the regions in Ω. Using these distances, we can find two distances r^- and r^+, such that $r^- \leq r^* \leq r^+$, and such that at least one of $D(q, r^-)$ and $D(q, r^+)$ crosses ω.

We can then discard all cells in Ω that are not crossed by either of the disks, and construct new candidate distances with the kept cells. Using the bounds of Agarwal *et al.* [1], we get that the total number of candidate distances considered is $O(n^{1/2} \operatorname{polylog} n)$. This gives the following result.

▶ **Theorem 2.** *With $O(n^{1+\delta})$ expected preprocessing time and $O(n)$ space, we can find $\mathcal{D}_2^k(q)$ in $O(n^{1/2} \operatorname{polylog} n)$ time, for an arbitrarily small constant $\delta > 0$.*

2.2 Range mode queries

For range mode queries, we use the results of Chan *et al.* [3]. In the case where $\mathcal{D}_m^k(q)$ is an axis-aligned square (when $m = \infty$), we can use their result for orthogonal range mode queries. However, by moving to three dimensions, we can augment their data structure for halfspace range mode queries, to obtain a faster query time complexity.

We summarize our results for chromatic k-nearest neighbor queries, which are dominated by those of range mode queries, in the following theorem.

▶ **Theorem 3.** *With $O(n^{5/3})$ preprocessing time and $O(n)$ space, we can answer chromatic k-NN queries under the L_∞ metric in $O(n^{2/3+\delta})$ time. With $O(n^{5/3})$ expected preprocessing time, we can answer queries under the L_2 metric in $O(n^{5/6} \operatorname{polylog} n)$ time.*

References

1 Pankaj K. Agarwal, Jirí Matousek, and Micha Sharir. On range searching with semialgebraic sets. II. *SIAM Journal on Computing*, 42(6):2039–2062, 2013. doi:10.1137/120890855.

2 Charu C. Aggarwal. *Data classification: algorithms and applications*. CRC press, 2014.

3 Timothy M. Chan, Stephane Durocher, Kasper Green Larsen, Jason Morrison, and Bryan T. Wilkinson. Linear-space data structures for range mode query in arrays. *Theory of Computing Systems*, 55:719–741, 2014. doi:10.1007/s00224-013-9455-2.

4 Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967. doi:10.1109/TIT.1967.1053964.

5 Jerome H. Friedman, Jon Louis Bentley, and Raphael A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977. doi:10.1145/355744.355745.

6 W. E. Henley and D. J. Hand. A k-nearest-neighbour classifier for assessing consumer credit risk. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 45(1):77–95, 1996. doi:10.2307/2348414.

7 Yan-Nei Law and Carlo Zaniolo. An adaptive nearest neighbor classification algorithm for data streams. In *Knowledge Discovery in Databases: PKDD 2005*, pages 108–120. Springer Berlin Heidelberg, 2005.

8 David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. Chromatic nearest neighbor searching: A query sensitive approach. *Computational Geometry*, 17:97–119, 2000. doi:10.1016/S0925-7721(00)00021-3.

Tractability Frontiers in Multi-Robot Coordination and Geometric Reconfiguration

Tzvika Geft ✉ 🏠
Tel Aviv University

Dan Halperin ✉ 🏠
Tel Aviv University

Yonatan Nakar ✉
Tel Aviv University

Abstract

In multi-robot motion planning (MRMP), the goal is to plan collision-free motion of robots from given start to target positions. Although MRMP has been known to be intractable for decades, it is still not well-understood why that is, which motivates exploring the boundary between tractable and intractable MRMP variants. To this end, we consider a restricted yet still NP-hard version of MRMP, called *monotone* MRMP, in which robots must move one by one to their targets with no intermediate stops. We show that two further simplified variants of monotone MRMP remain NP-hard. These variants serve as tractability frontiers in the sense that they are slight deviations from known tractable problems. All of our reductions are based on an NP-hard job scheduling problem that we refer to as Pivot Scheduling. The connection to scheduling sheds light on the hardness of monotone MRMP.

2012 ACM Subject Classification Computing methodologies → Multi-agent planning; Theory of computation → Computational geometry; Theory of computation → Problems, reductions and completeness

Keywords and phrases Multi-Robot Motion Planning, Geometric Reconfiguration, NP-hardness

Related Version A full version will be available at `https://arxiv.org/abs/2104.07011`.

1 Introduction

In the multi-robot motion planning (MRMP) problem, the aim is to plan the motion of several robots operating in a common workspace, without incurring collisions with obstacles or with fellow robots. MRMP has been shown to be hard in various planar settings [2,8,9,11,12]. Recently, we have shown that *monotone* MRMP, a natural variant in which robots move one by one to their targets, is also NP-hard [7]. This variant is related to geometric reconfiguration [4,6] and robotic object rearrangement, where the problem has been recently shown to be empirically challenging for as little as 10-30 objects [13]. These empirical difficulties prompt the question of which ingredients are underlying the hardness of monotone MRMP, motivating us to improve the theoretical understanding of the problem's intractability.

Our first contribution is introducing Pivot Scheduling as the base problem for our reductions. The problem is a useful scheduling-based abstraction layer between 3-SAT and monotone MRMP. Next, we present two problem variants that serve as tractability frontiers, i.e., tractable problems that become NP-hard with a slight formulation change.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

2 Preliminaries

Monotone MRMP. We define an instance as a set R of n rectangular robots in a planar workspace, where each robot $r \in R$ has a start and a target position, denoted by $s(r)$ and $t(r)$, respectively. The goal is to decide whether there is a *monotone motion plan*, i.e., a sequence of moves in which each robot $r \in R$ moves once from $s(r)$ to $t(r)$ without inducing collisions, while the other robots remain stationary. If such a motion plan exists, we say that the given instance is *feasible*.

Pivot Scheduling. An instance has the form $(V, \mathcal{C})$, where V is a set of *jobs* that come in pairs and $\mathcal{C}$ is a set of *ordering constraints*. Namely, let $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ be $2n$ distinct jobs, where for each $1 \leq i \leq n$, we view the jobs x_i and y_i as *paired*. Denote by V the set of all jobs, i.e., $V = \{x_1, y_1, \ldots, x_n, y_n\}$. Each ordering constraint $C \in \mathcal{C}$ is a set $C \subseteq V$ of three jobs. The task is to partition V into two subsets, a *before set* V_1 and an *after set* V_2, such that the following hold:

1. *Before constraints:* For each pair x_i, y_i, we have either $x_i \in V_1$ or $y_i \in V_1$.
2. *After constraints:* For each $C \in \mathcal{C}$, one of the jobs in C must be in V_2, i.e., $C \cap V_2 \neq \emptyset$.

Intuitively, the before/after constraints implicitly imply the existence of a distinguished *pivot* job with respect to which the input jobs must be ordered. To be precise, the partition into a before and after set, specifies which jobs come before and after the pivot job. Pivot Scheduling can be seen as a special case of generalized AND/OR Scheduling, which is NP-hard [10]. We show the following.

▶ **Theorem 1.** *Pivot Scheduling is NP-hard.*

3 Results

We present the two tractability frontiers.

Frontier 1: Nearly well-formed environments. To simplify MRMP, [5] introduce the notion of a *well-formed environment* (WFE), in which a robot located at an *endpoint*, i.e., a start or target position, cannot block other robots from reaching their target. That is, the following assumption **(*)** must hold for each robot: there exists a path to the robot's target that does not intersect with any endpoint belonging to other robots. In a WFE, robots can be moved to their target one by one in any order, and so any monotone MRMP instance with this property is feasible. A natural question is whether we can relax the WFE property and still solve the problem efficiently. We show that for a *nearly WFE*, in which there exists exactly one robot for which assumption **(*)** does not hold, monotone MRMP becomes NP-hard:

▶ **Theorem 2.** *Monotone MRMP in a nearly well-formed environment is NP-hard.*

Frontier 2: All but one robot have a fixed path. Another way to simplify MRMP is to restrict the paths that the robots can take. Suppose that as part of the input, we also get the path that each robot must take to its target. In this case, monotone MRMP can be solved in polynomial time by means of a precedence graph [1, 3]. However, even if we only have one robot that is not given a specific path, while the rest of the robots must follow given paths, then the problem becomes NP-hard:

▶ **Theorem 3.** *Monotone MRMP in which each robot is constrained to follow a path, except for one robot, is NP-hard even for unit square robots.*

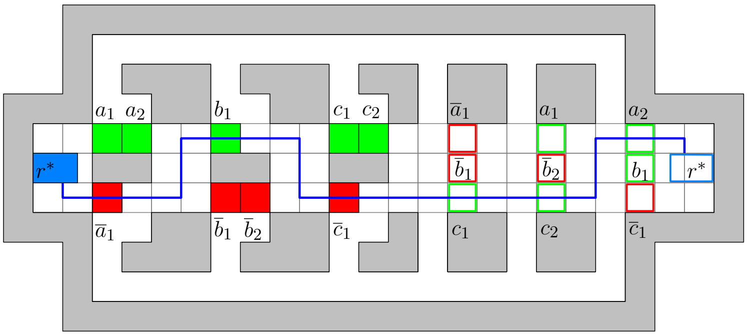


Figure 1 The monotone MRMP instance corresponding the to Pivot Scheduling with $V = \{a, \overline{a}, b, \overline{b}, c, \overline{c}\}$, $\mathcal{C} = \{\{\overline{a}, \overline{b}, c\}, \{a, \overline{b}, c\}, \{a, b, \overline{c}\}\}$. Obstacles are gray. The start and target positions are the filled and unfilled colored rectangles, respectively. The robots corresponding to the x_i's and y_i's (and their target positions) are colored green and red, respectively. Regular robots are labeled with unique indices in order to distinguish between appearances of the same job in $\mathcal{C}$. The path P of r^* in a motion plan corresponding to the partition $V_1 = \{\overline{a}, b, \overline{c}\}$, $V_2 = \{a, \overline{b}, c\}$ is shown (blue).

Figure 1 illustrates our construction for proving the hardness of the two aforementioned problem variants.[1] Complete proofs will appear in the full version of the paper and we expect to present additional variants at the YRF. Here we provide high level intuition on the hardness of monotone MRMP with nearly WFE. Given a Pivot Scheduling instance $S = (V, \mathcal{C})$, we construct a corresponding monotone MRMP instance I that is feasible if and only if S has a valid job partition. For each job $j \in V$ we have a corresponding set of robots $R(j)$ that contains a robot for each *appearance* of j in a constraint in $\mathcal{C}$. We call all the latter robots *regular* robots and they are unit squares. We have a special *pivot* robot r^*, which is a 1×1.5 rectangle, that has to move across the workspace from left to right, which is the only robot for which assumption **(*)** does not hold. The main idea is that since r^* is a bit fatter, it serves as the pivot job that must be carefully sequenced with respect to other robots in the motion plan. All the other robots can reach their targets regardless of when they move in a monotone motion plan, since they are thin enough to use the paths at the top and bottom of the construction. Start positions are placed such that for each x_i, y_i, either all of $R(x_i)$ or all of $R(y_i)$ must move before r^*. Similarly, targets are placed so that for each $C \in \mathcal{C}$, one of the robots corresponding to C must move after r^*.

References

1 Manuel Abellanas, Sergey Bereg, Ferran Hurtado, Alfredo García Olaverri, David Rappaport, and Javier Tejel. Moving coins. *Comput. Geom.*, 34(1):35–48, 2006.

2 Thomas Brocken, G. Wessel van der Heijden, Irina Kostitsyna, Lloyd E. Lo-Wong, and Remco J. A. Surtel. Multi-robot motion planning of k-colored discs is PSPACE-hard. In *FUN*, volume 157 of *LIPIcs*, pages 15:1–15:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

3 Stephen J. Buckley. Fast motion planning for multiple moving robots. In *ICRA*, pages 322–326. IEEE Computer Society, 1989.

4 Gruia Călinescu, Adrian Dumitrescu, and János Pach. Reconfigurations in graphs and grids. *SIAM J. Discret. Math.*, 22(1):124–138, 2008.

5 Michal Cáp, Peter Novák, Alexander Kleiner, and Martin Selecký. Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Trans Autom. Sci. Eng.*, 12(3):835–849, 2015.

[1] A different construction is required for the latter variant in the case of only unit square robots.

6 Adrian Dumitrescu and Minghui Jiang. On reconfiguration of disks in the plane and related problems. *Comput. Geom.*, 46(3):191–202, 2013.

7 Tzvika Geft, Dan Halperin, and Yonatan Nakar. Tractability Frontiers in Multi-Robot Coordination and Geometric Reconfiguration. *arXiv preprint arXiv:2104.07011*, 2021.

8 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343(1-2):72–96, 2005.

9 John E. Hopcroft, Jacob Theodore Schwartz, and Micha Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the "warehouseman's problem". *The International Journal of Robotics Research*, 3(4):76–88, 1984.

10 Rolf H. Möhring, Martin Skutella, and Frederik Stork. Scheduling with AND/OR precedence constraints. *SIAM J. Comput.*, 33(2):393–415, 2004.

11 Kiril Solovey and Dan Halperin. On the hardness of unlabeled multi-robot motion planning. *Int. J. Robotics Res.*, 35(14):1750–1759, 2016.

12 Paul G. Spirakis and Chee-Keng Yap. Strong NP-hardness of moving many discs. *Inf. Process. Lett.*, 19(1):55–59, 1984.

13 Rui Wang, Kai Gao, Daniel Nakhimovich, Jingjin Yu, and Kostas E. Bekris. Uniform object rearrangement: From complete monotone primitives to efficient non-monotone informed search. In *ICRA*, pages 6621–6627. IEEE, 2021.

An Efficient Algorithm for the Computation of Reeb Spaces from Roadmaps

Sarah Percival

Department of Biochemistry and Molecular Biology, Michigan State University, USA

Abstract

The Reeb graph is a tool from Morse theory that has recently found use in applied topology due to its ability to track the changes in connectivity of level sets of a function. The roadmap, a construction that arises in semi-algebraic geometry, is a one-dimensional set that encodes information about the connected components of a set. In this paper, we present a sketch of an algorithm with singly-exponential complexity that realizes the Reeb graph of a function $f\colon X \to \mathrm{R}$ as a semi-algebraic quotient using the roadmap of X with respect to f.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases Reeb graphs, roadmaps, semi-algebraic geometry, computational topology

Acknowledgements I want to thank Saugata Basu for his guidance throughout this project.

1 Introduction

Originally developed as a tool in Morse theory [6], the Reeb graph of a function has recently found use in applied topology due to its ability to track changes in connectivity of level sets of a function. The Reeb graph of a continuous function $f\colon X \to \mathrm{R}$, where X is a topological space and R is a real closed field, which for the sake of this paper can be taken to be $\mathbb{R}$, is defined as the set $X/\sim$, where $x \sim x' \in X$ if and only if $f(x) = f(x')$ and x and x' are in the same connected component of $f^{-1}(f(x)) = f^{-1}(f(x')) \subseteq X$.

Reeb graphs have previously been used to simplify and study data sets. Shinagawa, Kunii and Kergosien [7] observe that, since shape data of natural objects is often available in cross-sections, Reeb graphs can be used to reconstruct the original three-dimensional surface of an object. Doraiswamy and Natarajan [4] developed an algorithm that constructs the Reeb graph of a Morse function defined on a d-manifold in $O(n \log n (\log \log n)^3)$ time.

Our current work is motivated by a result in [1] which shows that there is a singly exponential upper bound on the Betti numbers of the Reeb graph of f in terms of the number and degrees of the polynomials defining X and f. It is often the case in semi-algebraic geometry that upper bounds on topological complexity of objects are closely related to the worst-case time complexity of algorithms computing the topological invariants of such objects, so we can expect that there is an algorithm with singly-exponential complexity which computes a semi-algebraic set homeomorphic to the Reeb graph of f. This leads us to the statement of our main result:

▶ **Theorem 1.** *There is an algorithm that takes as input a family $\mathcal{P} \subset \mathrm{R}[X_1, \ldots, X_k]$ of polynomials and formulas describing a semi-algebraic set S, and a continuous semi-algebraic map $f\colon S \to \mathrm{R}$, and computes as output a semi-algebraic set homeomorphic to the Reeb graph of f with time complexity $s^{k+1}d^{\mathcal{O}(k^2)}$, where s is a bound on the number of polynomials in $\mathcal{P}$ and d is a bound on their degree.*

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

2 Preliminaries

Because the goal of applied topology is to study real-world data, we choose to work within the framework of semi-algebraic geometry since the class of semi-algebraic sets is large enough to approximate any set that will occur in applications. This framework allows us to make use of prior results in semi-algebraic geometry related to algorithmic complexity. To ensure that the resulting Reeb graphs are semi-algebraic sets, we restrict ourselves to working with proper continuous maps, that is, maps with the property that inverse images of compact subsets are compact. We begin by defining concept of *semi-algebraic sets*, which play a critical role throughout this work.

▶ **Definition 2** (Semi-algebraic Sets). *We define the* semi-algebraic sets *of* R^k *as the smallest family of sets in* R^k *that contains the sets of the form* $\{x \in \mathrm{R}^k \mid P(x) = 0\}$ *and* $\{x \in \mathrm{R}^k \mid P(x) > 0\}$ *for some polynomial* $P \in \mathrm{R}[X_1, \ldots, X_k]$ *that is closed under finite unions, finite intersections, and complement. A map* $f : X \to Y$ *between two semi-algebraic sets is is* semi-algebraic *if its graph is a semi-algebraic set.*

The algorithm that proves Theorem 1 makes use of a tool from semi-algebraic geometry called the *roadmap* [3], which provides a useful description of the connected components of a set.

▶ **Definition 3** (Roadmap). *A* roadmap *for a semi-algebraic set* $S \in \mathrm{R}^k$ *is a semi-algebraic set* $M \subseteq S$ *of dimension at most one which satisfies the following properties:*

- *RM_1: For every semi-algebraically connected component D of S, the set $D \cap M$ is semi-algebraically connected, that is, $D \cap M$ cannot be represented as the disjoint union of two non-empty semi-algebraic sets that are both closed in S.*
- *RM_2: For every $x \in R$ and for every semi-algebraically connected component D' of $S_x = \{y \in \mathrm{R}^{k-1} \mid (x, y) \in S\}$, the intersection $D' \cap M$ is nonempty.*

Roadmaps in semi-algebraic geometry have been well-studied, and the authors of [2] produced an algorithm to compute the roadmap of a semi-algebraic set with complexity bounded singly-exponentially.

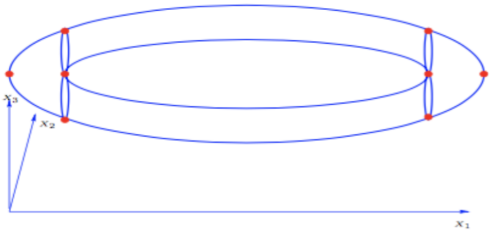


Figure 1 A roadmap of the torus T^2.

3 Algorithm to Compute the Reeb Graph

Because the proof of Theorem 1 is rather long, we instead provide an example of how the algorithm that proves Theorem 1 works. The full proof can be found in [5].

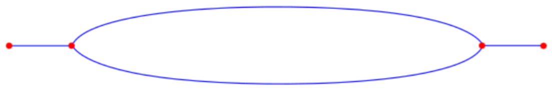


Figure 2 The Reeb graph of $\pi\colon T^2 \to \mathbb{R}$

▶ **Example 4.** Consider the projection map $\pi\colon T^2 \to \mathbb{R}$ of the torus onto the first coordinate and the corresponding roadmap, shown in Figure 1. The Reeb graph $\mathrm{Reeb}(\pi)$ is the set shown in Figure 2.

By choosing one curve per connected component of the fiber of each point on $\mathbb{R}$, as shown in Figure 3, truncating them, and joining them at their endpoints as in Figure 4, we obtain a set homeomorphic to $\mathrm{Reeb}(\pi)$. In the case of the roadmap of the torus, the subset chosen in Figure 3 is not unique; any subset satisfying this criterion will produce the Reeb graph after joining the curves in this manner.

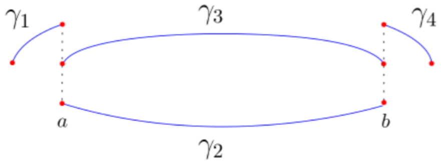


Figure 3

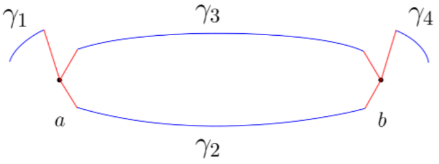


Figure 4

To ensure that the curves in Figure 3 are truncated in such a way that does not interfere with the topology of the resulting Reeb graph, we work over field of algebraic Puiseux series in ε with coefficients in $\mathbb{R}$, whose ordering makes $\varepsilon < x$ for every $x \in \mathbb{R}$, $x > 0$, as is typical in semi-algebraic geometry (see [2]). The local conic structure theorem, a standard result in semi-algebraic geometry (see [2]), guarantees that no two curves intersect within an ε-ball around their endpoints, therefore truncating each curve by ε does not change the topology of the set in Figure 3. The resulting set in Figure 4 is homeomorphic Reeb graph of $\pi\colon T^2 \to \mathbb{R}$.

References

1 Saugata Basu, Nathanael Cox, and Sarah Percival. On the Reeb spaces of definable maps. *arXiv e-prints*, Apr 2018. to appear in *Discrete and Computational Geometry*. `arXiv:1804.00605`.

2 Saugata Basu, Richard Pollack, and Marie-Francoise Roy. *Algorithms in Real Algebraic Geometry*. Springer-Verlag, 2011.

3 J. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1987.

4 Harish Doraiswamy and Vijay Natarajan. Computing Reeb graphs as a union of contour trees. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):249–262, 2013. `doi:10.1109/TVCG.2012.115`.

5 Sarah B Percival. Efficient computation of Reeb spaces and first homology groups, Jul 2021. URL: `https://hammer.purdue.edu/articles/thesis/Efficient_Computation_of_Reeb_Spaces_and_First_Homology_Groups/15078906/1`, `doi:10.25394/PGS.15078906.v1`.

6 Georges Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus de l'Académie des Sciences*, 222:847–849, 1946.

7 Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien. Surface coding based on Morse theory. *IEEE Computer Graphics and Applications*, 11(5):66–78, 1991.

The Edit Distance for Smoothings of Reeb Graphs

Erin Chambers
Saint Louis University

Kathleen Kramer
Saint Louis University

David Letscher
Saint Louis University

Abstract

In this work, we study Reeb graphs under the smoothing functor, and prove that a Reeb graph and its smoothing are ε apart in both interleaving distance and (combinatorial) graph edit distance.

2012 ACM Subject Classification cs.CG

Keywords and phrases Reeb graphs, smoothing, interleaving distance, Reeb graph edit distance

1 Introduction

Reeb graphs have become an important tool in computational topology for the purpose of visualizing continuous functions on complex spaces as a simplified discrete structure. Essentially, these graphs capture how level sets of a function evolve and behave in a topological space; components of each level set become a vertex, and edges connect vertices on adjacent level sets which are connected. We refer to recent surveys [4, 9] for a more exhaustive overview of their uses. Given their utility in shape analysis, it is natural to ask how can we measure the difference between Reeb graphs. There are many possible metrics studied; again, we refer to a recent survey [5] for details of the options. In this paper, we focus on two commonly studied such metrics, the interleaving distance d_I [7] and the Reeb graph edit distance d_E [3]. It is known that $d_E \leq 5d_I$ [3, 2]; in this paper, we show that for a Reeb graph and its smoothing, $d_I = d_E$.

2 Reeb Graphs and Smoothing

Formally, a Reeb graph is a pair (X, f) where X is topological space and $f : X \to \mathbb{R}$ is a continuous real-valued function. We assume that all Reeb graphs $\mathcal{R}_f$ are generic, so that all vertices have degree 1 or 3 and occur at distinct heights. In this work, we will study Reeb graph smoothings, first introduced in [7], defined as follows. For $\varepsilon \geq 0$, let $(f + Id) : X \times [-\varepsilon, \varepsilon] \to \mathbb{R}$ be defined as $(x, t) \mapsto f(x) + t$. We define the ε-smoothing $S_\varepsilon(X, f)$ to be the Reeb graph of $(X \times [-\varepsilon, \varepsilon], f + Id)$, we often refer to this as a smoothing of the Reeb graph. Recent work [1] completely characterizes the combinatorial effect of smoothing on a graph. In general, we see up forks and maximums being shifted upwards by ε and down forks and minimums being shifted downwards by ε. In addition, cycles disappear when the length of the cycle is shorter than 2ε, where the length of a cycle is the difference in height between the highest and the lowest vertices on the cycle. In the space $X \times [-\varepsilon, \varepsilon]$, f_ε is the induced height function where $f_\varepsilon(x, t) = f(x) + t$. Throughout this paper we will abuse this notation and also use f_ε to be the induced height function in $S_\varepsilon(X, f)$.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

$$\begin{tikzcd}
(G,f) \arrow[r, "\eta"] \arrow[dr, "\varphi"' near start] & S_\varepsilon(G,f) \arrow[r, "S_\varepsilon[\eta]"] \arrow[dr, dashed, "S_\varepsilon[\varphi]"' near start] & S_{2\varepsilon}(G,f) \\
(H,h) \arrow[r, "\eta"'] \arrow[ur, "\psi" near start] & S_\varepsilon(H,h) \arrow[r, "S_\varepsilon[\eta]"'] \arrow[ur, dashed, "S_\varepsilon[\psi]" near start] & S_{2\varepsilon}(H,h)
\end{tikzcd}$$

Figure 1 The Reeb graph interleaving distance is the smallest ε such that the diagram commutes. The map η is the composition of the inclusion of (G,f) into $(G \times [0,1], f_\varepsilon)$ and the quotient map from the thickening to $S_\varepsilon(G,f)$; see [7, 6] for more details.

We begin with a lemma that extends these ideas to study the effect of smoothing on edges of the Reeb graph:

▶ **Lemma 1.** *Let (X,f) be a generic Reeb graph with vertex set $V(X,f) = \{v_1, v_2, ..., v_n\}$. We denote the critical set $S = \{a_1 < a_2 < ... < a_n\}$ and assume that the vertices are sorted such that $f(v_i) = a_i$. We denote the set of edges as $E(X,f)$. Let C be a constant equal to half the minimum height difference between any connected down fork and up fork with the down fork above the up fork. Let $0 < \varepsilon < C$ such that $S_\varepsilon(X,f)$ is still generic. Then there is a bijection $\Phi : E(X,f) \to E(S_\varepsilon(X,f), f_\varepsilon)$.*

▶ **Corollary 2.** *For $0 < \varepsilon < C$ such that $S_\varepsilon(X,f)$ is still generic, there exists a graph isomorphism $\Psi : \mathcal{R}_f \to S_\varepsilon(X,f)$ induced by Φ.*

3 Reeb Graph Distances

This work focuses on showing the connections between common distance metrics on the space of Reeb graphs. Specifically, we will study the interleaving distance and the Reeb graph edit distance, both of which we briefly introduce in this section.

The idea of smoothing in fact originates from the interleaving distance [7]. An ε-interleaving with respect to S_ε is a pair of maps, $\varphi : (G,f) \to S_\varepsilon(H,h)$ and $\psi : (H,h) \to S_\varepsilon(G,f)$ such that the diagram 1 commutes. The *interleaving distance* is defined to be $d_I((G,f),(H,h)) = \inf_{\varepsilon} \{\text{there exists an } \varepsilon\text{-interleaving of } (G,f) \text{ and } (H,h)\}$

The Reeb graph edit distance [8] takes a combinatorial approach using a set of deformations to create a sequence transforming one Reeb graph into another. We assign cost values to each deformation and take the sum over the sequence to get a distance. The elementary deformations are birth (B-type), death (D-type), relabel (R-type), K_1-type, K_2-type, and K_3-type. We see a few examples of such these types of deformations in Figure 2. The K_i-type deformations are needed to change adjacencies, while R-type simply change the heights of vertices. The cost of the R-type and K_i-type deformations is the maximum displacement of any vertex moved during that edit; we emphasize that an R-type move may relabel many vertices, but the end cost is only the maximum distance any vertex is moved. The R-type deformation can be rather limiting as it does not allow for all adjacency changes that are needed, for these adjacency changes we turn to K_i deformation. The cost of B-type and D-type deformations is the absolute value of the difference of the function values of the two vertices we see being born or dying. We define a edit sequence T of $\mathcal{R}_f$ to be any finite ordered sequence $T = (T_1, T_2, ..., T_n)$ such that T_1 is an elementary deformation of $\mathcal{R}_f$, T_2 is an elementary deformation of $T_1(\mathcal{R}_f)$, ..., and T_n is an elementary deformation of $T_{n-1}T_{n-2}...T_2T_1(\mathcal{R}_f)$. If the result of the deformations is $\mathcal{R}_g$, then the edit sequence takes $\mathcal{R}_f$ to $\mathcal{R}_g$. The set of all such edit sequences is denoted $\mathcal{T}(\mathcal{R}_f, \mathcal{R}_g)$. The cost of an edit sequence

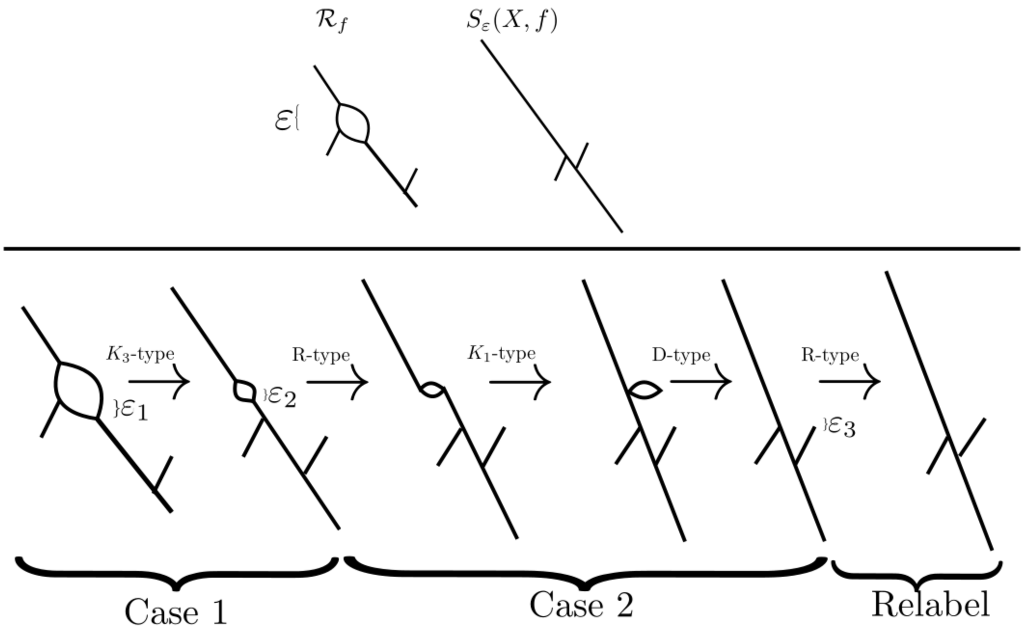


Figure 2 Examples of combinatorial edits to Reeb graphs.

is defined as $c(T) = \sum_{i=1}^{n} c(T_i)$, where $T = (T_1, T_2, ..., T_n)$ and $c(T_i)$ refers to the cost of the elementary deformation as described above. The *Reeb Graph edit distance*, d_E, between two Reeb graphs $\mathcal{R}_f, \mathcal{R}_g$ is defined to be $d_E((\mathcal{R}_f), (\mathcal{R}_g)) = \inf_{T \in \mathfrak{T}((\mathcal{R}_f),(\mathcal{R}_g))} \sum_{i=1}^{n} c(T_i)$.

4 Edit distance of smoothed Reeb graphs

Since smoothing is limited in how it changes a Reeb graph, we can analyze precisely which edits will transform a graph to its smoothing. When ε is chosen small enough (less than C from lemma 1), we will only need relabeling deformations. For larger ε, we will break up our edit sequence to focus on a single adjacency change at a time. We have 2 cases in which we see adjacency changes; see Figure 2. Essentially, case 1 deals with eliminating small loops, while case 2 deals with adjacent down forks above up forks (non loops) in which smoothing causes the down fork and up fork to swap height ordering. Outside of a small neighborhood of these changes, every other vertex is simply being relabeled by ε. We conclude with the following result:

▶ **Theorem 3.** *For any $\varepsilon > 0$ the interleaving distance between a Reeb graph and its ε-smoothing is equal to the Reeb graph edit distance.*

It remains to consider how to generalize this result to the universal Reeb graph edit distance, as described in [3], as well as if we can adapt this proof technique to identify other classes of Reeb graphs which attain equality in these two distance metrics.

References

1 Rehab Alharbi, Erin Wolf Chambers, and Elizabeth Munch. Realizable piecewise linear paths of persistence diagrams with Reeb graphs. ArXiv preprint 2107.04654, 2021.

2 Ulrich Bauer, Håvard Bakke Bjerkevik, and Benedikt Fluhr. Quasi-universality of Reeb graph distances. ArXiv preprint 2112.00720, 2021.

3 Ulrich Bauer, Claudia Landi, and Facundo Mémoli. The Reeb graph edit distance is universal. *Found Comput Math*, page 1441–1464, 2021.

4 Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science: Computational Algebraic Geometry and Applications*, 392(13):5 – 22, 2008.

5 Brian Bollen, Erin Chambers, Joshua A. Levine, and Elizabeth Munch. Reeb graph metrics from the ground up. ArXiv preprint 2110.05631, 2021.

6 Erin Wolf Chambers, Elizabeth Munch, and Tim Ophelders. A Family of Metrics from the Truncated Smoothing of Reeb Graphs. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

7 Vin de Silva, Elizabeth Munch, and Amit Patel. Categorified Reeb graphs. *Discrete & Computational Geometry*, pages 1–53, 2016.

8 Barbara Di Fabio and Claudia Landi. The edit distance for Reeb graphs of surfaces. *Discrete Comput Geom*, page 423–461, 2016.

9 Lin Yan, Talha Bin Masood, Raghavendra Sridharamurthy, Farhan Rasheed, Vijay Natarajan, Ingrid Hotz, and Bei Wang. Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. *Computer Graphics Forum*, 40(3):599–633, 2021.

On the Width of Complicated JSJ Decompositions

Kristóf Huszár
Inria Sophia Antipolis - Méditerranée and Université Côte d'Azur, France

Jonathan Spreer
School of Mathematics and Statistics F07, The University of Sydney, NSW 2006 Australia

Abstract

Motivated by the algorithmic study of 3-manifolds, we explore the structural relationship between the JSJ decomposition of a given 3-manifold and its triangulations. Building on work of Bachman, Derby-Talbot and Sedgwick, we show that a "sufficiently complicated" JSJ decomposition of a 3-manifold enforces a "complicated structure" for all of its triangulations. More concretely, we show that, under certain conditions, the treewidth (resp. pathwidth) of the graph that captures the incidences between the pieces of the JSJ decomposition of a 3-manifold yields a linear lower bound on the treewidth (resp. pathwidth) of the dual graph of any triangulation thereof.

Using these results, we give the first example of an infinite family of bounded-treewidth 3-manifolds with unbounded pathwidth. We also obtain Haken 3-manifolds with arbitrary large treewidth. Previously the existence of such 3-manifolds was only known in the non-Haken case.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Theory of computation → Fixed parameter tractability

Keywords and phrases computational 3-manifold topology, fixed-parameter tractability, generalized Heegaard splittings, weak reductions, JSJ decompositions, pathwidth, treewidth, triangulations

Funding *Kristóf Huszár*: Supported by the French government through the *3IA Côte d'Azur Investments in the Future* and *AlgoKnot* projects managed by the National Research Agency (ANR) under the reference numbers ANR-19-P3IA-0002 and ANR-20-CE48-0007, respectively.
Jonathan Spreer: Partially supported by the Australian Research Council's Discovery funding scheme (project no. DP190102259).

Acknowledgements We thank Clément Maria and Pascal Wild for proofreading this abstract and for suggesting several improvements for the exposition.

1 Background and Motivation

For computational purposes, a compact 3-*manifold* is often presented as a *triangulation*: a finite collection of abstract tetrahedra glued together along pairs of their triangular faces. In this setting, recent years have seen an emergence of *fixed-parameter tractable* (FPT) algorithms that efficiently solve computationally hard problems for triangulated 3-manifolds as soon as the *dual graph* of the input triangulation has bounded *treewidth*[1] [5, 6, 7, 8, 9].

To understand the scope of these algorithms, it is instructive to define the *treewidth* $\mathrm{tw}(\mathcal{M})$ *of a compact* 3-*manifold* $\mathcal{M}$ as the smallest possible treewidth of the dual graph of any triangulation of $\mathcal{M}$. The relationship between the treewidth and other topological invariants of 3-manifolds has recently been investigated in various contexts [11, 12, 13, 14, 19]. Together with Wagner [14] we have shown, for instance, that the treewidth of a *non-Haken*[2] 3-manifold

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

[1] The treewidth is a graph parameter that quantifies the "tree-likeness" of a given graph, cf. [3].

[2] A compact, orientable 3-manifold $\mathcal{M}$ is *non-Haken* if it is irreducible—i.e., every sphere embedded in $\mathcal{M}$ bounds a ball—and does not contain two-sided, properly embedded incompressible surfaces.

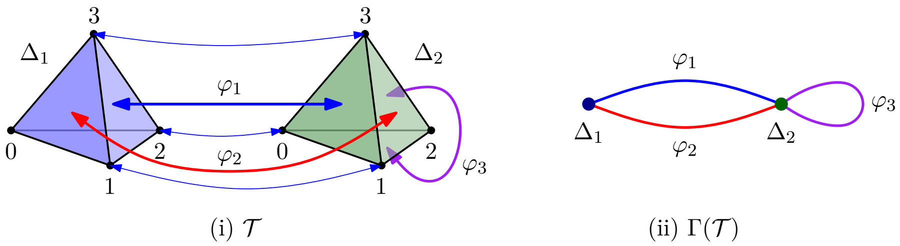


Figure 1 (i) Example of a triangulation $\mathcal{T}$ with two tetrahedra Δ_1 and Δ_2, and three face gluing maps φ_1, φ_2 and φ_3. The map φ_1 is specified to be $\Delta_1(123) \longleftrightarrow \Delta_2(103)$. (ii) The dual graph $\Gamma(\mathcal{T})$ of $\mathcal{T}$ that encodes the gluing maps between the tetrahedra. Reproduced from [12, Figure 1].

is always bounded below in terms of its *Heegaard genus*.[3] Combined with earlier work of Agol [1]—who constructed an infinite family of non-Haken 3-manifolds with arbitrary large Heegaard genus—this implies the existence of 3-manifolds with arbitrary large treewidth. In spite of the fact that, asymptotically, most triangulations of most 3-manifolds must have dual graphs of large treewidth [14, Appendix A], this collection described by Agol has remained, to this date, the only known family of 3-manifolds with arbitrary large treewidth.

In this project we unravel new structural connections between the triangulations of a given 3-manifold and its *JSJ decomposition*[4] [15, 16, 17]. Employing the machinery of *generalized Heegaard splittings*[5] [22], and building on work of Bachman, Derby-Talbot and Sedgwick [2], we show that, under suitable conditions, the dual graph of any triangulation of a given 3-manifold $\mathcal{M}$ inherits structural properties from the *decomposition graph* that encodes the incidences between the pieces of the JSJ decomposition of $\mathcal{M}$. This allows us to exhibit *Haken* 3-manifolds with arbitrary large treewidth, and to construct bounded-treewidth 3-manifolds with unbounded *pathwidth.* In what follows, we elaborate on these results.

2 The Main Result and Its Applications

▶ **Theorem 1** (Width inheritance)**.** *For any closed* 3*-manifold* $\mathcal{M}$ *with* sufficiently complicated[6] *torus gluings in its JSJ decomposition* $\mathcal{D}$*, the treewidth and pathwidth of* $\mathcal{M}$ *and those of the decomposition graph* $\Gamma(\mathcal{D})$ *of* $\mathcal{D}$ *satisfy*

$$\operatorname{tw}(\Gamma(\mathcal{D})) \leq 18 \cdot (\operatorname{tw}(\mathcal{M}) + 1) \quad \text{and} \quad \operatorname{pw}(\Gamma(\mathcal{D})) \leq 4 \cdot (3\operatorname{pw}(\mathcal{M}) + 1). \tag{1}$$

Outline of the proof. We only sketch the proof of the first inequality, as they are analogous. To prove that $\operatorname{tw}(\Gamma(\mathcal{D})) \leq 18(\operatorname{tw}(\mathcal{M}) + 1)$, we start with any triangulation $\mathcal{T}$ of $\mathcal{M}$. By our earlier work [14, Section 6], we can construct from $\mathcal{T}$ a generalized Heegaard splitting $\mathcal{H}$ of $\mathcal{M}$, where the genera of all level surfaces is bounded above by $18 \cdot (\operatorname{tw}(\Gamma(\mathcal{T})) + 1)$. By

[3] The Heegaard genus is one of the oldest invariants of 3-manifolds [10]. For a 3-manifold $\mathcal{M}$, it is the smallest number g, such that $\mathcal{M}$ can be obtained as a *Heegaard splitting* of genus g, i.e., from two handlebodies of genus g identified along their boundaries with a gluing homeomorphism, cf. [21].

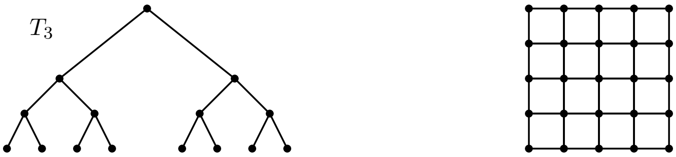
[4] A central result by Jaco–Shalen [15, 16] and Johannson [17] asserts that every closed, irreducible and orientable 3-manifold $\mathcal{M}$ admits a collection $\mathbf{T}$ of pairwise disjoint embedded tori, where each piece of the complement $\mathcal{M} \setminus \mathbf{T}$ is either Seifert fibered or atoroidal. A minimal such collection of tori is unique up to isotopy and gives rise to the so-called JSJ decomposition of $\mathcal{M}$. See [20] for a simplified proof.

[5] Generalized Heegaard splittings provide very useful instruments to navigate between different kinds of decompositions of 3-manifolds. They originate from [23]; see [22] for a comprehensive monograph.

[6] This is made precise via the notion of *c-distance* developed in [2, Section 2], cf. Definition 2.13 therein.

construction, $\mathcal{H}$ naturally admits a *sweep-out* $\mathcal{S} = \{\mathcal{S}_x : x \in H\}$ along a tree H. If $\mathcal{H}$ is not already *strongly irreducible*, we repeatedly perform *weak reductions* until we reach a strongly irreducible splitting $\mathcal{H}'$. Throughout this process we maintain the sweep-out to follow H. Crucially, weak reductions do not increase the genera of level surfaces [22, Section 5.2], thus $18 \cdot (\mathrm{tw}(\Gamma(\mathcal{T})) + 1)$ is still an upper bound on those in $\mathcal{H}'$. Now, by Corollary 4.5 of [2], each *JSJ torus* of $\mathcal{M}$ can be isotoped to coincide with a connected component of some *thin level* of $\mathcal{H}'$. This implies that, after a perturbation, the level set $\mathcal{S}_x$ is incident to at most $18 \cdot (\mathrm{tw}(\Gamma(\mathcal{T})) + 1) + 1$ JSJ pieces of $\mathcal{M}$, for any $x \in H$. Sweeping along H, we can construct a *tree decomposition* of $\Gamma(\mathcal{D})$ where each *bag* contains at most $18 \cdot (\mathrm{tw}(\Gamma(\mathcal{T})) + 1) + 1$ vertices, from which $\mathrm{tw}(\Gamma(\mathcal{D})) \leq 18 \cdot (\mathrm{tw}(\Gamma(\mathcal{T})) + 1)$ immediately follows.

Applications. With Theorem 1 at hand, the existence of bounded-treewidth 3-manifolds with arbitrary large pathwidth follows by taking the *complete binary tree* T_h as the decomposition graph of a JSJ decomposition with sufficiently complicated (in the sense of [2]) torus gluings between its constant-sized JSJ pieces, akin to a construction by Lackenby [18, Section 3]. The construction of Haken 3-manifolds with arbitrary large treewidth is similar, but instead of T_h we take the $k \times k$ grid as the decomposition graph. See Figures 2 and 3.



(i) $\mathrm{tw}(T_h) = 1$, $\mathrm{pw}(T_h) = \lceil h/2 \rceil$ (ii) $\mathrm{tw}(k \times k\text{-grid}) = k$

Figure 2 (i) The complete binary tree T_h of height h has pathwidth $\lceil h/2 \rceil$, cf. [4, Theorem 67]. (ii) The $k \times k$ grid-graph has pathwidth and treewidth both equal to k.

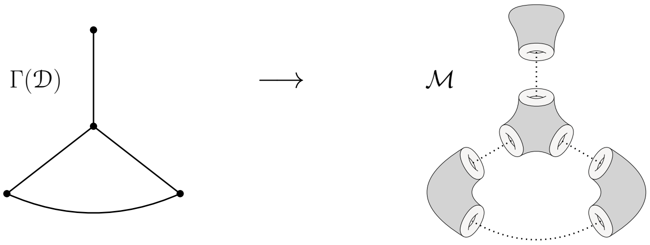


Figure 3 Schematic example of the construction. The pieces of $\mathcal{M}$ are described in [18, Sec. 3].

References

1 I. Agol. Small 3-manifolds of large genus. *Geom. Dedicata*, 102:53–64, 2003. doi:10.1023/B:GEOM.0000006584.85248.c5.

2 D. Bachman, R. Derby-Talbot, and E. Sedgwick. Heegaard structure respects complicated JSJ decompositions. *Math. Ann.*, 365(3-4):1137–1154, 2016. doi:10.1007/s00208-015-1314-9.

3 H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybern.*, 11(1-2):1–21, 1993. URL: https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/3417.

4 H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1–2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.

5 B. A. Burton and R. G. Downey. Courcelle's theorem for triangulations. *J. Comb. Theory, Ser. A*, 146:264–294, 2017. doi:10.1016/j.jcta.2016.10.001.

6 B. A. Burton, T. Lewiner, J. Paixão, and J. Spreer. Parameterized complexity of discrete Morse theory. *ACM Trans. Math. Softw.*, 42(1):6:1–6:24, 2016. doi:10.1145/2738034.

7 B. A. Burton, C. Maria, and J. Spreer. Algorithms and complexity for Turaev–Viro invariants. *J. Appl. Comput. Topol.*, 2(1–2):33–53, 2018. doi:10.1007/s41468-018-0016-2.

8 B. A. Burton and W. Pettersson. Fixed parameter tractable algorithms in combinatorial topology. In *Proc. 20th Int. Conf. Comput. Comb. (COCOON 2014)*, pages 300–311, 2014. doi:10.1007/978-3-319-08783-2_26.

9 B. A. Burton and J. Spreer. The complexity of detecting taut angle structures on triangulations. In *Proc. 24th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA 2013)*, pages 168–183, 2013. doi:10.1137/1.9781611973105.13.

10 P. Heegaard. Sur l'"Analysis situs". *Bull. Soc. Math. France*, 44:161–242, 1916. doi:10.24033/bsmf.968.

11 K. Huszár. *Combinatorial width parameters for 3-dimensonal manifolds.* PhD thesis, IST Austria, June 2020. doi:10.15479/AT:ISTA:8032.

12 K. Huszár. On the pathwidth of hyperbolic 3-manifolds. *Comput. Geom. Topol.*, 1(1):1:1–1:19, 2022. doi:10.57717/cgt.v1i1.4.

13 K. Huszár and J. Spreer. 3-Manifold triangulations with small treewidth. In *35th Int. Symp. Comput. Geom. (SoCG 2019)*, volume 129 of *LIPIcs. Leibniz Int. Proc. Inf.*, pages 44:1–44:20. Schloss Dagstuhl–Leibniz-Zent. Inf., 2019. doi:10.4230/LIPIcs.SoCG.2019.44.

14 K. Huszár, J. Spreer, and U. Wagner. On the treewidth of triangulated 3-manifolds. *J. Comput. Geom.*, 10(2):70–98, 2019. doi:10.20382/jogc.v10i2a5.

15 W. Jaco and P. B. Shalen. A new decomposition theorem for irreducible sufficiently-large 3-manifolds. In *Algebraic and Geometric Topology*, volume 32, part 2 of *Proc. Sympos. Pure Math.*, pages 71–84. Amer. Math. Soc., Providence, RI, 1978. doi:10.1090/pspum/032.2/520524.

16 W. H. Jaco and P. B. Shalen. *Seifert fibered spaces in 3-manifolds*, volume 220 of *Mem. Amer. Math. Soc.* Amer. Math. Soc., Providence, RI, 1979. doi:10.1090/memo/0220.

17 K. Johannson. *Homotopy equivalences of 3-manifolds with boundaries*, volume 761 of *Lect. Notes Math.* Springer, Berlin, 1979. doi:10.1007/BFb0085406.

18 M. Lackenby. Some conditionally hard problems on links and 3-manifolds. *Discrete Comput. Geom.*, 58(3):580–595, 2017. doi:10.1007/s00454-017-9905-8.

19 C. Maria and J. Purcell. Treewidth, crushing and hyperbolic volume. *Algebr. Geom. Topol.*, 19(5):2625–2652, 2019. doi:10.2140/agt.2019.19.2625.

20 W. D. Neumann and G. A. Swarup. Canonical decompositions of 3-manifolds. *Geom. Topol.*, 1:21–40, 1997. doi:10.2140/gt.1997.1.21.

21 M. Scharlemann. Heegaard splittings of compact 3-manifolds. In *Handbook of Geometric Topology*, pages 921–953. North-Holland, Amsterdam, 2001. doi:10.1016/B978-044482432-5/50019-6.

22 M. Scharlemann, J. Schultens, and T. Saito. *Lecture Notes on Generalized Heegaard Splittings.* World Sci. Publ., Hackensack, NJ, 2016. Draft version available at https://arxiv.org/abs/math/0504167. doi:10.1142/10019.

23 M. Scharlemann and A. Thompson. Thin position for 3-manifolds. In *Geometric topology (Haifa, 1992)*, volume 164 of *Contemp. Math.*, pages 231–238. Amer. Math. Soc., Providence, RI, 1994. doi:10.1090/conm/164/01596.

Edge-unfolding nested prismatoids

Manuel Radons
Technische Universität Berlin, Chair of Discrete Mathematics/Geometry, Germany

Abstract

A 3-Prismatoid is the convex hull of two convex polygons A and B which lie in parallel planes $H_A, H_B \subset \mathbb{R}^3$. Let A' be the orthogonal projection of A onto H_B. A prismatoid is called nested if A' is properly contained in B, or vice versa. We show that every nested prismatoid has an edge-unfolding to a non-overlapping polygon in the plane.

2012 ACM Subject Classification Computational Geometry

Keywords and phrases Dürers problem, edge unfolding polytopes, nets

Acknowledgements I want to thank my advisor, Michael Joswig, and Joseph O'Rourke for helpful discussions on the topic.

1 Introduction

Dürer's problem asks whether for every 3-polytope P there exists a spanning tree T of its edge graph, so that if we cut P along T the resulting surface can be unfolded into the plane without self-overlaps. We call a polytope for which such a spanning tree exists edge-unfoldable, or, briefer, unfoldable. It is known that non-convex polytopes which are combinatorially equivalent to a convex 3-polytope may be ununfoldable, cf. [2, 7, 1]. However, the classical question for convex polytopes remains open.

In an extended abstract for last year's YRF the author presented the outlines of a proof that sufficiently flat nested prismatoids are unfoldable. A prismatoid is the convex hull of two polygons A, B in parallel planes H_A, H_B. It is called nested if the orthogonal projection of A onto H_B is properly contained in B, or vice versa. We assume the former and then call A the top of P and B its base. The set of lateral facets of the prismatoid is called the band.

A nested prismatoid is a special case of a polyhedral cap, that is, a polytope P which possesses a designated facet F so that the orthogonal projection of $P' := P \setminus F$ onto F is one-to-one. A cap is nearly flat if the vertices of P' are "close" to F. The meaning of "close" depends on the context. Last year's near-flatness result was obtained in a manner similar to O'Rourke's result on unfoldability of nearly flat, acutely triangulated polyhedral caps [5, 4]: Establish a cutting scheme for a flat polytope, then lift the latter into a nearly flat polytope, and unfold. A different approach has now made it possible to remove the height constraint and arrive at the new straightforward statement.

▶ **Theorem 1.** *Every nested prismatoid is edge-unfoldable.*

2 Outline of the proof

We will here outline the proof of Theorem 1. The full paper can be found on the arXiv, cf. [6]. Also available under the link is the first version of the preprint, which contains, as an intermediary step, a simple derivation of last year's near-flatness result.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

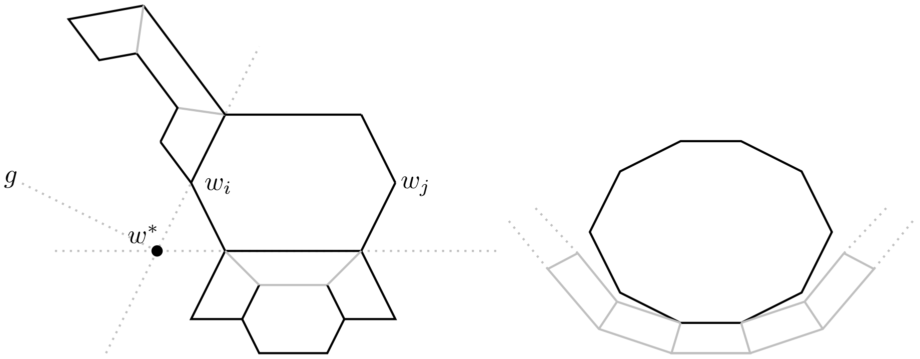


Figure 1 Unfolding scheme for flat prismoid, second case (left) and unfolding of lateral facets and top after flattening

The proof has two steps. In the first part we take a step back and prove unfoldability of nested prismoids. A prismoid is a prismatoid whose facets are trapezoids. It is already known that prismoids, nested or not, are unfoldable [3]. In this respect the first part is not new. But it presents a very specific cutting scheme. In the second part, the cutting scheme for prismoids acts as a catalyst. Instead of dealing with the full complexity of the prismatoid, one merely has to deal with the differences between prismoids and prismatoids.

Part 1 A key property of prismoids is that top and base A, B have the same number of vertices, say $v_1, \ldots, v_k$ and $w_1, \ldots, w_k$, and corresponding edges $(v_i, v_{i+1}), (w_i, w_{i+1})$ are parallel. We say the curvature at a vertex of A or B is the angle spanned by its outward normal cone. The curvature of a connected segment of the boundary of A or B is the sum of the curvatures at its interior vertices. We can always find indices i, j so that the curvatures of the four polygonal paths $(v_i, v_j) := [v_i, \ldots, v_j]$, (v_j, v_i), (w_i, w_j), (w_j, w_i) is smaller than π. This induces a subdivision of the band into two pieces. These two pieces cannot self-overlap when flattened into the plane.

Further, there exists an index ℓ so that the paths (v_i, v_ℓ) and $(v_{\ell+1}, v_j)$ either have length 0 or a curvature $\leq \frac{\pi}{2}$. These paths are *radially monotone*, that is, when traversing them, the distance to the starting point monotonically increases. If we attach the top A to the edge $(v_\ell, v_{\ell+1})$ and flatten the resulting polyhedral patch, the radial monotonicity of (v_i, v_ℓ) and $(v_{\ell+1}, v_j)$ ensures that they peel off A without intersecting it, cf. Figure 1, right.

It then remains to show that the two band pieces, one with attached top, one without, can be reattached to the base without causing intersections. Two cases are distinguished: There exists a base vertex w_i with curvature $\geq \frac{\pi}{2}$, or not, cf. Figure 1, left. For a detailed explanation we refer to the preprint [6].

A quick and dirty argument for unfoldability of nearly flat prismatoids goes as follows: Select w_i, w_j as before. Unlike in the prismoid case, there may be several lateral edges incident to each of these vertices. At each vertex cut one arbitrary lateral edge. At least one of the resulting top boundaries must have a curvature smaller π. This allows to attach the top and prevents self-intersection when flattening. If we select the height of the top sufficiently small, the other band piece will not self-intersect after flattening by a continuity argument and the fact that the band piece is closed. By a similar continuity argument the band pieces can be reattached to the base and will not cause intersections for small heights.

Part 2 The key to unfolding nested prismatoids of arbitrary height is to invest more work into the selection of the edges incident to w_i, w_j which are cut. Three main components

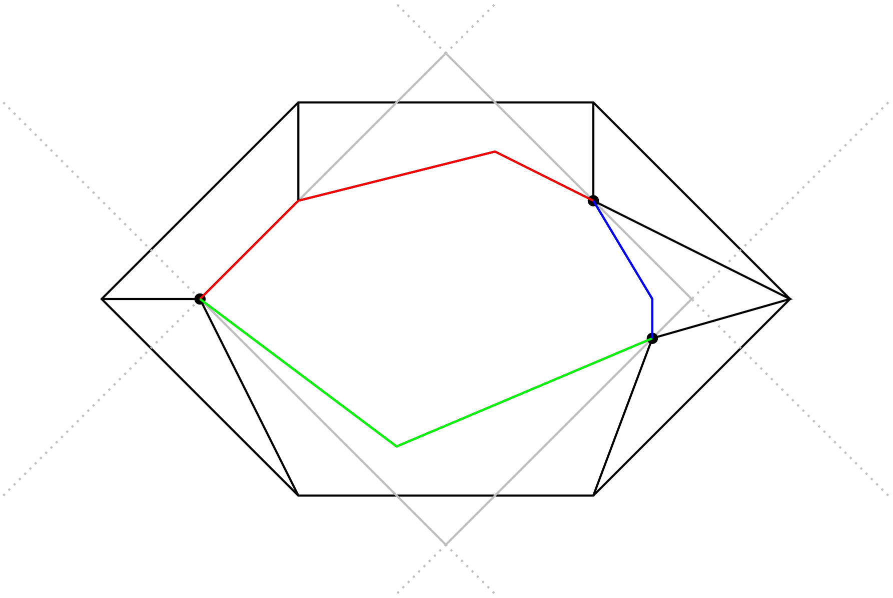


Figure 2 Shadow prismoid

enter into this work. The first is the observation that the curvature of convex polygonal paths solely depends on the relative angles of their first and last edge. In the analysis of the prismoid only the last and first edges of the top and base boundaries and their respective parallelity come into play.

The shadow prismoid technique is based on this observation. A key difficulty of the analysis of prismatoid bands is that they can contain triangles with an edge in the top or in the base and trapezoids in an arbitrary order. The shadow prismoid consists of four supporting lines to the top which are parallel to the four base edges incident to w_i and w_j, cf. Figure 2. Between the vertices where such a pair of lines meets the top, all lateral facets are triangles with an edge contained in the top. This simplifies the analysis of the bands.

The third idea is to embed the end triangles of bands in trapezoids to simplify angle-comparisons. If such an enhanced band unfolds overlap-free then so does the original.

References

1 Erik D. Demaine, Martin L. Demaine, and David Eppstein. Acutely triangulated, stacked, and very ununfoldable polyhedra. In *CCCG 2020, Saskatoon, Canada, August 5–7*, 2020.

2 Branko Grünbaum. No-net polyhedra. *Geombinatorics*, 11:111 – 114, 2002.

3 Joseph O'Rourke. Unfolding prismoids without overlap. *Unpublished manuscript*, 2001.

4 Joseph O'Rourke. Addendum to: Edge-unfolding nearly flat convex caps. *CoRR*, abs/1709.02433, 2017. URL: `http://arxiv.org/abs/1709.02433`, `arXiv:1709.02433`.

5 Joseph O'Rourke. Edge-Unfolding Nearly Flat Convex Caps. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry (SoCG 2018)*, volume 99 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 64:1–64:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2018/8777`, `doi:10.4230/LIPIcs.SoCG.2018.64`.

6 Manuel Radons. Edge-unfolding nested prismatoids. 2021. URL: `https://arxiv.org/abs/2105.00555`.

7 A.S. Tarasov. Polyhedra that do not admitnatural unfoldings. *Uspekhi Matematicheskikh.*

Algebraic Degrees of 3-Dimensional Polytopes

Mara Belotti
Chair of Discrete Mathematics/Geometry, Technische Universität Berlin, Germany

Michael Joswig
Chair of Discrete Mathematics/Geometry, Technische Universität Berlin, Berlin, Germany
MPI-MIS, Leipzig, Germany

Marta Panizzut
Chair of Discrete Mathematics/Geometry, Technische Universität Berlin, Germany

Abstract

We study realizations of 3-polytopes constrained by edge tangency conditions and determine the resulting algebraic degrees.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Discrete mathematics

Keywords and phrases realization spaces, polytopes, algebraic degrees

Related Version Full version hosted on arXiv
Full Version: https://arxiv.org/abs/2103.06835

Funding This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH$^+$ (EXC-2046/1, project ID 390685689).
Mara Belotti: Supported by DFG – Project-ID 286237555 – TRR 195
Michael Joswig: Supported by DFG – Project-ID 286237555 – TRR 195, and GRK 2434: Facets of Complexity.

1 Introduction

Given a polytope P, its *realization space* is the set of all polytopes which are combinatorially equivalent to P. Various parameterization exist depending on different ways of encoding a polytope. For example, in [5] polytopes are given by vertices and facets normals, and in [3] realization spaces are constructed based on slack matrices.

Yet all these realization spaces arise as real semialgebraic sets. Results of Mnëv [4] and Richter-Gebert [6] show that the realization space of a polytope of dimension at least 4 can be extremely topologically complicated. In contrast, realization spaces of 3-polytopes are always contractible, and each 3-polytope admits a realization with *rational coordinates*; cf. [8]. Our point of departure is the following strong version of Steinitz' theorem, proved by Springborn [7]: For every 3-connected planar graph there is a representation as the graph of a 3-polytope such that all edges are tangent to the unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$, and such that 0 is the barycenter of the contact points, i.e., the tangency points of the edges. This representation is unique up to rotations and reflections.

We introduce the following notions. A *Koebe realization* of a polytope P is a realization of P with edges tangent to the unit sphere $\mathbb{S}^2$, and a *Springborn realization* is a Koebe realization whose barycenter of the contact points is the origin. If we now look at the set of Koebe realizations inside the realization space of P, we obtain an interesting real

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

semialgebraic subset. The questions we address here are the following: Does P admit rational Koebe and Springborn realizations? If not, what is the minimal algebraic degree of these realizations?

These questions are related to computational aspects in geometric combinatorics. Bobenko and Springborn [1] developed a variational method to compute Springborn realizations numerically. Our work highlights that any exact computation is much more involved, since the algebraic degrees are unbounded.

2 Main Results

Given a Koebe realization Q of a polytope P, the field extension $\mathbb{Q}[Q]$ is the extension of $\mathbb{Q}$ given by the (vertex) coordinates. This turns out to be the same as the extension of $\mathbb{Q}$ given by the contact points of Q. The *Koebe degree* $\kappa(P)$ is the minimum degree $|\mathbb{Q}[Q] : \mathbb{Q}|$ among all possible Koebe realizations Q.

▶ **Theorem 1.** *Let P be a 3-polytope with n vertices and a triangular facet. Then its Koebe degree is at most*

$$\kappa(P) \;\leq\; 8^{2^{3n-9}-1} \cdot 2^{-(3n-9)} \;<\; 2^{2^{3n}} \;. \tag{1}$$

Note that if P does not have a triangular facet, then its dual does; in that case the above result bounds the Koebe degree in terms of the facets.

To prove Theorem 1, we first remark that given a triangular facet, there is a finite number of Koebe realizations of P such that the three vertices of the facet have rational coordinates. These realizations are cut out by rational polynomial equations and inequalities of degree at most 4. We can then use bounds on the algebraic degree of their solutions coming from cylindrical algebraic decomposition CAD [2].

Two Koebe realizations differ by a projective transformation preserving the sphere $\mathbb{S}^2$. The action of these transformations on $\mathbb{S}^2$ is equivalent to the action of $\mathrm{PGL}_2(\mathbb{C})$ on $\mathbb{C} \cup \{\infty\}$ through the stereographic projection and one obvious invariant of this last group is given by the cross ratio of 4 points.

▶ **Proposition 2.** *Let Q be a Koebe realization of P with contact points in $\mathbb{K} = \mathbb{Q}[Q]$. Then the cross ratio of the stereographic projections of any four contact points lies in $\mathbb{L} = \mathbb{K}[i]$, and the degree of its real and imaginary part over $\mathbb{Q}$ is a lower bound for $\kappa(P)$.*

In the paper we also focus on two classes of polytopes which give more insight concerning the behavior of these degrees. Analyzing the class of *bipyramids*, we were able to show that the Koebe degree is not uniformly bounded on the class of polytopes. We use ϕ to denote the Euler's totient function, so $\phi(k)$ counts the positive integers smaller than k and relatively prime to k.

▶ **Theorem 3.** *Let $k \geq 4$ be an integer. Then the Koebe degree $\kappa(B_k)$ is at least $\phi(k)/4$. In particular, the Koebe degree of a 3-polytope is not bounded by any constant.*

Interestingly, we observed the opposite behavior for the class of *stacked polytopes*, that is simplicial polytopes obtained by starting with a 3–simplex and successively adding vertices beyond a facet. For stacked polytopes, we can always find a rational Koebe realization.

▶ **Theorem 4.** *The Koebe degree of any stacked 3-polytope equals one.*

It is decidable whether a polytope admits a rational Koebe realization or not. If P is a polytope which admits a rational Koebe realization Q and Q' is a Koebe realization of P with three rational contact points, then Q' is rational. We can therefore solve a finite semialgebraic system, e.g., by cylindrical algebraic decomposition CAD, to find a realization of P with three rational contact points, and establish from the result if the Koebe degree of P is one or not. However, often this is prohibitively expensive. More generally, the Koebe degree is always computable, also via CAD.

The *Springborn degree* $\sigma(P)$ is the minimum degree $|\mathbb{Q}[Q] : \mathbb{Q}|$ among all possible Springborn realizations Q. Two Springborn realizations differ by the action of an element of the orthogonal group $\mathrm{O}_3(\mathbb{R})$, which in fact has a lot of nice invariants. The degree of the extension given by the volume of the polytope or by the squared norm of a vertex provide lower bounds for the Springborn degree, which gets easier to compute than the Koebe degree.

▶ **Example 5.** The six columns of the matrix

$$\begin{pmatrix} \pm\sqrt{2} & 0 & 0 \\ 0 & \pm\sqrt{2} & 0 \\ 0 & 0 & \pm\sqrt{2} \end{pmatrix}$$

provide a degree two Springborn realization of the octahedron. We can conclude that the Springborn degree is two because its volume equals $\frac{8\sqrt{2}}{3}$.

The following result connects the two notions of degree.

▶ **Theorem 6.** *For a 3-polytope P with m edges we have*

$$\kappa(P) \ \leq \ \sigma(P) \ \leq \ \frac{(2m+2)^7}{4} \cdot \kappa(P) \ .$$

References

1 Alexander I. Bobenko and Boris A. Springborn. Variational principles for circle patterns and Koebe's theorem. *Trans. Amer. Math. Soc.*, 356(2):659–689, 2004. doi:10.1090/S0002-9947-03-03239-2.

2 George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975)*, Lecture Notes in Comput. Sci., pages 134–183. Springer, Berlin, 1975.

3 Joao Gouveia, Antonio Macchia, Rekha R. Thomas, and Amy Wiebe. The slack realization space of a polytope. *SIAM J. Discrete Math.*, 33(3):1637–1653, 2019.

4 N. E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In *Topology and geometry—Rohlin Seminar*, volume 1346 of *Lecture Notes in Math.*, pages 527–543. Springer, Berlin, 1988. doi:10.1007/BFb0082792.

5 Laith Rastanawi, Rainer Sinn, and Günter M. Ziegler. On the dimensions of the realization spaces of polytopes. *Mathematika*, 67(2):342–365, 2021.

6 Jürgen Richter-Gebert. *Realization spaces of polytopes*, volume 1643 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1996. doi:10.1007/BFb0093761.

7 Boris A. Springborn. A unique representation of polyhedral types. Centering via Möbius transformations. *Math. Z.*, 249(3):513–517, 2005. doi:10.1007/s00209-004-0713-5.

8 Günter M. Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. doi:10.1007/978-1-4613-8431-1.

Stability of circumcentres for small metric perturbations of spaces of constant curvature [*]

Hana Dal Poz Kouřimská
IST Austria
[Klosterneuburg, Austria]

Mathijs Wintraecken
IST Austria
[Klosterneuburg, Austria]

Abstract

In this short note we explore how circumcentres of simplices in manifolds of almost constant sectional curvature are perturbed upon distortion of the metric.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Riemannian manifolds, adaptive sampling, Delaunay triangulation

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement No. 339025 GUDHI (Algorithmic Foundations of Geometry Understanding in Higher Dimensions), from the Wittgenstein Prize, Austrian Science Fund (FWF), grant no. Z 342-N31, and from the DFG Collaborative Research Center TRR 109, 'Discretization in Geometry and Dynamics', Austrian Science Fund (FWF), grant no. I 02979-N35.
Mathijs Wintraecken: Supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 754411 and the by the FWF Der Wissenschaftsfonds's Lise Meitner programme under grant agreement No. M–3073.

Acknowledgements We are greatly indebted to Jean-Daniel Boissonnat, Ramsay Dyer, and Mael Rouxel-Labbé for discussion. We thank all members of the Edelsbrunner group for the atmosphere in which this research was conducted.

1 Introduction

Triangulating manifolds is difficult mainly because of the occurrence of badly shaped simplices. To battle this difficulty, quality criteria on point samples of a manifold have been established that guarantee the Delaunay complex of the sample to be a triangulation, i.e., homeomorphic to the manifold. These criteria demand that the density and protection (Figure 1 right) of the point sample are large compared to *how far the manifold is from being flat*, and its injectivity radius (Figure 1 left) [1, 2, 3, 5]. However, it is more natural to formulate these criteria in terms of *how far the manifold is from having constant sectional curvature*, since Delaunay triangulating sphere or hyperbolic spaces is as difficult as Delaunay triangulating the Euclidean space.

In our ongoing work, we derive quality criteria that guarantee a successful construction of Delaunay triangulations in manifolds of nearly constant curvature. These criteria can be used in adaptive sampling based on the change of curvature. Algorithms dealing with triangulations of hyperbolic spaces, for example, are often plagued by numerical imprecision [8,

[*] This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

Remark 4.9 (i)]. Our work helps succumb this imprecision by interpreting it as a distortion on the hyperbolic metric, thereby providing stability guarantees of the algorithms [4, 6]. In this note we present a key element of our work — we control the distortion of circumcentres of simplices in spaces of constant curvature upon a small distortion of the metric.

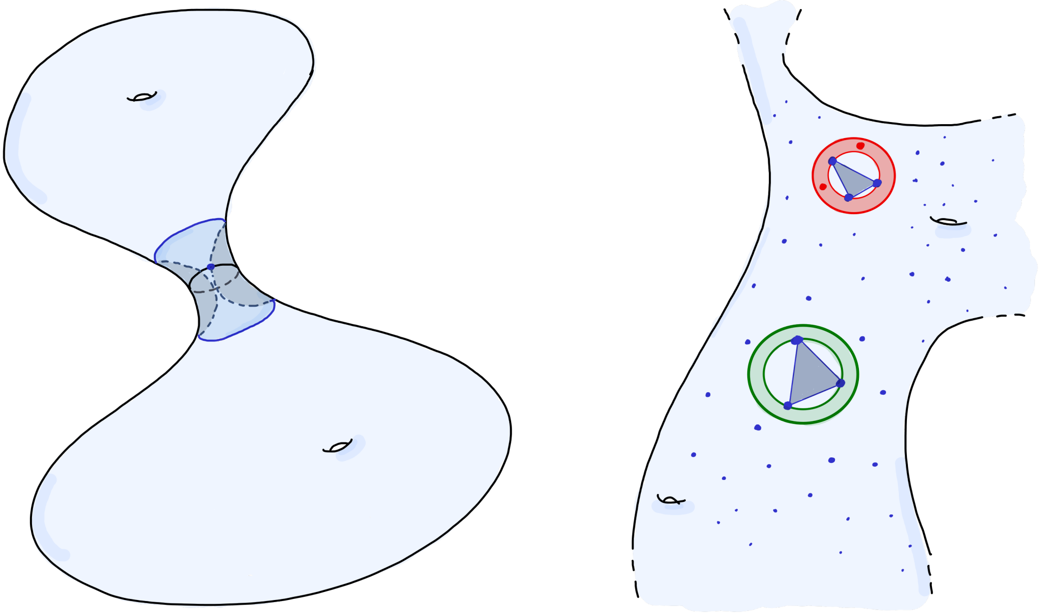


Figure 1 On the left: the injectivity radius. On the right: the triangle with the green circumcircle is protected; the triangle with the red circumcircle is not, since its red belt contains two points.

Space forms A *space form* is a sphere or a hyperbolic space. We denote n-dimensional space forms by $\mathbb{H}^n(K)$ or $\mathbb{H}(K)$, and rescale the space such that $|K| = 1$. We model the space forms on the $n-$dimensional sphere $\mathbb{S}^n$ (if $K = 1$) or the $n-$dimensional hyperboloid $\mathbb{H}^n$ (if $K = -1$). We use d or $d_{\mathbb{H}(K)}$ to denote the metric on either $\mathbb{S}^n$ or $\mathbb{H}^n$, and denote the orthogonal complement of a set S with respect to either the Euclidean or the Minkowski product by $S^\perp$.

Simplices in space forms An n-dimensional *simplex* σ in $\mathbb{H}^n(K)$ is a convex hull of a set of $n+1$ points $v_0, \ldots, v_n \in \mathbb{H}^n(K)$. If σ is non-degenerate, there exists a unique n-sphere in $\mathbb{H}^n(K)$ containing the vertices of σ. We call this sphere the *circumsphere* of σ, and its centre the *circumcentre* of σ. The circumcentre of σ is the intersection of bisectors of pairs of its vertices v_i and v_j, where a *bisector* of v_i and v_j is the set of all points in $\mathbb{H}^n(K)$ that are equidistant to v_i and v_j. We further define the *secant simplex* $\overline{\sigma}$ as the convex hull of the vertices $v_0, \ldots, v_n$ in $\mathbb{R}^{n+1}$. The secant simplices are illustrated in Figure 2.

Riemannian simplices and metric distortion We study full-dimensional, non-degenerate simplices[1] on a Riemannian manifold $\mathcal{M}$ which is essentially a space form, with a slightly

[1] Such a Riemannian simplex can be constructed by Karcher's centre of mass construction [7].

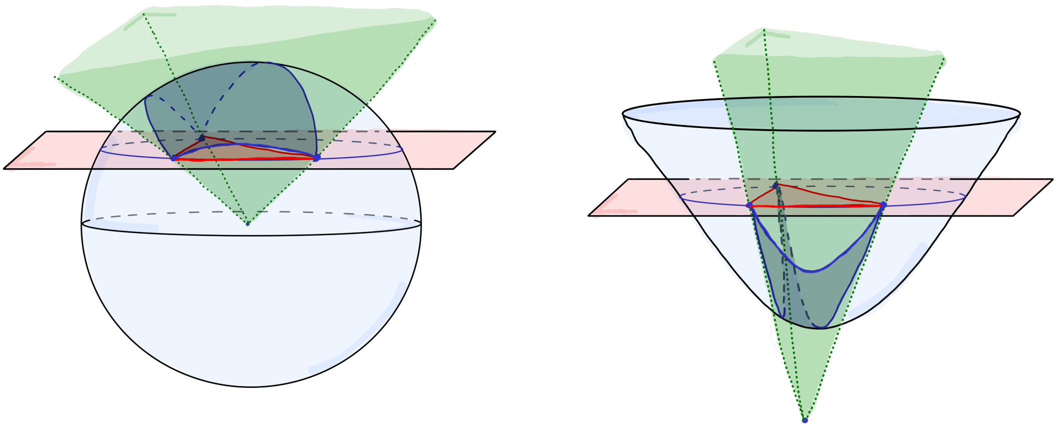


Figure 2 A two-dimensional simplex (in blue) and its secant simplex (in red).

distorted metric. Every such Riemannian simplex $\sigma_{\mathcal{M}}$ possesses one or multiple circumcentre(s), that lie(s) in the intersection of the bisectors of its vertices. Our goal is to bound these circumcentres in a small neighbourhood. We achieve this bound by comparing $\sigma_{\mathcal{M}}$ to a 'similar' simplex σ in a space form $\mathbb{H}(K)$ — we assume that there exists a (local) near-isometry Ψ between $\mathcal{M}$ and $\mathbb{H}(K)$. That is, there exists a distortion parameter $\nu > 0$ such that, for all points x and y in a neighbourhood of $\sigma_{\mathcal{M}}, \left| d_{\mathcal{M}}(x, y) - d_{\mathbb{H}(K)}(\Psi(x), \Psi(y)) \right| < \nu$.

We adopt the notation of Figure 3, where $C_{\sigma_{\mathcal{M}}}$ and C denote circumcentres of $\sigma_{\mathcal{M}}$ and σ, respectively. Our goal is to upper-bound the distance $d_{\mathbb{H}(K)}(C, \Psi(C_{\sigma_{\mathcal{M}}}))$.

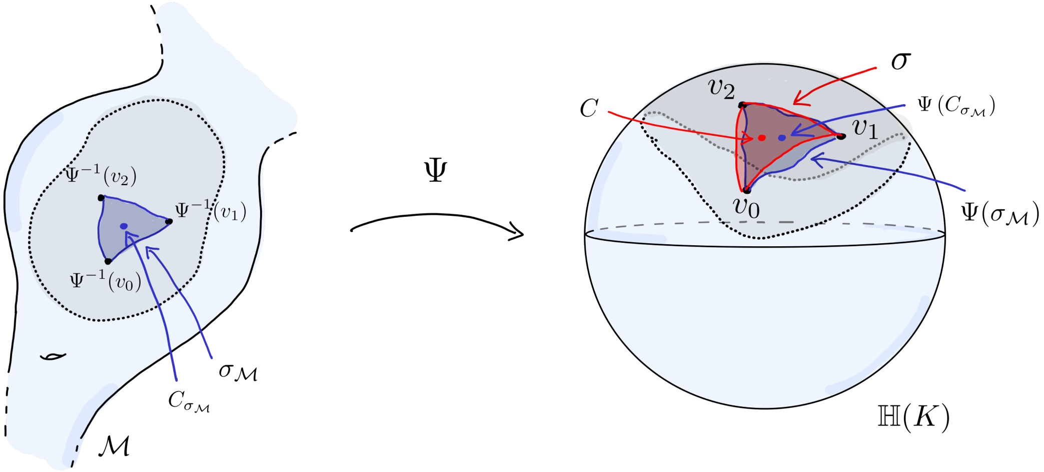


Figure 3 We compare the image of a Riemannian simplex with the convex hull of its vertices.

Let p, q be two vertices of σ. Since the point $C_{\sigma_{\mathcal{M}}}$ lies at the bisector of $\Psi^{-1}(p)$ and $\Psi^{-1}(q)$, its image satisfies $\left| d_{\mathbb{H}(K)}(p, \Psi(C_{\sigma_{\mathcal{M}}})) - d_{\mathbb{H}(K)}(q, \Psi(C_{\sigma_{\mathcal{M}}})) \right| \leq 2\nu$. We call the set of all points in $\mathbb{H}(K)$ satisfying this bound the *thickened bisector* of p and q (see Figure 4), and denote it by $\mathfrak{B}_\nu(p, q)$ or $\mathfrak{B}_\nu^{hyp}(p, q)$ when $K = 1$ or $K = -1$, respectively.

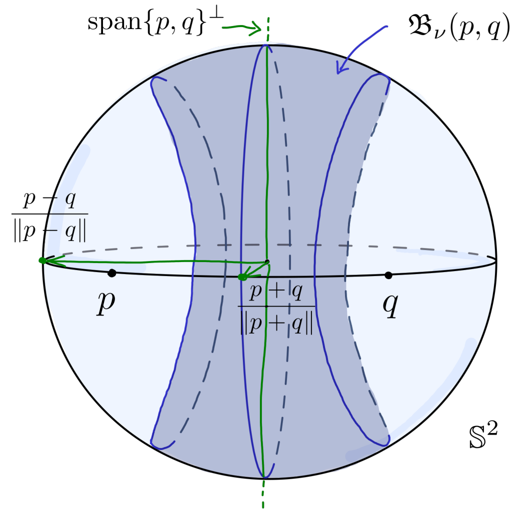


Figure 4 The thickened bisector of p and q.

2 Geometric interpretation of thickened bisectors

Let $\lambda = d(p,q)/2$ be **half** of the distance between p and q, and let $\nu \in [0,\lambda]$. If $\mathbb{H}(K) = \mathbb{S}^n$ we denote the following union of ellipses related to p and q by $\mathscr{E}_\nu(p,q)$:

$$\mathscr{E}_\nu(p,q) = \bigcup_{\rho\in[0,\nu]} \left\{ \frac{\cos\rho\cos t}{2\cos^2\lambda}(p+q) + \frac{\sin\rho\sin t}{2\sin^2\lambda}(p-q) \ \mid t \in [0,2\pi] \right\} \subseteq \mathbb{R}^{n+1}. \tag{1}$$

If $\mathbb{H}(K) = \mathbb{H}^n$, we denote the following union of hyperbolas related to p and q by $\mathscr{H}_\nu(p,q)$:

$$\mathscr{H}_\nu(p,q) = \bigcup_{\rho\in[0,\nu]} \left\{ \frac{\cosh\rho\cosh t}{2\cosh^2\lambda}(p+q) + \frac{\sinh\rho\sinh t}{2\sinh^2\lambda}(p-q) \mid t \in \mathbb{R} \right\} \subseteq \mathbb{R}^{n+1}. \tag{2}$$

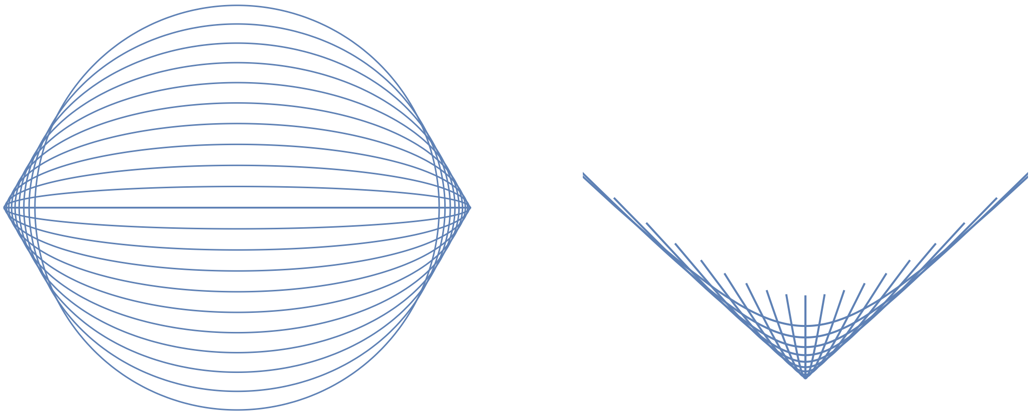


Figure 5 The sets $\mathscr{E}_\nu(p,q)$ resp. $\mathscr{H}_\nu(p,q)$.

The sets $\mathscr{E}_{\nu=\lambda}(p,q)$ and $\mathscr{H}_{\nu=\lambda}(p,q)$ are depicted in Figure 5.

▶ **Definition 1** (Elliptic and hyperbolic cylinder). **1.** *The* elliptic cylinder $Z_\nu(p,q)$ *is the cartesian product of* $\mathscr{E}_\nu(p,q)$ *and the* $(n-1)$*-plane* $\operatorname{span}\{p,q\}^\perp \subseteq \mathbb{R}^{n+1}$*,*

$$Z_\nu(p,q) = \left\{v + u \mid v \in \mathscr{E}_\nu(p,q), u \in \operatorname{span}\{p,q\}^\perp\right\}.$$

2. *The* hyperbolic cylinder $Z_\nu^{hyp}(p,q)$ *is the cartesian product of* $\mathscr{H}_\nu(p,q)$ *and the* $(n-1)$*-plane* $\operatorname{span}\{p,q\}^\perp \subseteq \mathbb{R}^{n+1}$*,*

$$Z_\nu^{hyp}(p,q) = \left\{v + u \mid v \in \mathscr{H}_\nu(p,q), u \in \operatorname{span}\{p,q\}^\perp\right\}.$$

The sets $Z_\nu(p,q)$ and $Z_\nu^{hyp}(p,q)$ are unions of degenerate quadrics.

▶ **Proposition 2** (Thickened bisectors are cylinders). *The thickened bisector of* p *and* q *is the intersection of* $\mathbb{H}^n\{\pm 1\}$ *with the elliptic resp. hyperbolic cylinder:*

$$\mathfrak{B}_\nu(p,q) = \mathbb{S}^n \cap Z_\nu(p,q), \tag{3}$$

$$\mathfrak{B}_\nu^{hyp}(p,q) = \mathbb{H}^n \cap Z_\nu^{hyp}(p,q). \tag{4}$$

Recall that the *height* of the simplex $\bar{\sigma}$ is the minimum altitude of its vertices (see Figure 6).

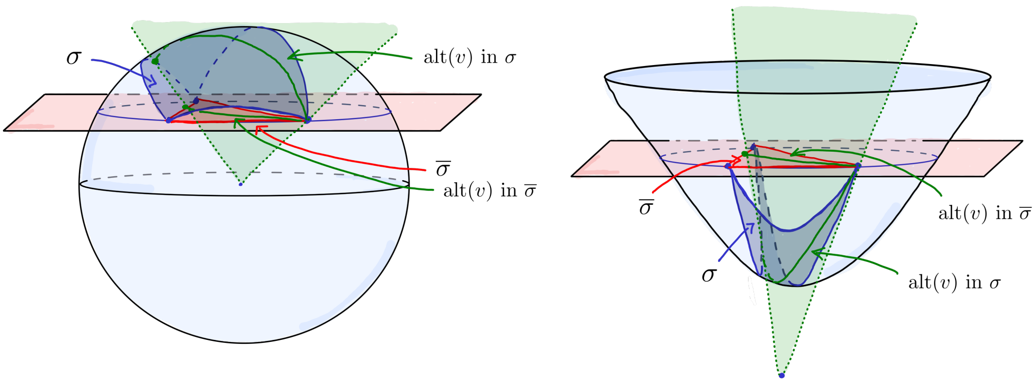


Figure 6 An altitude alt(v) of σ and of the secant simplex $\bar{\sigma}$.

▶ **Proposition 3** (Intersection of cylinders). *Let* $0 < \nu \leq \min_{i\neq j} d(v_i, v_j)/2$*. The distance* $d_{\mathbb{H}(K)}(C, \Psi(C_{\sigma_\mathcal{M}}))$ *is bounded from above by* ν*, the radius* r *of the simplex* σ*, and the height of the secant simplex* $\overline{\sigma}$*.*

$$d_{\mathbb{H}(K)}(C, \Psi(C_{\sigma_\mathcal{M}})) \leq \begin{cases} \arcsin\left(\frac{(n+1)\sin\nu}{\operatorname{height}(\overline{\sigma})}\right), & \text{if } K = 1, \\ \operatorname{arctanh}\left(\frac{(n+1)\tanh\nu}{\tanh^2 r \operatorname{height}(\overline{\sigma})}\right), & \text{if } K = -1. \end{cases}$$

References

1 Jean-Daniel Boissonnat, Ramsay Dyer, and Arijit Ghosh. Delaunay triangulation of manifolds. *Found. Comput. Math.*, 18(2):399–431, 2018. doi:10.1007/s10208-017-9344-1.

2 Jean-Daniel Boissonnat, Ramsay Dyer, Arijit Ghosh, and Mathijs Wintraecken. Local criteria for triangulation of manifolds. In *34th International Symposium on Computational Geometry (SoCG 2018)*, volume 99, pages 9:1–9:14, 2018. doi:10.4230/LIPIcs.SoCG.2018.9.

3 Jean-Daniel Boissonnat, Mael Rouxel-Labbé, and Mathijs Wintraecken. Anisotropic triangulations via discrete Riemannian Voronoi diagrams. *SIAM Journal on Computing*, 48(3):1046–1097, 2019. doi:10.1137/17M1152292.

4 Vincent Despré, Jean-Marc Schlenker, and Monique Teillaud. Flipping geometric triangulations on hyperbolic surfaces. In *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164, pages 35:1–35:16, 2020. `doi:10.4230/LIPIcs.SoCG.2020.35`.

5 Ramsay Dyer, Gert Vegter, and Mathijs Wintraecken. Riemannian simplices and triangulations. *Geometriae Dedicata*, 2015. `doi:10.1007/s10711-015-0069-5`.

6 Iordan Iordanov and Monique Teillaud. Implementing Delaunay triangulations of the Bolza surface. In *33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77, pages 44:1–44:15, 2017. `doi:10.4230/LIPIcs.SoCG.2017.44`.

7 H. Karcher. Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.*, 30(5):509–541, 1977. `doi:10.1002/cpa.3160300502`.

8 Boris Springborn. Ideal hyperbolic polyhedra and discrete uniformization. *Discrete and Computational Geometry*, 2019. `doi:10.1007/s00454-019-00132-8`.

Euler-Fourier transform of constructible functions

Vadim Lebovici ✉
Université Paris-Saclay, CNRS, Inria, Laboratoire de Mathématiques d'Orsay, 91405, Orsay, France.

Abstract

We present an integral transform of constructible functions, the *Euler-Fourier transform*, combining Lebesgue integration and constructible pushforward — a topological dimensionality reduction. Lebesgue integration gives access to regularity results while constructible pushforward conveys topological information, making it strictly more discriminating than the classical Fourier transform. This transform is an example of the more general notion of hybrid transform defined in [8]. In this note, we adapt the exposition to this specific example and illustrate it in various ways. We also show that it can be efficiently computed in practical scenarios.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases topological data analysis, Euler calculus, constructible functions, integral transforms

Related Version A full version of the paper is available at `https://arxiv.org/abs/2111.07829`.

1 Introduction

Euler calculus – the integral calculus of constructible functions with respect to the Euler characteristic – is of increasing interest in topological data analysis and computational geometry. Already in [9], it was developed as an alternative definition of convolution for polygonal tracings with multiplicities, a useful notion in robotics [5, 7]. In persistence theory, Schapira's result on Radon transform [11] positively answers an important question [2, Thm. 4.11]: are two constructible subsets of $\mathbb{R}^n$ with the same persistent homology in all degrees and for all height filtrations equal? More generally, the constructible functions naturally associated to multiparameter persistent modules stand as simple and well-behaved, albeit incomplete, invariants of these objects. For instance, the *persistent magnitude* [4] is actually defined on the constructible functions associated to the persistence modules.

In [8], we introduced a general definition and conducted a systematic study of integral transforms combining Lebesgue integration and Euler calculus for constructible functions. Such transforms generalize the *Bessel* and *Fourier* transforms of Ghrist and Robinson [3], as well as the *Euler characteristic of barcodes* of Bobrowski and Borman [1]. In this note, we illustrate the theory on one example, the *Euler-Fourier transform*. We state some of its characteristics and illustrate its differences from its classical analogue in various situations (see Figure 1). More general results are proven in [8].

2 Definition

A function $\varphi : \mathbb{R}^n \to \mathbb{Z}$ is called *constructible*[1] if it can be written as a finite sum $\varphi = \sum_{i=1}^{r} m_i \mathbf{1}_{K_i}$, where the m_i's are integers and the K_i's are compact subanalytic subsets of $\mathbb{R}^n$. We denote by $\mathrm{CF}(\mathbb{R}^n)$ the group of constructible functions on $\mathbb{R}^n$. We refer to [6, Sec. 8.2, Sec. 9.7] for more details on subanalytic sets and constructible functions.

This is an abstract of a presentation given at CG:YRF 2022. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

[1] In this note, we consider only compactly supported constructible functions.

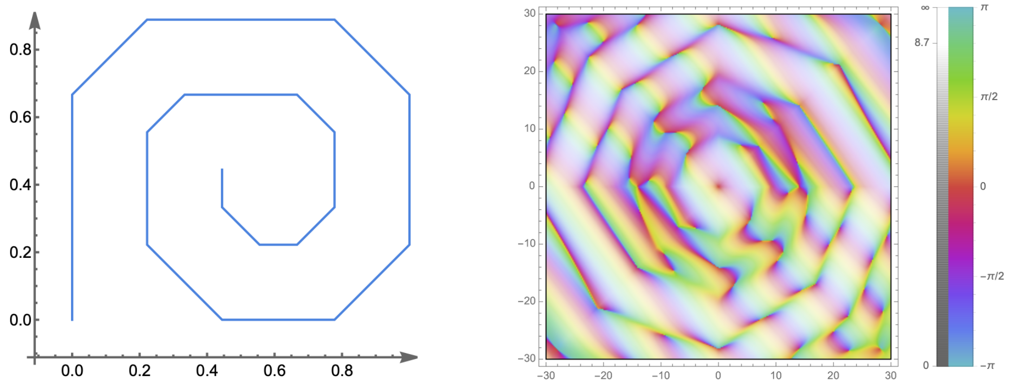


Figure 1 Left: a piecewise-linear closed curve C in $\mathbb{R}^2$. Right: Euler-Fourier transform of $\mathbf{1}_C$.

▶ **Example 2.1.** Any polytope of $\mathbb{R}^n$ — the convex hull of a finite set of points of $\mathbb{R}^n$ — is subanalytic. If the subsets K_i in the decomposition of φ are polytopes, then φ is said *PL-constructible*. We denote by $\mathrm{CF}_{\mathrm{PL}}(\mathbb{R}^n)$ the group of PL-constructible functions on $\mathbb{R}^n$.

▶ **Definition 2.2.** Let $\xi \in \mathbb{R}^n$ and $\varphi = \sum_{i=1}^r m_i \mathbf{1}_{K_i}$ be a constructible function. The *pushforward of φ along ξ* is the constructible function $\xi_*\varphi$ over $\mathbb{R}$ defined for any $t \in \mathbb{R}$ by

$$\xi_*\varphi(t) = \sum_{i=1}^{r} m_i \cdot \chi\left(\xi^{-1}(t) \cap K_i\right),$$

where[2] $\xi^{-1}(t) = \{x \in \mathbb{R}^n\,;\, \langle \xi; x\rangle = t\}$ and χ is the Euler characteristic, that is $\chi(Z) = \sum_{j\in\mathbb{Z}}(-1)^j \dim_{\mathbb{Q}} H^j(Z;\mathbb{Q})$ for any $Z \subseteq \mathbb{R}^n$ compact and subanalytic. See Figure 2.

The fact that this definition does not depend on the decomposition of φ and that $\xi_*\varphi$ is a constructible function on $\mathbb{R}$ is proven by Schapira [9, 10].

▶ **Example 2.3.** If $P \subseteq \mathbb{R}^n$ is a polytope, then $\xi_*\mathbf{1}_P = \mathbf{1}_{[\min_P(\xi),\max_P(\xi)]}$, where $\min_P(\xi) = \min\{\langle\xi;x\rangle\,;\, x \in P\}$ and $\max_P(\xi) = \max\{\langle\xi;x\rangle\,;\, x \in P\}$. In fact, there is a vertex p (resp. q) of P, depending on ξ, such that $\min_P(\xi) = \langle\xi;p\rangle$ (resp. $\max_P(\xi) = \langle\xi;q\rangle$).

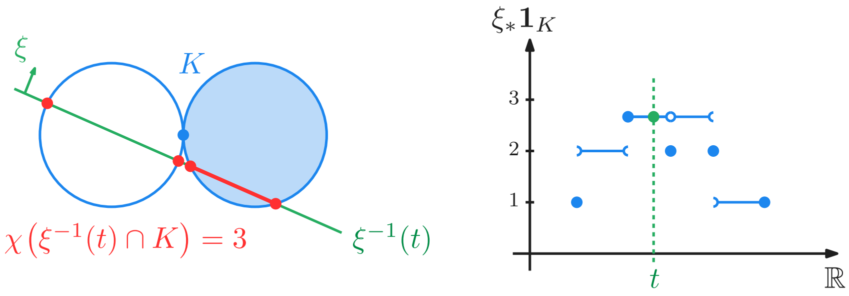


Figure 2 The pushforward along $\xi \in \mathbb{R}^2$ of $\mathbf{1}_K$ for the compact subanalytic $K \subseteq \mathbb{R}^2$.

[2] For any two $x, y \in \mathbb{R}^n$, we denote by $\langle x; y\rangle$ their canonical scalar product.

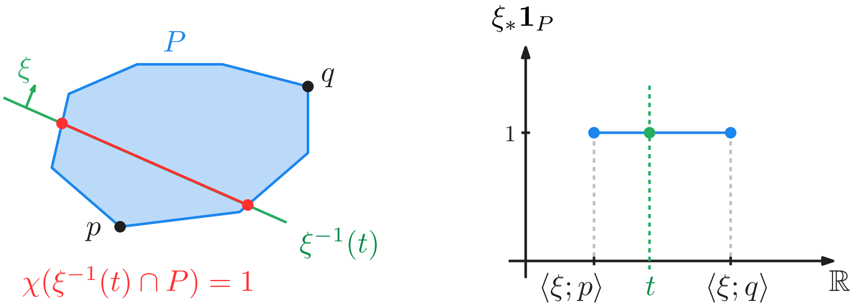

P
ξ
q
p
$\xi^{-1}(t)$
$\chi(\xi^{-1}(t) \cap P) = 1$
$\xi_* \mathbf{1}_P$
1
$\langle \xi ; p \rangle$
t
$\langle \xi ; q \rangle$
$\mathbb{R}$

Figure 3 The pushforward along $\xi \in \mathbb{R}^2$ of $\varphi = \mathbf{1}_P$ for the polytope $P \subseteq \mathbb{R}^2$.

▶ **Definition 2.4.** The *Euler-Fourier transform* of $\varphi \in \mathrm{CF}(\mathbb{R}^n)$ is defined for $\xi \in \mathbb{R}^n$ by:

$$\mathcal{EF}[\varphi](\xi) = \int_{\mathbb{R}} e^{-it} \xi_* \varphi(t) \, \mathrm{d}t.$$

Choosing any kernel $\kappa \in \mathrm{L}^1_{\mathrm{loc}}(\mathbb{R})$ instead of $t \mapsto e^{-it}$ leads to the general definition of hybrid transform, studied in [8]. We now turn to examples. The reader's attention is drawn to the effect of the successive application of topological pushforward and of classical integral.

▶ **Example 2.5.** Denote by $\mathbb{S}_r$ the sphere of radius $r > 0$ in $\mathbb{R}^n$. For any $\xi \in \mathbb{R}^n$,

$$\mathcal{EF}[\mathbf{1}_{\mathbb{S}_r}](\xi) = 2 \cdot (1 + (-1)^n) \cdot \sin(r \|\xi\|).$$

▶ **Example 2.6.** Consider the constructible function $\varphi = \mathbf{1}_S - \mathbf{1}_C$, where $S = [-1/2, 1/2]^2$ and C is the piecewise linear closed curve of $\mathbb{R}^2$ represented by the dotted line in Figure 4b. Since C has zero volume, the (classical) Fourier transforms of $\mathbf{1}_S$ and of $\mathbf{1}_S - \mathbf{1}_C$ are equal. However, their Euler-Fourier transforms differ, as shown in Figure 4.

3 Properties

The Euler-Fourier transform enjoys a regularity result on PL-constructible functions.

▶ **Proposition 3.1.** *Let $\varphi \in \mathrm{CF}_{\mathrm{PL}}(\mathbb{R}^n)$. The function $\mathcal{EF}[\varphi]$ is continuous, bounded and piecewise smooth on $\mathbb{R}^n$.*

The Euler-Fourier transform enjoys several invariance properties. We emphasize here specific ones which are also satisfied by the classical Fourier transform.

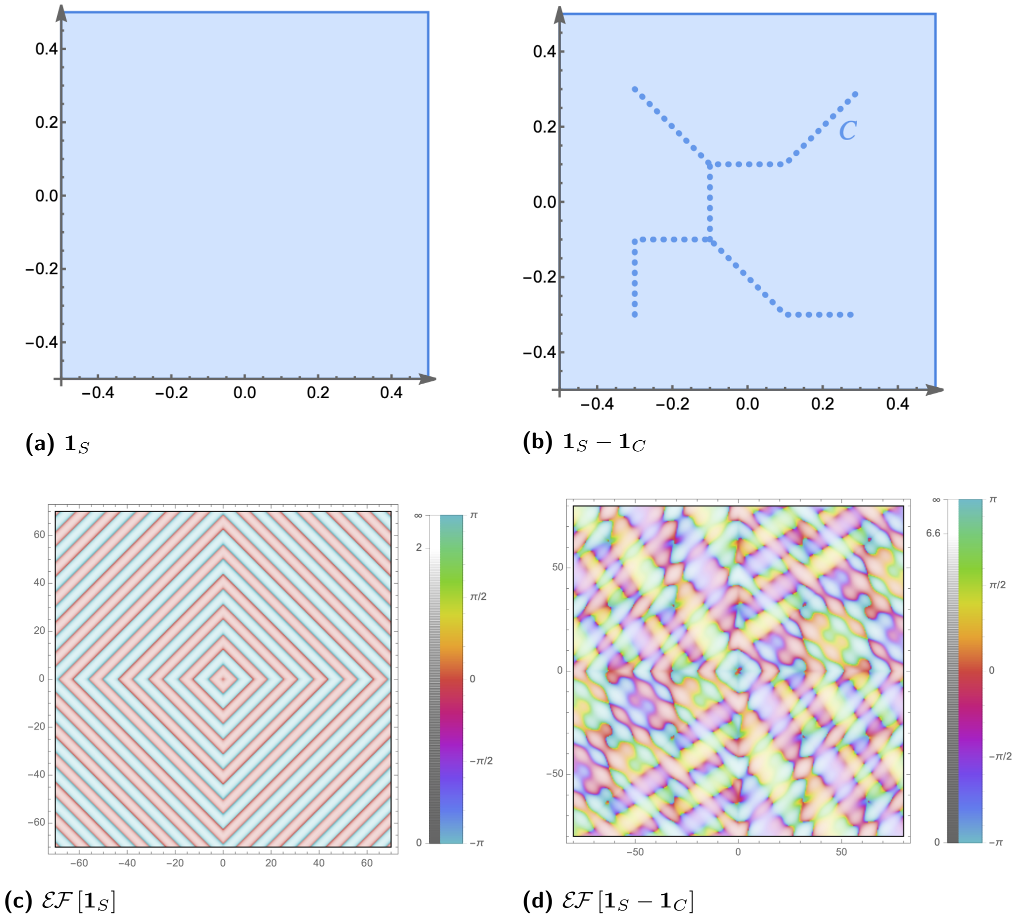
▶ **Proposition 3.2.** *Let $\varphi \in \mathrm{CF}(\mathbb{R}^n)$ and $A \in \mathrm{GL}_n(\mathbb{R})$. Denote by $A_*\varphi$ the constructible function on $\mathbb{R}^n$ given by $A_*\varphi(x) = \varphi(A^{-1}x)$, for any $x \in \mathbb{R}^n$. For any $\xi \in \mathbb{R}^n$, we have:*

$$\mathcal{EF}[A_*\varphi](\xi) = \mathcal{EF}[\varphi]\left({}^t\!A\,\xi\right).$$

▶ **Proposition 3.3.** *Let $\varphi \in \mathrm{CF}(\mathbb{R}^n)$ and $x_0 \in \mathbb{R}^n$. Denote by $\tau_{x_0 *}\varphi$ the constructible function on $\mathbb{R}^n$ given by $\tau_{x_0 *}\varphi(x) = \varphi(x - x_0)$, for any $x \in \mathbb{R}^n$. For any $\xi \in \mathbb{R}^n$, we have:*

$$\mathcal{EF}[\tau_{x_0 *}\varphi](\xi) = e^{-i\langle \xi ; x_0 \rangle} \cdot \mathcal{EF}[\varphi](\xi).$$

These operations are not the only operations available on constructible functions. In [8], we study the compatibility of hybrid transforms with numerous operations.



(a) $\mathbf{1}_S$

(b) $\mathbf{1}_S - \mathbf{1}_C$

(c) $\mathcal{EF}\left[\mathbf{1}_S\right]$

(d) $\mathcal{EF}\left[\mathbf{1}_S - \mathbf{1}_C\right]$

Figure 4 Euler-Fourier transforms of the constructible functions $\mathbf{1}_S$ and $\mathbf{1}_S - \mathbf{1}_C$ in Example 2.6.

4 Computations

Let $\varphi \in \mathrm{CF}_{\mathrm{PL}}(\mathbb{R}^n)$ be written as $\varphi = \sum_{l=1}^r m_l \cdot \mathbf{1}_{P_l}$ where the subsets P_l are polytopes. By $\mathbb{Z}$-linearity of $\mathcal{EF}$ and Example 2.3, we have for any $\xi \in \mathbb{R}^n$,

$$\mathcal{EF}\left[\varphi\right](\xi) = \sum_{l=1}^r m_l \int_{\min_{P_l}(\xi)}^{\max_{P_l}(\xi)} e^{-it}\,\mathrm{d}t = i \sum_{l=1}^r m_l \left(e^{-i\max_{P_l}(\xi)} - e^{-i\min_{P_l}(\xi)}\right). \tag{4.1}$$

The extrema $\min_{P_l}(\xi)$ and $\max_{P_l}(\xi)$ being attained on vertices of P_l, computing the extrema of $\langle \xi; v_l\rangle$ for v_l ranging over the set of vertices of P_l yields the value of $\mathcal{EF}\left[\varphi\right](\xi)$ using (4.1).

Consider now a fixed finite collection of polytopes $\mathcal{P} = \{P_l\}_{l=1}^r$ and denote by $\mathrm{CF}_{\mathcal{P}}(\mathbb{R}^n)$ the set of $\varphi \in \mathrm{CF}(\mathbb{R}^n)$ that can be written as $\varphi = \sum_{l=1}^r m_l \cdot \mathbf{1}_{P_l}$. Precomputing the extrema of ξ on the set of vertices of each polytope of $\mathcal{P}$, the Euler-Fourier transform of any $\varphi \in \mathrm{CF}_{\mathcal{P}}(\mathbb{R}^n)$ is easily computed using (4.1). As an important example, greyscale images of size $n \times m$ can naturally be seen as constructible functions on a fixed cubical complex of $\mathbb{R}^2$. Each value of their transforms can thus be computed in $\mathcal{O}(nm)$ operations.

References

1 Omer Bobrowski and Matthew Strom Borman. Euler integration of gaussian random fields and persistent homology. *Journal of Topology and Analysis*, 4(1):49–70, 2012.

2 Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. *arXiv:1805.09782*, 2018.

3 Robert Ghrist and Michael Robinson. Euler–Bessel and Euler–Fourier transforms. *Inverse Problems*, 27(12), 2011.

4 Dejan Govc and Richard Hepworth. Persistent magnitude. *Journal of Pure and Applied Algebra*, 225(3), 2021.

5 Leo Guibas, Lyle Ramshaw, and Jorge Stolfi. A kinetic framework for computational geometry. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 100–111. IEEE Computer Society, 1983.

6 Masaki Kashiwara and Pierre Schapira. *Sheaves on Manifolds*, volume 292 of *Grundlehren der Mathematischen Wissenschaften.* Springer-Verlag, Berlin, 1990.

7 Tomas Lazano-Perez. Spatial planning: A configuration approach. *IEEE Trans. on Computers*, 100(2):108–120, 1983.

8 Vadim Lebovici. Hybrid transforms of constructible functions. *arXiv:2111.07829*, 2021.

9 Pierre Schapira. Cycles lagrangiens, fonctions constructibles et applications. *Séminaire Équations aux dérivées partielles (Polytechnique) dit aussi "Séminaire Goulaouic-Schwartz"*, 1988-1989.

10 Pierre Schapira. Operations on constructible functions. *Journal of Pure and Applied Algebra*, 72(1):83–93, 1991.

11 Pierre Schapira. Tomography of constructible functions. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 427–435. Springer, 1995.