

Computational Geometry: Young Researchers Forum 2024

— book of abstracts —

This volume contains the abstracts of presentations given at “Computational Geometry: Young Researchers Forum” (CG:YRF), a satellite event of the 40th International Symposium on Computational Geometry, held in Athens, Greece, on June 11-14, 2024.

The CG:YRF program committee consisted of the following people:

- Omrit Filtser, Open University of Israel
- Mayank Goswami, CUNY Queens College, USA
- Irina Kostitsyna, KBR at NASA Ames Research Center, USA
- Erik Krohn, University of Wisconsin, USA
- Yakov Nekrich, Michigan Technological University, USA
- Valentin Polishchuk (chair), Linköping University, Sweden
- Frank Staals, Utrecht University, the Netherlands
- Csaba D. Tóth, California State University Northridge, USA
- Mikael Vejdemo-Johansson, CUNY College of Staten Island, USA
- Kevin Verbeek, TU Eindhoven, the Netherlands

There were 20 abstracts submitted to CG:YRF. Of these, 15 were accepted with revisions after a limited refereeing process. One abstract was withdrawn. 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 — Programme

Tuesday, 12 June 2024, 10:00–10:15, CG:YRF Fast Forward. Chair: Mikael Vejdemo-Johansson

Tuesday, 12 June 2024, 16:00–18:00, CG:YRF Session A. Chair: Erin Wolf Chambers

Labeled Interleaving Distance for Reeb Graphs 1
Fangfei Lan, Salman Parsa and Bei Wang

An Interleaving Distance for Ordered Merge Trees 5
Thijs Beurskens, Tim Ophelders, Bettina Speckmann and Kevin Verbeek

Computing Loss Function to Bound the Interleaving Distance for Mapper Graphs 9
Erin Wolf Chambers, Ishika Ghosh, Elizabeth Munch, Sarah Percival and Bei Wang

Any Graph is a Mapper Graph 13
Enrique G. Alvarado, Robin Belton, Kang-Ju Lee, Sourabh Palande, Sarah Percival and Emilie Purvine

Differential of Generalized Rank Invariant Landscape (D-GRIL) 19
Soham Mukherjee, Shreyas Samaga, Cheng Xin, Steve Oudot and Tamal K. Dey

Spatiotemporal Persistence Landscapes 23
Martina Flammer

Tagged barcodes for the topological analysis of gradient-like vector fields 29
Clemens Bannwart and Claudia Landi

The Walk-Length Filtration for Persistent Homology on Weighted Directed Graphs 33
David Munoz, Elizabeth Munch, Brittany Terese Fasy and Firas A. Khasawneh

Tuesday, 12 June 2024, 16:00–18:00, CG:YRF Session B. Chair: Omrit Filtser

A Symmetry-Aware Vietoris-Rips Algorithm 37
Jordan Matuszewski, Daniel Hope and Mikael Vejdemo-Johansson

On Totally-Concave Polyominoes 41
Gill Barequet, Noga Keren, Johann Peters and Adi Rivkin

The Gate-Cover Problem in Thin Polyominoes 47
Ariel Rosenberg, Esther Arkin, Alon Efrat, Omrit Filtser and Joseph S. B. Mitchell


Counting Crossing-Free Structures in 2-Page-Book Drawings 51
Joachim Orthaber and Javier Tejel

Comparison of Graph Distance Measures 55
Maike Buchin and Lea Thiel

A New Separator Theorem for Unit Disk Graphs 59
Elfarouk Harb, Zhengcheng Huang and Da Wei Zheng

Extending Hausdorff Distances to Asymmetric Geometries 63
Tuyen Pham and Hubert Wagner

Labeled Interleaving Distance for Reeb Graphs

Fangfei Lan ✉ 

Scientific Computing and Imaging Institute, University of Utah, USA

Salman Parsa ✉ 

School of Computing, DePaul University, USA

Bei Wang ✉ 

Scientific Computing and Imaging Institute, University of Utah, USA

Abstract

We define the labeled interleaving distance for labeled Reeb graphs (where all vertices are labeled from a fixed set). We prove that the (ordinary) interleaving distance between Reeb graphs equals the minimum of the labeled interleaving distance over all labelings. We also show that under mild conditions, the labeled interleaving distance is a metric on the isomorphism classes of Reeb graphs.

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

Keywords and phrases Reeb graphs, interleaving distance, topological data analysis

Related Version A full version of the paper is available at <https://arxiv.org/abs/2306.01186>.

Funding This work was partially supported by a grant from the Department of Energy (DOE) DE-SC0021015, and grants from the National Science Foundation (NSF) IIS-1910733 and IIS-2145499.

1 Introduction and Background

Reeb graphs are topological descriptors that capture topological changes of level sets of scalar fields, with many applications in topological data analysis and visualization [1, 6]. Formally, a Reeb graph, denoted $R = (G, f)$, is a finite multi-graph G equipped with a function $f : G \rightarrow \mathbb{R}$ such that the restriction of f on each edge is strictly monotone. It is considered to be a continuous topological space in our setting.

Discriminatory distances for Reeb graphs such as the interleaving distance [2] are often difficult to compute [6]. Recently, Munch and Stefanou [5] introduced a labeled interleaving distance on merge trees with labeled vertices that can be computed in $O(n^2)$ (n being the number of critical points of f). Gasparovic et al. [4] proved that the (ordinary) interleaving distance of merge trees is the minimum of the labeled interleaving distance over all labelings.

In this work, we define a labeled interleaving distance for Reeb graphs, prove that it is a metric on the isomorphism classes of Reeb graphs (under certain conditions on the labelings), and that the (ordinary) interleaving distance is the minimum of the labeled interleaving distance. In particular, our distance can be computed in $O(n^2)$ time for contour trees.

The ordinary interleaving distance is defined using the ε -smoothed Reeb graph [3] R^ε , i.e., the Reeb graph of $G \times [-\varepsilon, \varepsilon]$ together with a function inherited from $\pi : G \times [-\varepsilon, \varepsilon] \rightarrow \mathbb{R}$. See Fig. 1 for an example in which R^ε is an ε -smoothed Reeb graph of R .

► **Definition 1.** Let R_1 and R_2 be Reeb graphs. An ε -interleaving between R_1 and R_2 is given

This is an abstract of a presentation given at CG:YRF 2024. 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 Labeled Interleaving Distance for Reeb Graphs

by two morphisms, $\phi : R_1 \rightarrow R_2^\varepsilon$, $\psi : R_2 \rightarrow R_1^\varepsilon$, such that the following diagram commutes,

$$\begin{array}{ccccc} R_1 & \xrightarrow{\eta_1} & R_1^\varepsilon & \xrightarrow{\eta_1^\varepsilon} & R_1^{2\varepsilon} \\ & \searrow \phi & & \searrow \phi^\varepsilon & \\ & & & & \\ & \swarrow \psi & & \swarrow \psi^\varepsilon & \\ R_2 & \xrightarrow{\eta_2} & R_2^\varepsilon & \xrightarrow{\eta_2^\varepsilon} & R_2^{2\varepsilon}. \end{array}$$

The interleaving distance is defined to be

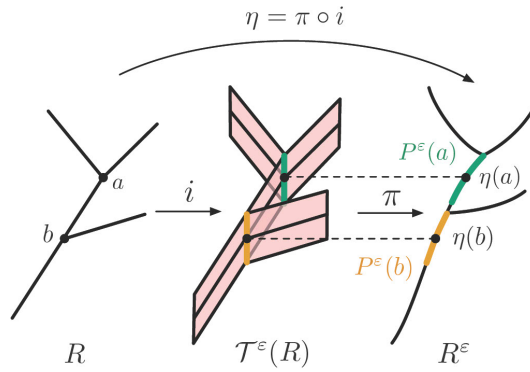
$$d_I(R_1, R_2) := \inf\{\varepsilon \geq 0 \mid \text{there exists an } \varepsilon\text{-interleaving between } R_1 \text{ and } R_2\}.$$

For a Reeb graph R and its node set V , we define a correspondence $s : V \rightarrow R^\varepsilon$ on all nodes in R . Intuitively, s maps a node in R to the point in R^ε where it arrives after moving up (for a split node) or down (for a join node) by ε .

2 Labeled Reeb Graphs and Their Distance

Given a finite label set $L = [N] := \{1, \dots, N\}$ and a Reeb graph $R = (G, f)$ with the node set V , a *labeling* of R is a function $\lambda : L \rightarrow V$. We call the triple $R_\lambda = (G, f, \lambda)$ an *L-labeled Reeb graph*. R_λ is *fully-labeled* if λ is surjective. λ is not necessarily injective. A *morphism* between labeled Reeb graphs is defined to be the morphism of the underlying unlabeled Reeb graphs. We want to define a *labeled ε -interleaving* between two labeled Reeb graphs by adding label-preserving properties to the *ordinary ε -interleaving* defined by the commutative diagram in Def. 1. To do so, we first introduce an ε -path-neighborhood.

► **Definition 2.** Let $a \in V(R)$ be a node in the Reeb graph $R = (G, f)$. The ε -path-neighborhood of a , denoted $P^\varepsilon(a)$, is $\pi(\{a\} \times [-\varepsilon, \varepsilon]) \subset R^\varepsilon$ in the ε -smoothed Reeb graph R^ε . $\pi : G \times [-\varepsilon, \varepsilon] \rightarrow R^\varepsilon$ is the quotient map. $\mathcal{T}^\varepsilon(R)$ is $G \times [-\varepsilon, \varepsilon]$ with the product topology.



■ **Figure 1** The ε -path-neighborhoods of a and b are highlighted in green and yellow in R^ε .

As illustrated in Fig. 1, we observe that for any point $x \in R$ and any $\varepsilon \geq 0$, $P^\varepsilon(x)$ is a monotonic path in R^ε such that $f^\varepsilon(P^\varepsilon(x)) = [f(x) - \varepsilon, f(x) + \varepsilon]$.

► **Definition 3.** Let $R_{1,\lambda_1} = (G_1, f_1, \lambda_1)$ and $R_{2,\lambda_2} = (G_2, f_2, \lambda_2)$ be two L -labeled Reeb graphs, and let $\varepsilon \geq 0$. We say a pair of morphisms $\phi : R_1 \rightarrow R_2^\varepsilon$ and $\psi : R_2 \rightarrow R_1^\varepsilon$ define a labeled ε -interleaving between R_{1,λ_1} and R_{2,λ_2} if the following hold:

1. ϕ and ψ define an ε -interleaving between R_1 and R_2 .

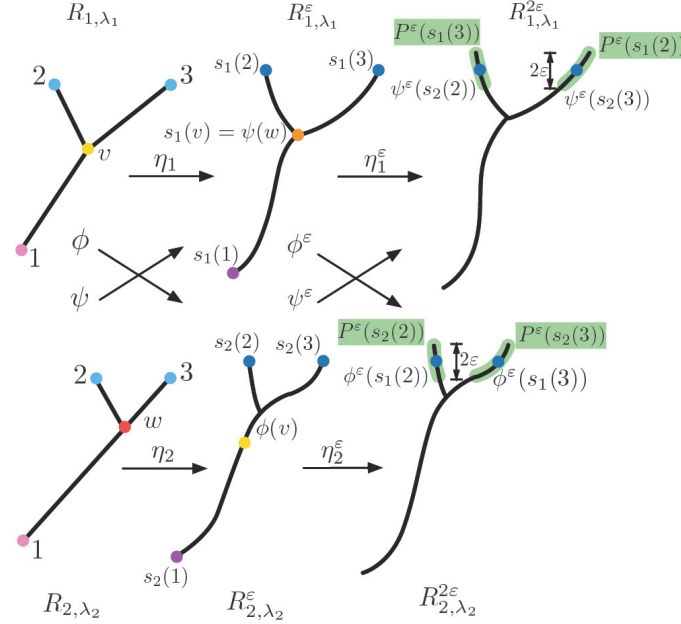
2. For each $\ell \in L$, we have the following label-preserving properties,

$$\begin{aligned} \phi^\varepsilon(s(\lambda_1(\ell))) &\in P^\varepsilon(s(\lambda_2(\ell))), \\ \psi^\varepsilon(s(\lambda_2(\ell))) &\in P^\varepsilon(s(\lambda_1(\ell))). \end{aligned} \quad (1)$$

In the above formulae, $P^\varepsilon(s(\lambda_1(\ell))) \subset R_1^{2\varepsilon}$ and $P^\varepsilon(s(\lambda_2(\ell))) \subset R_2^{2\varepsilon}$, since $s(\lambda_1(\ell)) \in R_1^\varepsilon$ and $s(\lambda_2(\ell)) \in R_2^\varepsilon$. The labeled interleaving distance $d_I^L(R_{1,\lambda_1}, R_{2,\lambda_2})$ is defined as

$$\inf\{\varepsilon \geq 0 \mid \text{there exists a labeled } \varepsilon\text{-interleaving between } R_{1,\lambda_1} \text{ and } R_{2,\lambda_2}\}. \quad (2)$$

Fig. 2 illustrates an example for Def. 3 where R_{1,λ_1} and R_{2,λ_2} are labeled ε -interleaving. We simplify the notations for node labels. We label nodes $\lambda_1(2)$ with 2, and $s(\lambda_1(2))$ with $s_1(2)$. Nodes with the same function value are assigned the same color. The highlighted areas demonstrate the label-preserving properties in Eqn. (1). It follows from Def. 3 that for all R_1, R_2, λ_1 , and λ_2 , $d_I^L(R_{1,\lambda_1}, R_{2,\lambda_2}) \geq d_I(R_1, R_2)$. If the label set L is empty, the labeled interleaving distance equals the unlabeled one.



■ **Figure 2** R_{1,λ_1} and R_{2,λ_2} are labeled ε -interleaving.

It turns out that the interleaving distance depends only on the topological features significant enough with respect to ε . Therefore, we introduce the ε -essential and ε -inessential nodes. Intuitively, we can disregard nodes connecting to small loops and short edges that do not affect the labeled distance. We call a labeling an ε -essential labeling if every ε -essential node is labeled. We consider all possible ε -essential labelings for the following theorem.

► **Theorem 4.** Let R_1, R_2 be Reeb graphs and set $\varepsilon = d_I(R_1, R_2)$. There exist a label set L and ε -essential labelings λ_1 and λ_2 such that for L -labeled Reeb graphs R_{1,λ_1} and R_{2,λ_2} ,

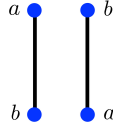
$$d_I(R_1, R_2) = d_I^L(R_{1,\lambda_1}, R_{2,\lambda_2}).$$

3 Metric Properties

The labeled interleaving distance can be infinite when two nodes with the same label are originally apart and move in opposite directions when smoothed, see Fig. 3 for an example.

4 Labeled Interleaving Distance for Reeb Graphs

For a label set L , we denote the set of all L -labeled Reeb graphs by \mathcal{R}^L . We say two labeled Reeb graphs $R_{1,\lambda_1}, R_{2,\lambda_2}$ are *consistently labeled* if for each label $\ell \in L$, the nodes $v_1 = \lambda_1(\ell)$ and $v_2 = \lambda_2(\ell)$ move in the same direction when smoothed. A set of labeled Reeb graphs $\mathcal{R} \subset \mathcal{R}^L$ is *consistently labeled* if each pair of labeled Reeb graphs in \mathcal{R} are consistently labeled. Under this condition, the labeled interleaving distance becomes an extended metric.



■ **Figure 3** A pair of inconsistently labeled Reeb graphs

► **Theorem 5.** *Let \mathcal{R}_c^L be a maximal set of isomorphism classes of consistently L -labeled Reeb graphs. The labeled interleaving distance is an extended metric on \mathcal{R}_c^L .*

References

- 1 Vincent Barra and Silvia Biasotti. 3D shape retrieval using kernels on extended Reeb graphs. *Pattern Recognition*, 46(11):2985–2999, 2013. doi:10.1016/j.patcog.2013.03.019.
- 2 Erin Wolf Chambers, Elizabeth Munch, and Tim Ophelders. A family of metrics from the truncated smoothing of Reeb graphs. *37th International Symposium on Computational Geometry (SoCG 2021)*, 189:22:1–22:17, 2021. doi:10.4230/LIPIcs.SoCG.2021.22.
- 3 Vin De Silva, Elizabeth Munch, and Amit Patel. Categorified Reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, 2016. doi:10.1007/s00454-016-9763-9.
- 4 Ellen Gasparovic, Elizabeth Munch, Steve Oudot, Katharine Turner, Bei Wang, and Yusu Wang. Intrinsic interleaving distance for merge trees. arXiv preprint arXiv:1908.00063, 2022.
- 5 Elizabeth Munch and Anastasios Stefanou. *The ℓ^∞ -cophenetic metric for phylogenetic trees as an interleaving distance*, volume 17 of *Association for Women in Mathematics Series*, pages 109–127. Springer International Publishing, Cham, 2019. doi:10.1007/978-3-030-11566-1_5.
- 6 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 (CGF)*, 40(3):599–633, 2021. doi:10.1111/cgf.14331.

An Interleaving Distance for Ordered Merge Trees

Thijs Beurskens ✉

Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Tim Ophelders ✉ 

Department of Information and Computing Science, Utrecht University, The Netherlands

Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Bettina Speckmann ✉ 

Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Kevin Verbeek ✉ 

Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

Merge trees are a common topological descriptor for data with a hierarchical component. The interleaving distance, in turn, is a common distance measure for comparing merge trees. In this abstract, we introduce a form of ordered merge trees and extend the interleaving distance to a measure that preserves orders. Exploiting the additional structure of ordered merge trees, we then describe an $\mathcal{O}(n^2)$ time algorithm that computes a 2-approximation of this new distance with an additive term G that captures the maximum height differences of leaves of the input merge trees.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Monotone Interleaving Distance, Merge Tree, Approximation

Related Version A full version of the paper is available at arxiv.org/abs/2312.11113.

Funding *Thijs Beurskens*: Supported by the Dutch Research Council (NWO) under project number OCENW.M20.089.

Tim Ophelders: Supported by the Dutch Research Council (NWO) under project number VI.Veni.212.260

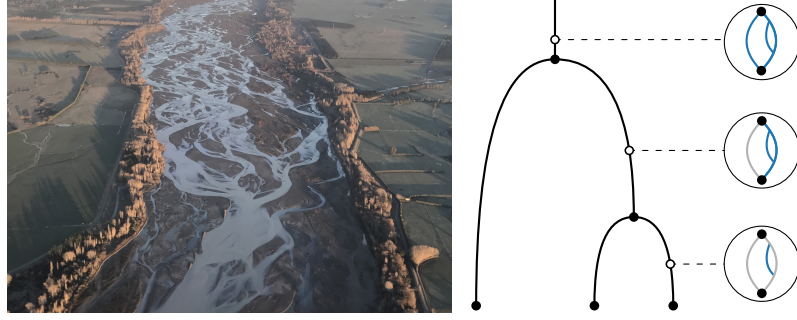
1 Introduction

Merge trees [6] are a common tool in the field of topological data analysis. They are usually defined on terrains, where they capture connected components of sublevel sets. More generally, a merge tree represents some hierarchical structure in the data. Our work is motivated by the study of *braided rivers*: multi-channel river systems, known to evolve rapidly [3, 5]. We model a river network [4] as a hierarchy of *braids*, and use a merge tree to represent this hierarchy: each leaf represents a single channel in the network, and each internal vertex represents two braids merging (see Figure 1).

It is our goal to analyse the evolution of braided rivers over time. The standard way to compare two merge trees is the interleaving distance [2, 6, 9]. However, the interleaving distance has two main drawbacks. Firstly, the standard interleaving distance is unable to capture any intrinsic order, e.g. from bank to bank in braided rivers, that might be present in the data. Secondly, there is no known efficient algorithm to compute even an approximation of the interleaving distance.¹ To tackle both issues, we impose our merge trees with an

This is an abstract of a presentation given at CG:YRF 2024. 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 Agarwal et al. [1] actually prove that approximating the Gromov-Hausdorff distance with a factor better than 3 is NP-hard. As many have observed, this proof also applies to the interleaving distance.



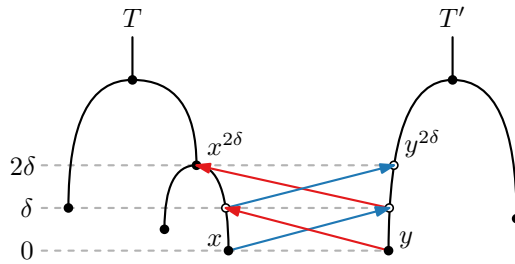
■ **Figure 1** (Left) a braided river (Greg O’Beirne [8]). (Right) a river network as a merge tree.

additional structure, and introduce the *monotone interleaving distance*. Omitted proofs and details can be found in the full version on arXiv.

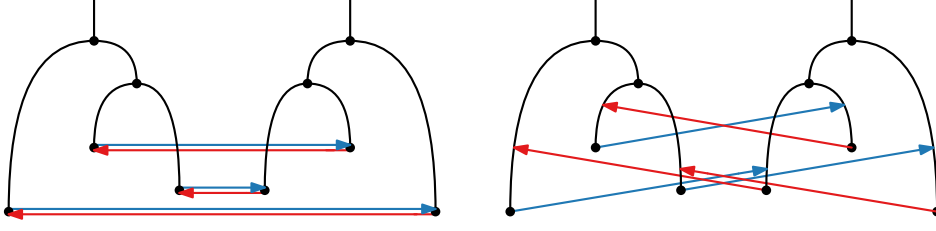
2 Merge Trees and Interleavings

A *merge tree* is a pair (T, f) , where T is a rooted tree and $f: T \rightarrow \mathbb{R} \cup \{\infty\}$ is a continuous height function that is increasing towards the root, with $f(v) = \infty$ if and only if v is the root. Here, f is defined not only on the vertices of T , but also on the interior of the edges. Consider two merge trees (T, f) and (T', f') and a fixed value $\delta \geq 0$. Morozov et al. [6] define a δ -*interleaving* as a continuous mapping α from T to T' that sends points exactly δ upwards, together with a similar map β from T' to T , such that both compositions of α and β send any point to its unique ancestor 2δ higher. The *interleaving distance* d_{\uparrow} is defined as the smallest δ such that there exists a δ -interleaving. See Figure 2 for an example.

The maps α and β are both δ -*shift maps*, i.e. continuous maps that send points exactly δ higher. Touli and Wang [9] give an alternative definition of the interleaving distance in terms of a single δ -shift map with additional requirements. They call this map a δ -*good map*. Gasparovich et al. [2] give another alternative definition in terms of *labelled merge trees* [7]. For $n \geq 0$, we denote $[n] := \{1, \dots, n\}$. A labelled merge tree is a triple (T, f, π) , where (T, f) is a merge tree and $\pi: [n] \rightarrow T$ is a *label-map* that is surjective on the leaves of T . A labelled merge tree naturally induces a matrix, defined by $M(T, f, \pi)_{i,j} = f(\text{lca}(\pi(i), \pi(j)))$. Here, $\text{lca}(\cdot, \cdot)$ is the lowest common ancestor of a pair of points. For a matrix M , we consider the ℓ^∞ -norm, defined as $\|M\|_\infty := \max_{i,j} |M_{i,j}|$. Lastly, for two *unlabelled* merge trees (T, f) and (T', f') , a δ -*labelling* is defined as a pair (π, π') such that $\|M(T, f, \pi) - M(T', f', \pi')\|_\infty = \delta$.



■ **Figure 2** Two merge trees and part of a δ -interleaving.



■ **Figure 3** Parts of an optimal regular (left) and an optimal monotone (right) interleaving.

Imposing Orders. For a point $x \in T$ with $f(x) \leq h$, we denote by $x|_h$ the unique ancestor of x at height h . An ordered merge tree $(T, f, (\leq_h)_{h \geq 0})$ is a merge tree (T, f) equipped with a set of total orders on the level sets of T , such that it is *consistent*. Formally, for two heights $h_1 \leq h_2$ and two points x_1, x_2 in $f^{-1}(h_1)$, we require that $x_1 \leq_{h_1} x_2$ implies $x_1|_{h_2} \leq_{h_2} x_2|_{h_2}$. We now define an order-preserving distance measure for ordered merge trees $(T, f, (\leq_h)_{h \geq 0})$ and $(T', f', (\leq'_h)_{h \geq 0})$. Specifically, we say a δ -shift map $\alpha: T \rightarrow T'$ is *monotone* if for all height values $h \geq 0$ and for any two points $x_1, x_2 \in f^{-1}(h)$ it holds that $x_1 \leq_h x_2$ implies $\alpha(x_1) \leq'_{h+\delta} \alpha(x_2)$. A *monotone δ -interleaving* is a δ -interleaving (α, β) such that the maps α and β are both monotone (see Figure 3). We define the monotone interleaving distance d_{MI} as the smallest δ for which there exists a monotone δ -interleaving. Similar to the regular setting, we also study the two alternative definitions: the *monotone δ -good interleaving distance* d_{MI}^{G} and the *monotone label interleaving distance* d_{MI}^{L} . Our main result is the following.

► **Theorem 1.** *The distances d_{MI} , d_{MI}^{G} and d_{MI}^{L} are equal.*

Monotone leaf-label interleaving distance. If we restrict a label map to map *only* to the leaves of T , we obtain a *leaf-label map*. A δ -*leaf-labelling* is a pair of leaf-label maps that is also a δ -labelling. The *leaf-label interleaving distance* d_{L}^{L} , in turn, is defined as the smallest δ for which there exists a δ -leaf labelling. We can show that this distance is an approximation of the interleaving distance, in both the regular and monotone setting.

► **Theorem 2.** *The (monotone) leaf-label interleaving distance between two (ordered) merge trees T and T' is bounded by $2\delta + G$, where $\delta = d_{\text{MI}}(T, T')$ and G is maximum height difference of any pair of leaves in T and T' .*


3 Computation

For a monotone δ -leaf-labelling (π, π') , we can show that the value δ lies on the diagonal or upper-diagonal of the matrix $\mathcal{M} = |M(T, f, \pi) - M(T', f', \pi')|$. Exploiting this property, we can use a dynamic program to compute the monotone leaf-labelled interleaving distance between two ordered merge trees T and T' . In particular, we maintain the monotone leaf-label interleaving distance between the subtrees T_i and T'_j , which are the subtrees of T and T' consisting of the first i and j leaves respectively. By the above property, in combination with the imposed order on the trees, it suffices to consider only few options at each iteration of the dynamic program. The dynamic program runs in $\mathcal{O}(n^2)$, where n is the combined number of leaves of both trees. From Theorem 2, it follows that the algorithm is a 2-approximation of the monotone interleaving distance, with an additive term capturing the height difference between any two leaves.


References

- 1 P.K. Agarwal, K. Fox, A. Nath, A. Sidiropoulos, and Y. Wang. Computing the Gromov-Hausdorff distance for metric trees. *ACM Transactions on Algorithms*, 14(2):1–20, 2018. doi:10.1145/3185466.
- 2 E. Gasparovich, E. Munch, S. Oudot, K. Turner, B. Wang, and Y. Wang. Intrinsic interleaving distance for merge trees. [arXiv:1908.00063](https://arxiv.org/abs/1908.00063).
- 3 A.D. Howard, M.E. Keetch, and C.L. Vincent. Topological and geometrical properties of braided streams. *Water Resources Research*, 6(6):1674–1688, 1970. doi:10.1029/WR006i006p01674.
- 4 M.G. Kleinhans, M. van Kreveld, T. Ophelders, W. Sonke, B. Speckmann, and K. Verbeek. Computing representative networks for braided rivers. *Journal of Computational Geometry*, 10(1):423–443, 2019. doi:10.20382/jocg.v10i1a14.
- 5 W.A. Marra, M.G. Kleinhans, and E.A. Addink. Network concepts to describe channel importance and change in multichannel systems: test results for the jamuna river, bangladesh. *Earth Surface Processes and Landforms*, 39(6):766–778, 2014. doi:10.1002/esp.3482.
- 6 D. Morozov, K. Beketayev, and G. Weber. Interleaving distance between merge trees. Manuscript, 2013.
- 7 E. Munch and A. Stefanou. The ℓ^∞ -Cophenetic metric for phylogenetic trees as an interleaving distance. *Research in Data Science*, 17:109–127, 2019. doi:10.1007/978-3-030-11566-1_5.
- 8 G. O’Beirne. Waimakariri River (New Zealand), 2017. Image is slightly cropped; accessed on 30/11/2023. Photo is licensed under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>). URL: <https://commons.wikimedia.org/w/index.php?curid=61302377>.
- 9 E.F. Touli and Y. Wang. FPT-algorithms for computing the Gromov-Hausdorff and interleaving distances between trees. *Journal of Computational Geometry*, 13:89–124, 2022. doi:10.20382/jocg.v13i1a4.

Computing a Loss Function to Bound the Interleaving Distance for Mapper Graphs

Erin Wolf Chambers ✉ 

St. Louis University, USA

Ishika Ghosh ✉ 

Michigan State University, USA

Elizabeth Munch ✉ 

Michigan State University, USA

Sarah Percival ✉ 

Michigan State University, USA

Bei Wang ✉ 

University of Utah, USA

Abstract

Mapper graphs preserve the connected components of the inverse image function $f : \mathbb{X} \rightarrow \mathbb{R}$ over any given cover. Inspired by the interleaving distance for Reeb graphs, (Chambers et al. 2024) extends this notion of distance to discretized mapper graphs. The distance is upper-bounded using a loss function. Unlike the NP-hard interleaving distance computation for Reeb graphs, the algorithm of the loss function has polynomial complexity. In this paper, we implement the categorical framework of mapper graphs and compute the loss function to bound the interleaving distance.

2012 ACM Subject Classification Mathematics of computing \rightarrow Algebraic topology; Theory of computation \rightarrow Computational geometry

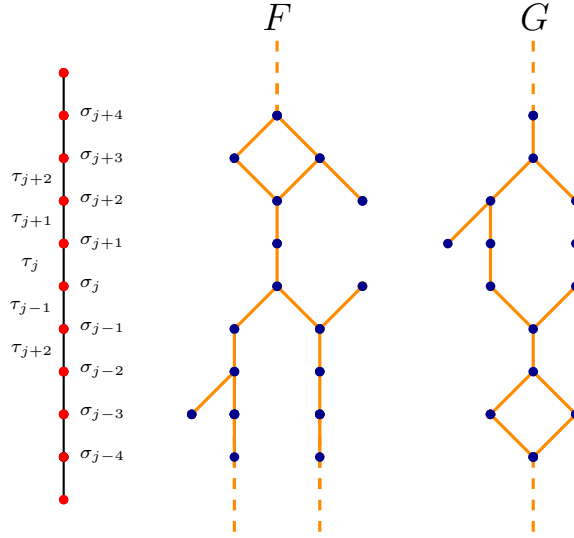
Keywords and phrases Mapper graphs, geometric graphs, interleaving distance

1 Introduction

Developing efficient and computable metrics to compare graphical representations of data is crucial for data analysis. Computationally, topological descriptors of discretized underlying space are essential, as such input data is common. Often, these datasets are equipped with a function $f : \mathbb{X} \rightarrow \mathbb{R}$. Here, we direct our attention to mapper graphs; see Fig. 1 for an example. These graphical data structures keep track of the relationship between connected components of the inverse image of elements of a particular choice of cover. Such mapper graphs can be compared using a variant of the interleaving distance [2, 3, 7]. However, the computation of the interleaving distance is NP-hard in general [1, 2]. Formally, we encode our mapper inputs as functors (see e.g. [5, 6]) of the form $F : \mathbf{Open}(K) \rightarrow \mathbf{Set}$ for a space K encoding the cover information. In [3], K is defined for a chosen $\delta > 0$ as a cubical complex over the bounding interval $[-B, B]$ with diameter δ .

The result is that the open sets of K are intervals of the form $(i\delta, j\delta)$ for $i, j \in \{-L, \dots, L\}$ where $L \cdot \delta = B$. In [3], a 1-thickening on these intervals is introduced, where the thickening $(i\delta, j\delta)^n$ is the interval $((i - n)\delta, (j + n)\delta)$. This can be pre-composed with the functor F to result in an n -thickened functor $F^n : \mathbf{Open}(K) \rightarrow \mathbf{Set}$ given by $F^n(U) = F(U^n)$. This in turn defines an interleaving distance d_I [4, 7] as follows. An interleaving is a pair of natural transformations $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$ which must satisfy certain commutativity

This is an abstract of a presentation given at CG:YRF 2024. 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** Two input mapper graphs F and G . Discretization on the left.

properties. We will denote the four diagrams required to commute by $\nabla_{\varphi}(U, V)$, $\nabla_{\psi}(U, V)$, $\nabla_{\varphi, \psi}(U)$, and $\Delta_{\varphi, \psi}(U)$; see [3] for details. Then the interleaving distance is the smallest n for which such interleaving exists, otherwise the distance is set to infinity [3].

Chambers et al. [3] defined a loss function for structures that have the format of a natural transformation without being provided the commutativity assumptions. They call a collection of maps $\varphi_U : F(U) \rightarrow G^n(U)$ and $\psi_U : G(U) \rightarrow F^n(U)$ an n -assignment; noting that if φ and ψ satisfy the commutativity properties it would constitute an interleaving. Then the loss function $L_B(\varphi, \psi)$ is defined in a way that results in finding the minimum k such that φ and ψ can be turned into an $(n + k)$ -interleaving. When storing the functor information in a graph, the result is that one must check whether two representatives under a particular map are in the same connected component of a slice of the graph; see Fig. 2 for these diagrams.

► **Theorem 1** (Chambers et al. [3]). *For an n -assignment, $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$,*

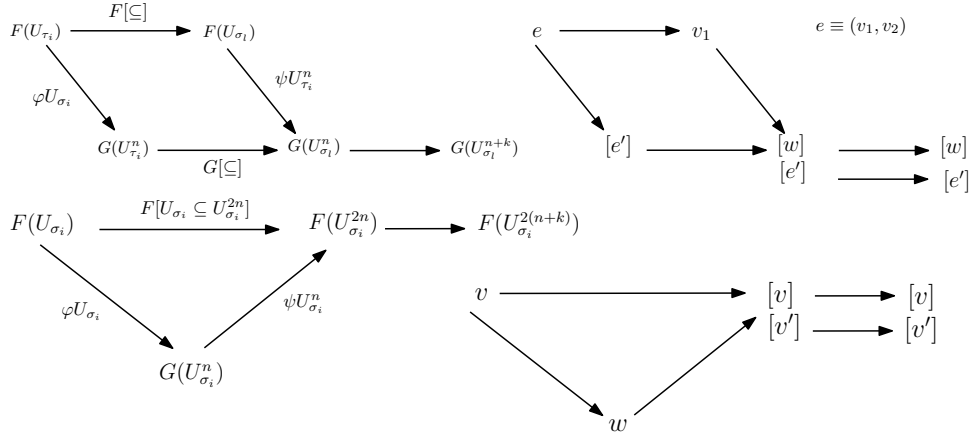
$$d_I(F, G) \leq n + L_B(\varphi, \psi).$$

In this work, we provide additional details of the algorithmic setup for computing this loss function on graph representations of the functor data. See [3] for additional details.

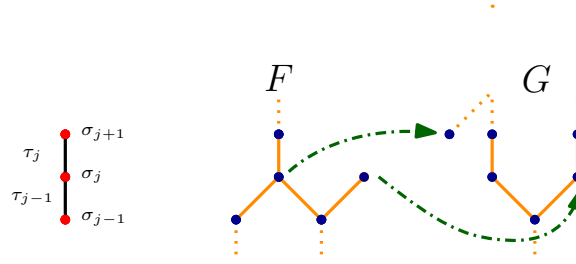
2 Algorithm and Computation

We set up the mapper graph data structure first. Our input is a pair of functors $F, G : \mathbf{Open}(K) \rightarrow \mathbf{Set}$ and an n -assignment φ, ψ . Here K (i.e., the discretization of $[-L\delta, L\delta] \subset \mathbb{R}$) consists of vertices $\sigma_i = i\delta$ for $-L \leq i \leq L$. Additionally, we have edges $\tau_j = (\sigma_j, \sigma_{j+1})$ for $-L \leq j < L$. We write a basis for $\mathbf{Open}(K)$ by defining the collection of intervals $U_{\sigma_i} = ((i-1)\delta, (i+1)\delta)$ and $U_{\tau_i} = (i\delta, (i+1)\delta)$ for all i . The vertex set of the graph representation of the functor is given by $V = \coprod_i F(U_{\sigma_i})$. The edge set is given by $E = \coprod_i F(U_{\tau_i})$ and are attached to the vertices using the functor. See Fig. 1 for an example and [3] for details.

We implement this structure in Python using the *NetworkX* package. We build a custom *MapperGraph* class to encode the functors F and G which constructed as graphs (V_F, E_F)



■ **Figure 2** Example diagrams that must be checked for commutativity to determine the loss function. At right are the representatives from the data structures which must be checked for being in the same connected component of the same slice of the representative graph. See [3] for details.

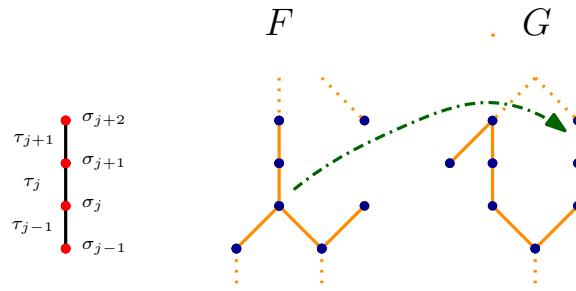


■ **Figure 3** Map the vertices of F at height σ_j with $n = 1$. The G slice contains vertices with heights $\sigma_{j-1}, \sigma_j, \sigma_{j+1}$. Vertices in same connected component in F end up in different components.

and (V_G, E_G) . We also store the height information for each vertex as node attributes. The *MapperGraph* class also contains some useful functions for visualization and retrieval of data.

The n -assignments φ, ψ are encoded as vertex and edge maps; see [3] for details. To define φ (ψ is similar), for height i of each vertex (or height of lower vertex for an edge) in F , we only look at the n -thickening of G at that height. In other words, we define a *slice* of the functor which only includes the vertices and edges within height $[i - n, i + n]$. Each element in F gets randomly paired with an element in the corresponding slice of G . The resulting map is stored as a dictionary, with (object, image) as key-value pairs. Figure 3 and 4 illustrate some examples.

Given two mapper graphs F, G , and assignments φ, ψ , we compute the loss separately for $L_{\nabla}^{U_{\tau}, U_{\sigma}}$ (or, $L_{\nabla}^{U_{\tau}, U_{\sigma}}$) and $L_{\nabla}^{U_{\sigma}}$ (or, $L_{\Delta}^{U_{\sigma}}$). Fix a k for each step with binary search on $[0, \dots, 2L]$, where $[-L, L]$ is the bounding box of the functors. For each k , we verify if $L_B(\varphi, \psi) \leq k$. We travel across the diagrams and note if the resulting edges or vertices are in same connected component. We show an example of two types of diagrams in Fig. 2. Loss is the smallest k for which all diagrams commute. If no such k exists, then the loss is deemed infinite.



■ **Figure 4** Map edges of F with lower vertex-height σ_j with $n = 1$. The G slice contains edges with lower vertex-heights $\sigma_{j-1}, \sigma_j, \sigma_{j+1}$. Notice how slicing varies from vertex mapping.

3 Discussion

Now that we have set up the data structure to encode 1-dimensional mapper graphs, we are focusing on optimizing the loss function. Given two mapper graph functors F, G and an initial n -assignment φ, ψ , our goal is to perturb the assignments cleverly such that the loss function is minimized. In future work, we will deploy the Metropolis–Hastings algorithm over the space of n -assignments to optimize and improve the bound. Further, our goal is to extend this implementation to higher dimensional mapper graphs; i.e. when the input data is of the form $f : \mathbb{X} \rightarrow \mathbb{R}^d$.

References

- 1 Håvard Bakke Bjerkevik and Magnus Bakke Botnan. Computational Complexity of the Interleaving Distance. 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 13:1–13:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.SoCG.2018.13>, doi:10.4230/LIPIcs.SoCG.2018.13.
- 2 Peter Bubenik, Vin De Silva, and Jonathan Scott. Metrics for generalized persistence modules. *Foundations of Computational Mathematics*, 15:1501–1531, 2015.
- 3 Erin W Chambers, Elizabeth Munch, Sarah Percival, and Bei Wang. Bounding the interleaving distance for geometric graphs with a loss function. *arXiv preprint arXiv:2307.15130*, 2023.
- 4 Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 237–246, 2009.
- 5 Justin Curry. *Sheaves, Cosheaves and Applications*. PhD thesis, University of Pennsylvania, 2014. arXiv:1303.3255.
- 6 Emily Riehl. *Category theory in context*. Courier Dover Publications, 2017.
- 7 V De Silva, E Munch, and A Stefanou. THEORY OF INTERLEAVINGS ON CATEGORIES WITH A FLOW. *Theory and Applications of Categories*, 33(21):583–607, 2018. URL: <http://www.tac.mta.ca/tac/volumes/33/21/33-21.pdf>, arXiv:1706.04095.

Any Graph is a Mapper Graph

Enrique G. Alvarado ✉ 🏠

Department of Mathematics, University of California, Davis, USA

Robin Belton¹ ✉ 🏠 

Department of Mathematical Sciences, Smith College, USA

Kang-Ju Lee ✉

Research Institute of Mathematics, Seoul National University, South Korea

Sourabh Palande ✉

Department of Computational Mathematics, Science, and Engineering, Michigan State University, USA

Sarah Percival ✉ 🏠

Department of Computational Mathematics, Science, and Engineering, Michigan State University, USA

Emilie Purvine ✉

Mathematics of Data Science, Pacific Northwest National Laboratory, USA

Abstract

Mapper graphs are a popular visualization tool in topological data analysis. We investigate the following type of inverse problem in the context of Mapper graphs: Given a graph G and data X , does there exist a set of Mapper parameters such that the Mapper graph of X is isomorphic to G ? We provide constructions showing that the answer to this question is yes under mild assumptions. This work shows that one can engineer the input Mapper parameters to get a desired graph. Nevertheless, the constructions we provide are contrived and answering the inverse problem using parameter choices that a data practitioner may use needs further investigation.

2012 ACM Subject Classification Mathematics of computing → Graph Theory; Mathematics of computing → Algebraic topology

Keywords and phrases Topological data analysis, Mapper graphs, clustering

Funding *Kang-Ju Lee*: National Research Foundation of Korea (NRF) Grants funded by the Korean Government (MSIP) (No.2021R1C1C2014185)

Acknowledgements This research is a product of one of the working groups at the American Mathematical Society (AMS) Mathematical Research Community: *Models and Methods for Sparse (Hyper)Network Science* in June 2022. The workshop and follow-up collaboration was supported by the National Science Foundation under grant number DMS 1916439. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the American Mathematical Society. The authors would like to thank Emily Fischer and Sarah Tymochko for helpful input during early conversations of this work.

1 Introduction

Topological data analysis (TDA) uses methods from topology to study the underlying structure and “shape” of a dataset. We refer the reader to [3] for an overview of this area.

This is an abstract of a presentation given at CG:YRF 2024. 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

This paper focuses on *Mapper* introduced in [11], a popular visualization tool in TDA that constructs a network representation of a dataset. Mapper has been used in many applications including analyzing breast cancer data [9] and identifying diabetes subgroups [7].

To apply the Mapper algorithm, the user needs to determine the following parameters: choosing a *lens* (or *filter*) function $f : X \rightarrow Y$ from a high-dimensional point cloud X to a lower-dimensional space Y , a *cover* of the target space Y , and a *clustering* algorithm for cover elements. Mapper is known to be sensitive to its parameters, and many researchers have studied how to optimize them [1, 4, 2, 13]. We illustrate Mapper’s large sensitivity to its parameters by providing constructions that show for a given graph G and data X , there exists a lens function $f : X \rightarrow Y$, cover \mathcal{U} of Y , and clustering algorithm whose Mapper graph is isomorphic to G .

2 Background

We define the necessary terms related to Mapper graphs for the results of this work. We refer the reader to [8, 5] for an overview of topology and computational topology respectively.

► **Definition 1** (Mapper Graph). *Let X and Y be sets, and let $f : X \rightarrow Y$ be a function. For any cover \mathcal{U} of Y , define the cover $f^*(\mathcal{U})$ of X as the collection of all clustered components of $f^{-1}(U)$ over all $U \in \mathcal{U}$. We define the Mapper construction of f as the simplicial complex $M(\mathcal{U}, f) := \text{Nrv}(f^*(\mathcal{U}))$ where $\text{Nrv}(f^*(\mathcal{U}))$ is the nerve of $f^*(\mathcal{U})$. The 1-skeleton of $M(\mathcal{U}, f)$, denoted by $M^{(1)}(\mathcal{U}, f)$ is called the Mapper graph of f .*

In practice, whenever X is a finite point cloud, one obtains the “clustered components of $f^{-1}(U)$ ” by running a clustering algorithm on each $f^{-1}(U)$. See Figure 1 for an example of a Mapper graph. One choice of a clustering algorithm is the *trivial clustering algorithm*, which returns $\{A\}$ for any given set $A \subseteq X$. Using the trivial clustering algorithm for the Mapper construction amounts to using the cover $f^{-1}(\mathcal{U}) := \{f^{-1}(U) : U \in \mathcal{U}\}$ of X . We call the simplicial complex $\text{MT}(\mathcal{U}, f) := \text{Nrv}(f^{-1}(\mathcal{U}))$ the *Mapper construction with trivial clustering*. In the construction of $\text{MT}(\mathcal{U}, f)$, we are simply taking the nerve of all inverse images of our cover \mathcal{U} . In the construction of $M(\mathcal{U}, f)$, we are taking the nerve *after* we split each inverse image into its clustered components.

3 Any graph is a Mapper graph when using trivial clustering

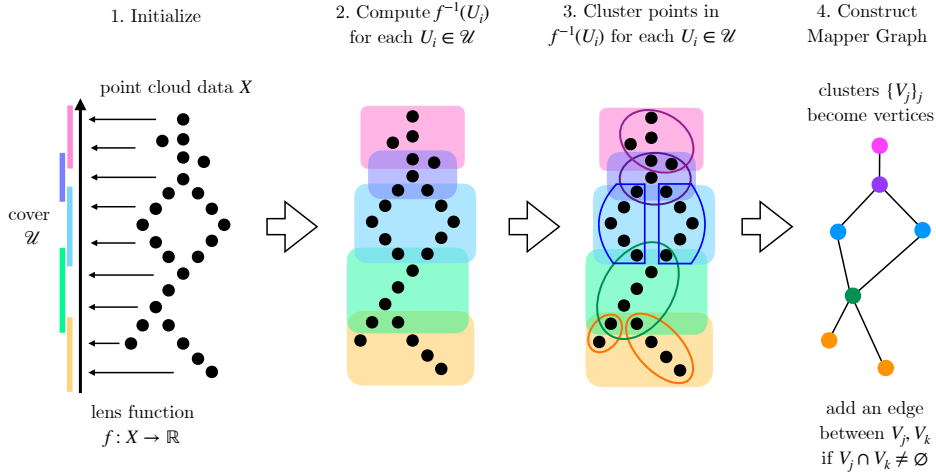
We show that any graph is a Mapper graph with trivial clustering using two constructions.

3.1 Using the star cover

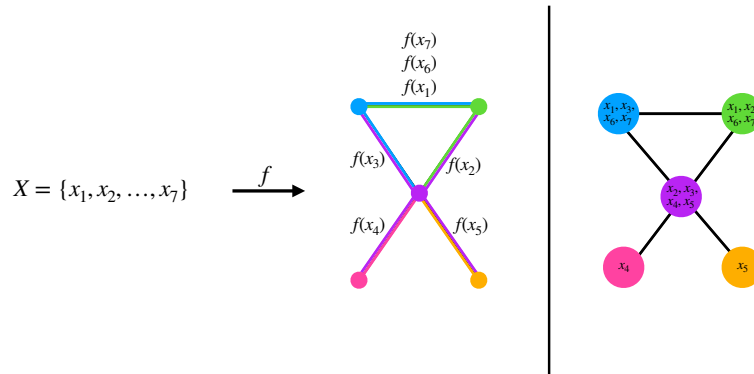
In this construction, we use the graph G as the co-domain. Additionally, we use the *star* of vertices in our graph to form the cover \mathcal{U} , where the star of a vertex v denoted as $\text{st}(v)$ in a graph is the subgraph consisting of all edges of G that contain v .

The construction leading to Theorem 2 involves using the star of each vertex as a cover element, and then mapping at least one element of X to each edge of G . See Figure 2 for an example of how to construct the cover and function.

► **Theorem 2.** *Let G be a graph with edge set E . If X is a finite set of points with cardinality $|X| \geq |E|$, then there exists a cover \mathcal{U} of E , and a function $f : X \rightarrow E$ such that the Mapper graph with trivial clustering $\text{MT}(\mathcal{U}, f)$ is isomorphic to G .*



■ **Figure 1** Mapper Graph Construction. An example of the four step procedure to construct a Mapper graph, which is the 1-skeleton of the nerve of the clustered sets in step 3.



■ **Figure 2** Mapper Graph Reconstruction. Left. Each color represents the star of a vertex. The set of all stars of vertices is the cover. The function f maps at least one element of X to each edge in G . Right. The Mapper graph is constructed using the function f and star cover. We get five vertices that are colored according to their corresponding cover element. The elements of X listed on each vertex are those mapped to that cover element. We verify that the Mapper graph is isomorphic to the original graph.

3.2 Using convex subsets as the co-domain

In the setting of using convex subsets of \mathbb{R}^3 as the co-domain, we can use a result from [14, 12, 10] that shows there exists a finite collection of convex subsets in \mathbb{R}^3 whose nerve is isomorphic to a given graph. We can then apply this result to give us a cover. We can choose our lens function so that f maps data points of X to the intersections between convex sets. This gives us Theorem 3.

► **Theorem 3.** *Let G be a graph with edge set E . If X is a finite set of points in \mathbb{R}^d with cardinality $|X| \geq |E|$, then there exists a collection \mathcal{C} of convex sets in \mathbb{R}^3 and a function $f : X \rightarrow \bigcup_{C \in \mathcal{C}} C$ such that the Mapper graph with trivial clustering $MT(\mathcal{C}, f)$ is isomorphic to G .*

4 Discussion


Although we are able to choose Mapper parameters to construct any graphical structure under mild assumptions, these parameters are contrived and often not what a data practitioner would choose. Investigating the inverse problem using a non-trivial clustering algorithm and standard software parameters is needed to understand if somebody could manipulate Mapper parameters to get a desired graphical structure. A result from [6] states that in any Euclidean space, there is a graph and a sufficiently large point cloud in general position that cannot be partitioned into parts such that the nerve of the collection of convex hulls of each part is isomorphic to the graph. This leads us to suspect that there exists a graph that is *not* a Mapper graph when restricted to more standard software Mapper parameters like cubical covers or partitional clustering algorithms.

References

- 1 Enrique Alvarado, Robin Belton, Emily Fischer, Kang-Ju Lee, Sourabh Palande, Sarah Percival, and Emilie Purvine. *g-mapper: Learning a cover in the mapper construction*. *arXiv preprint arXiv:2309.06634*, 2023.
- 2 Quang-Thinh Bui, Bay Vo, Hoang-Anh Nguyen Do, Nguyen Quoc Viet Hung, and Vaclav Snasel. F-mapper: A fuzzy mapper clustering algorithm. *Knowledge-Based Systems*, 189:105107, 2020.
- 3 Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
- 4 Nithin Chalapathi, Youjia Zhou, and Bei Wang. Adaptive covers for mapper graphs using information criteria. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3789–3800. IEEE, 2021.
- 5 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010. URL: <https://books.google.com/books?id=MDXa6gFRZuIC>.
- 6 Florian Frick and R. Amzi Jeffs. Colorful words and d-tverberg complexes, 2023. *arXiv:2205.04776*.
- 7 Li Li, Wei-Yi Cheng, Benjamin S Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P Bottinger, and Joel T Dudley. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science translational medicine*, 7(311):311ra174–311ra174, 2015.
- 8 James R. Munkres. *Topology: Second Edition*. Upper Saddle River: Prentice Hall, 2000.
- 9 Monica Nicolau, Arnold J Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

- 10 G. Ya Perel'man. Realization of abstract k -skeletons of intersections of convex polyhedra in \mathbb{R}^{2k-1} . *Geometric Questions in the Theory of Functions and Sets*, pages 129–131, 1985.
- 11 Gurjeet Singh, Facundo Mémoli, Gunnar E Carlsson, et al. Topological methods for the analysis of high dimensional data sets and 3D object recognition. *PBG@ Eurographics*, 2, 2007.
- 12 Martin Tancer. d -representability of simplicial complexes of fixed dimension. *Journal of Computational Geometry*, 2:183–188, 2011.
- 13 Mikael Vejdemo-Johansson and Alisa Leshchenko. Certified mapper: Repeated testing for acyclicity and obstructions to the nerve lemma. In *Topological Data Analysis*, pages 491–515, Cham, 2020. Springer International Publishing.
- 14 G Wegner. Eigenschaften der nerven homologisch-einfacher familien in \mathbb{R}^n . *Ph.D. thesis, GeorgAugust-Universität Göttingen*, 1967.

Differential of Generalized Rank Invariant Landscape (D-GRIL)

Tamal K. Dey ✉ 


Department of Computer Science, Purdue University, West Lafayette, USA

Soham Mukherjee ✉

Department of Computer Science, Purdue University, West Lafayette, USA

Steve Oudot ✉

Inria, Palaiseau, France

Shreyas N. Samaga ✉ 

Department of Computer Science, Purdue University, West Lafayette, USA

Cheng Xin ✉

Department of Computer Science, Rutgers University, USA

Abstract

Differentiating 1-parameter persistent homology vectorizations is well-known. We establish a theoretical foundation of differentiating Generalized Rank Invariant Landscape (GRIL), a vectorization technique for 2-parameter persistence modules. Further, we show that this framework can be used for bifiltration function learning in the context of machine learning.

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

Keywords and phrases Persistent homology, multiparameter persistence

1 Introduction

Recently, the authors in [5] introduced Generalized Rank Invariant Landscape (GRIL) as a vectorization method for 2-parameter persistence modules. GRIL is based on Generalized Rank Invariant [3] and the authors in [1] gave an algorithm to compute it. We establish the conditions under which GRIL is differentiable and compute an explicit formula for its differential. We perform experiments with various benchmark graph datasets and datasets from the domain of bio-activity prediction. A full version of the paper with all these details will be made public shortly.

1.1 GRIL as a piecewise affine map

► **Definition 1** (discrete ℓ -worm, [5]). Let $\square_{\mathbf{p}}^d := \{\mathbf{w} : \|\mathbf{p} - \mathbf{w}\|_\infty \leq d\}$ be the d -square centered at $\mathbf{p} \in \mathbb{R}^2$ with side $2d$ for a given $d > 0$. Given $\ell \geq 1$, the ℓ -worm, $\square_{\mathbf{p}}^{\ell, d}$, is defined as the union of all d -squares $\square_{\mathbf{q}}^d$ centered at some point \mathbf{q} on the off-diagonal line segment $\mathbf{p} \pm \alpha \cdot (1, -1)$ with $\alpha = j \cdot d$ where $j \in \{1, \dots, \ell - 1\}$.

► **Definition 2** (GRIL, [5]). For a 2-parameter persistence module M , the Generalized Rank Invariant Landscape (GRIL) is a function $\lambda^M : \mathbb{R}^2 \times \mathbb{N}_+ \times \mathbb{N}_+ \rightarrow \mathbb{R}$ defined as

$$\lambda^M(\mathbf{p}, k, \ell) := \sup \left\{ d \geq 0 : \text{rk}^M \left(\square_{\mathbf{p}}^{\ell, d} \right) \geq k \right\}.$$

This is an abstract of a presentation given at CG:YRF 2024. 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.

where $\text{rk}^M(I)$ denotes the Generalized Rank [3] of M over the interval I .

Let K be a simplicial complex with n simplices, labelled $\sigma_1, \sigma_2, \dots, \sigma_n$. A bifiltration function $f: K \rightarrow \mathbb{R}^2$ can be viewed as a vector $\mathbf{v}_f \in \mathbb{R}^{2n}$ where, for $k = 1, \dots, n$, $\mathbf{v}_f[2k-1] = f_x(\sigma_k)$ and $\mathbf{v}_f[2k] = f_y(\sigma_k)$. Here, $f_x(\sigma)$ and $f_y(\sigma)$ denote the x - and y -coordinates of the vector $f(\sigma)$ respectively. Notice that the vectors in \mathbb{R}^{2n} that correspond to a valid bifiltration function form a convex cone. We work with this set of vectors in \mathbb{R}^{2n} . In this setting, the authors in [5] show that GRIL is Lipschitz continuous in the following sense.

► **Proposition 3.** *Let X be a discrete space with $|X| = n$. For fixed k, ℓ, \mathbf{p} , let $\Lambda_{k,\ell}^{\mathbf{p}}: \mathbb{R}^{2n} \rightarrow \mathbb{R}$ be the map $\mathbf{v}_f \mapsto \lambda_{k,\ell}^{M_f}(\mathbf{p})$. Then, $\Lambda_{k,\ell}^{\mathbf{p}}$ is Lipschitz continuous.*

Let $\mathcal{G}_{k,\ell}: \mathbb{R}^{2n} \rightarrow \mathbb{R}^s$ be the map defined as:

$$\mathcal{G}_{k,\ell}(\mathbf{v}_f) = \left[\Lambda_{k,\ell}^{\mathbf{p}_1}(\mathbf{v}_f), \Lambda_{k,\ell}^{\mathbf{p}_2}(\mathbf{v}_f), \dots, \Lambda_{k,\ell}^{\mathbf{p}_s}(\mathbf{v}_f) \right]^T \quad (1)$$

where $\{\mathbf{p}_j\}_{j=1}^s$ are the s sampled center points. We drop the k, ℓ and refer to $\mathcal{G}_{k,\ell}$ as \mathcal{G} whenever k, ℓ are well understood. We show that \mathcal{G} is piecewise affine.

For notational convenience, in what follows we denote $f_x(\sigma)$ as σ^x and $f_y(\sigma)$ as σ^y , and we call them the *simplex coordinates* of σ . Similarly, we denote the x -coordinate and y -coordinate of \mathbf{p} as \mathbf{p}^x and \mathbf{p}^y respectively.

We observe that, if there are two simplices σ_i, σ_j such that for some $\rho \in \mathbb{Z}$ and $0 \leq \rho \leq \ell$ and $a, b \in \{x, y\}$, one has $|\sigma_i^a - \mathbf{p}_i^a| = \rho \cdot |\sigma_j^b - \mathbf{p}_j^b|$ then, say for $a = x$ and $b = y$, the point representing the vector \mathbf{v}_f lies on a hyperplane in \mathbb{R}^{2n} :

$$\{\mathbf{v} \in \mathbb{R}^{2n} : |\mathbf{v}[2i-1] - \mathbf{p}_i^x| = \rho \cdot |\mathbf{v}[2j] - \mathbf{p}_j^y|\}.$$

Corresponding to each such pair of simplex coordinates, we get one hyperplane. Combining all these hyperplanes, we get an *arrangement \mathcal{H} of hyperplanes* in \mathbb{R}^{2n} [2]. The arrangement \mathcal{H} partitions \mathbb{R}^{2n} into *relatively open r -cells*, $r \in \{0, \dots, 2n\}$. We observe that this arrangement induces an affine stratification $\mathcal{S}_{\mathcal{H}}$ (refer [4] for a formal definition) of \mathbb{R}^{2n} where the r -dimensional strata are precisely the r -cells.

► **Proposition 4.** *The top-dimensional strata of $\mathcal{S}_{\mathcal{H}}$ consist precisely of those bifiltration functions that have a unique constraining simplex coordinate for each ℓ -worm.*

► **Theorem 5.** *Let K be a simplicial complex with n simplices. Let $k, \ell \in \mathbb{N}$, and let $\{\mathbf{p}_j\}_{j=1}^s$ be the s sampled center points for the ℓ -worms. Then, \mathcal{G} , as defined in Eq.(1), is a piecewise affine map relative to the arrangement \mathcal{H} .*

Overview of the proof: \mathcal{G} depends affinely on the simplex coordinates in each top-dimensional stratum, because there is a unique constraining simplex coordinate for each ℓ -worm. By continuity of \mathcal{G} (Proposition 3), the restriction of \mathcal{G} to each (affine) lower dimensional stratum is affine.

1.2 Differential of GRIL

► **Definition 6** (Upper and lower boundary of ℓ -worm). *Let an ℓ -worm centered at \mathbf{p} with width d , denoted as $\square_{\mathbf{p}}^{\ell}$, be given. A point \mathbf{t} is said to be on the upper boundary of the worm if $\mathring{\mathbf{t}}^{(\uparrow \mathbb{R}^2)} \cap \square_{\mathbf{p}}^{\ell} = \emptyset$ where $\mathring{\mathbf{t}}^{(\uparrow \mathbb{R}^2)}$ denotes the open upper-set of \mathbf{t} in \mathbb{R}^2 . The collection of all such points constitutes the upper boundary of the worm. Similarly, a point t is on the lower boundary if $\mathring{\mathbf{t}}^{(\downarrow \mathbb{R}^2)} \cap \square_{\mathbf{p}}^{\ell} = \emptyset$ and the collection of all such points constitutes the lower boundary of the worm.*

► **Definition 7** (Constraining Simplex Coordinate). *Given a bifiltration function f , $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\mathbf{p} \in \mathbb{R}^2$ let $\lambda^{Mf}(\mathbf{p}, k, \ell) = d$. Let σ be a simplex with $\boxed{\mathbf{p}}_d^\ell \cap f(\sigma)^{\uparrow \mathbb{R}^2} \neq \emptyset$ such that one of the following two conditions holds: (1) $f_x(\sigma) = \mathbf{p}_x \pm j \cdot d$ or (2) $f_y(\sigma) = \mathbf{p}_y \pm j \cdot d$ for some $j \in \{1, \dots, \ell - 1\}$. Then σ is called a constraining simplex for $\boxed{\mathbf{p}}_d^\ell$. Depending on if σ satisfies (1) or (2), σ is called x -constraining or y -constraining respectively.*

► **Definition 8.** *Let f be a bifiltration function. Let σ be a constraining simplex for $\boxed{\mathbf{p}}_d^\ell$. The simplex σ is said to be an upper constraining simplex if $f(\sigma)^{\uparrow \mathbb{R}^2}$ intersects only the upper boundary of $\boxed{\mathbf{p}}_d^\ell$. The simplex σ is called a lower constraining simplex if $f(\sigma)^{\uparrow \mathbb{R}^2}$ intersects both lower and upper boundary of $\boxed{\mathbf{p}}_d^\ell$. σ is said to be lower x -constraining if σ is lower constraining and σ^x is the constraining simplex coordinate. The notions of upper x -constraining, lower y -constraining and upper y -constraining are similarly defined.*

► **Theorem 9.** *Let K be a simplicial complex with n simplices. Let $k, \ell \in \mathbb{N}$ and $\{\mathbf{p}_j\}_{j=1}^s$ be the s sampled center points for the ℓ -worms. Then, the differential of \mathcal{G} at any \mathbf{v}_f in a top-dimensional stratum is given by:*

$$\begin{pmatrix} \frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^1}(\mathbf{v}_f)}{\partial \sigma_1^x} & \frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^1}(\mathbf{v}_f)}{\partial \sigma_1^y} & \frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^1}(\mathbf{v}_f)}{\partial \sigma_2^x} & \cdots & \frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^1}(\mathbf{v}_f)}{\partial \sigma_n^y} \\ \vdots & & & & \vdots \\ \frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^s}(\mathbf{v}_f)}{\partial \sigma_1^x} & \cdots & \cdots & \cdots & \frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^s}(\mathbf{v}_f)}{\partial \sigma_n^y} \end{pmatrix}_{s \times 2n}$$

where,

$$\frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^j}(\mathbf{v}_f)}{\partial \sigma_i^x} = \begin{cases} -1, & \text{if } \sigma_i \text{ is lower } x\text{-constraining for } \boxed{\mathbf{p}}_{d_j}^\ell \\ +1, & \text{if } \sigma_i \text{ is upper } x\text{-constraining for } \boxed{\mathbf{p}}_{d_j}^\ell \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial \Lambda_{k,\ell}^{\mathbf{P}^j}(\mathbf{v}_f)}{\partial \sigma_i^y} = \begin{cases} -1, & \text{if } \sigma_i \text{ is lower } y\text{-constraining for } \boxed{\mathbf{p}}_{d_j}^\ell \\ +1, & \text{if } \sigma_i \text{ is upper } y\text{-constraining for } \boxed{\mathbf{p}}_{d_j}^\ell \\ 0, & \text{otherwise} \end{cases}$$

and d_j is the GRIL value at \mathbf{p}_j .

References

- 1 Tamal K. Dey, Woojin Kim, and Facundo Mémoli. Computing generalized rank invariant for 2-parameter persistence modules via zigzag persistence and its applications. In *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pages 34:1–34:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: <https://link.springer.com/article/10.1007/s00454-023-00584-z>.
- 2 J. E. Goodman and J. O'Rourke. *Handbook of Discrete and Computational Geometry, Second Edition*. Graduate Texts in Mathematics, Vol. 5. CRC Press, Boca Raton, FL, 2004.
- 3 Woojin Kim and Facundo Mémoli. Generalized persistence diagrams for persistence modules over posets. *Journal of Applied and Computational Topology*, 5(4):533–581, 12 2021. doi: 10.1007/s41468-021-00075-1.
- 4 Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. A framework for differential calculus on persistence barcodes. *Foundations of Computational Mathematics*, pages 1–63, 2021. URL: <https://link.springer.com/article/10.1007/s10208-021-09522-y>.

4 Differential of Generalized Rank Invariant Landscape (D-GRIL)

- 5 Cheng Xin, Soham Mukherjee, Shreyas N. Samaga, and Tamal K. Dey. GRIL: a 2-parameter persistence based vectorization for machine learning. In *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, volume 221 of *Proceedings of Machine Learning Research*, pages 313–333. PMLR, 7 2023. URL: <https://proceedings.mlr.press/v221/xin23a.html>.

Spatiotemporal Persistence Landscapes

Martina Flammer ✉

University of Würzburg, Germany

Ansbach University of Applied Sciences, Germany

Abstract

We propose an extended type of zigzag diagram to analyze time series via persistent homology. This type fits neither in the framework of standard zigzag persistent homology nor in the framework of multi-parameter persistent homology. We propose an algorithm to calculate *spatiotemporal persistence landscapes* for this type of persistence modules. This yields a vector space valued invariant that is amenable to statistics and can be used as an input for machine learning algorithms.

2012 ACM Subject Classification Theory of computation → Theory and algorithms for application domains

Keywords and phrases Topological data analysis, persistence landscapes, time series analysis, zigzag persistence, generalized rank invariant

Funding This work has been supported by the German Federal Ministry of Education and Research (BMBF, funding numbers: 05M20WBA and 05M20WWA, Verbundprojekt 05M2020 - DyCA).

1 Introduction

Persistent homology is one of the most frequently used methods in topological data analysis. It can be used to extract structural information of sampled data on several spatial scales. The object containing this information is called a persistence module. Mathematically, it is described as a functor from an indexing poset $S = \{\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_m\}$ to the category of finite dimensional vector spaces \mathbf{vec} . The *structure theorem* [8] states that any persistence module of this kind can be uniquely decomposed into a direct sum of interval modules, yielding a complete invariant, the *barcode*. The same holds for zigzag modules, that are functors where the poset S has the shape $S = \{\alpha_1 \leq \alpha_2 \geq \alpha_3 \leq \dots \geq \alpha_n\}$.

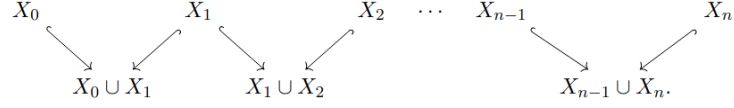
In case of multiple filtration values, persistence modules can be defined as functors from some poset $S \subset \mathbb{R}^n$ to \mathbf{vec} . This setting is called multi-parameter persistent homology. In this case no complete invariant exists and plenty of research has been done to find visualization methods or other useful, but incomplete, invariants. One of these invariants are persistence landscapes, invariants that have been initially proposed for one-parameter persistence modules [1] and afterwards extended to multi-parameter persistence modules [7]. Being valued in a Banach space, this invariant can be used for statistics as well as input for machine learning algorithms.

Time series analysis

In one-parameter persistent homology, one frequently considers how the homology changes when a filtration parameter of the Vietoris-Rips complex rises. However, when considering a windowed time series one would like to track the changes of the homology as the point cloud changes. A possibility to track the persistence of homological features throughout the time is zigzag persistence [2, 6, 3]. Since the simplicial complexes defined for each window are

This is an abstract of a presentation given at CG:YRF 2024. 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.

neither subcomplexes nor supercomplexes of each other, two neighboring complexes X_i and X_{i+1} are included into the bigger complex $X_i \cup X_{i+1}$, yielding the zigzag sequence



However, the filtration parameter ϵ for each complex has to be fixed. To account for several filtration parameters, we propose a filtered zigzag sequence and hence, a diagram as shown in Figure 1.

Since this diagram is not even a multi-parameter persistence module, no visualization method is known. In our work, we adapt the framework of persistence landscapes to our setting by using a generalization of the rank invariant [5].

2 Spatiotemporal Persistence Landscapes

In this section we define *spatiotemporal persistence landscapes*, an invariant and visualization method for modules that have a shape as in Figure 1.

2.1 Definition

We define $(\mathbb{Z}\mathbb{Z}, \ll)$, the underlying poset for the persistence modules that we regard, as follows: $\mathbb{Z}\mathbb{Z} = \mathbb{Z}^2$ and

$$(a, b) \ll (a', b') \Leftrightarrow a \leq a' \text{ and } b = b' \pm 1 \text{ if } b = 2z + 1 \text{ for some } z \in \mathbb{Z} \text{ or } b = b'.$$

As in the case of multi-parameter persistence landscapes, we are interested in the rank over regions R_x^δ in the parameter space that we define as

$$R_x^\delta = \{y \in \mathbb{Z}\mathbb{Z} : y = x + h \text{ with } h \in \mathbb{Z}\mathbb{Z}, d(h, 0) \leq \delta\},$$

where d is the maximum norm in \mathbb{Z}^2 . To calculate the rank over regions we use the generalized rank invariant [5].

► **Definition 1.** *The k -th persistence landscape λ_k of a persistence module $M : \mathbb{Z}\mathbb{Z} \rightarrow \mathbf{vec}$ considers the maximal radius over which k features persist in every (positive) direction through x in the parameter space*

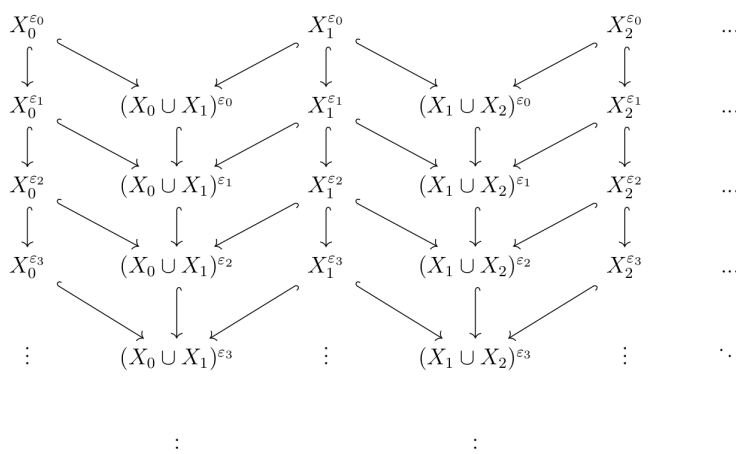
$$\lambda_k(x) := \max\{\delta \geq 0 : \text{rank}(M|_{R_x^\delta}) \geq k\}.$$

The persistence landscape λ of M is the map $\lambda : \mathbb{N} \times \mathbb{Z}\mathbb{Z} \rightarrow \mathbb{R}$, $(k, x) \mapsto \lambda_k(x)$.

Analogously to one- and multi-parameter persistence landscapes, we obtain the following lemma.

► **Lemma 2.** *The persistence landscapes have the properties:*

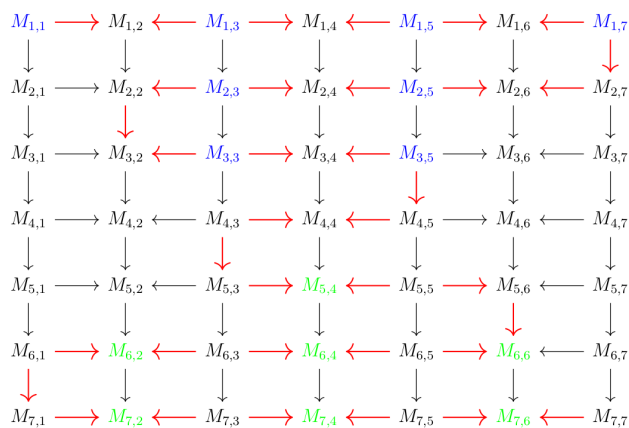
- (i) $\lambda_k(x) \geq 0$
- (ii) $\lambda_k(x) \geq \lambda_{k+1}(x)$
- (iii) λ_k is 1-Lipschitz.



■ **Figure 1** Extended zigzag diagram.

2.2 Algorithm

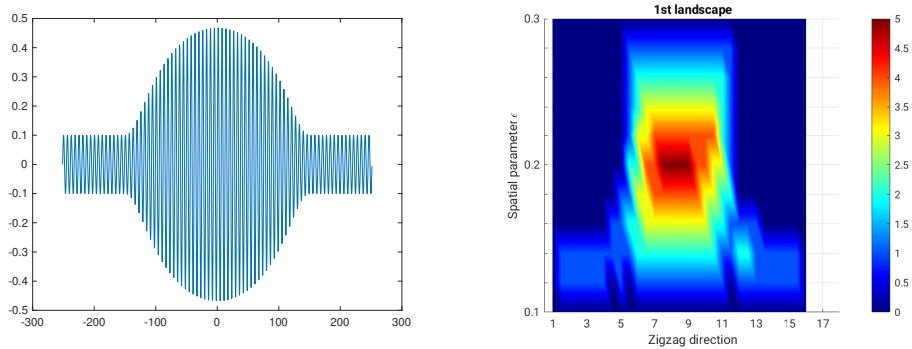
To compute the spatiotemporal persistence landscapes, we use a result by Dey *et al.* [4] that states that the generalized rank over a region can be computed as the rank of a certain zigzag sequence along the boundary of that region. Hence, only the barcodes of certain zigzag sequences have to be determined. Due to space limitations, we omit the detailed description of the choice of the zigzag paths at this point and show only an example for the point $x = (4, 4)$ in Figure 2.



■ **Figure 2** Red: Zigzag path in the parameter space to compute the landscapes at point $(4, 4)$.

3 Application

We apply our algorithm to the time series shown on the left hand side of Figure 3. We partition the time series into 16 windows of approximately 31 seconds length and use time delay embedding for every window with embedding dimension 2. The resulting sequence of point clouds can be seen in Figure 4. On the right hand side of Figure 3, the resulting first landscape in homological dimension one is shown. As expected, for a smaller parameter ϵ of the Vietoris-Rips complex one can see that also the small holes in the beginning and end

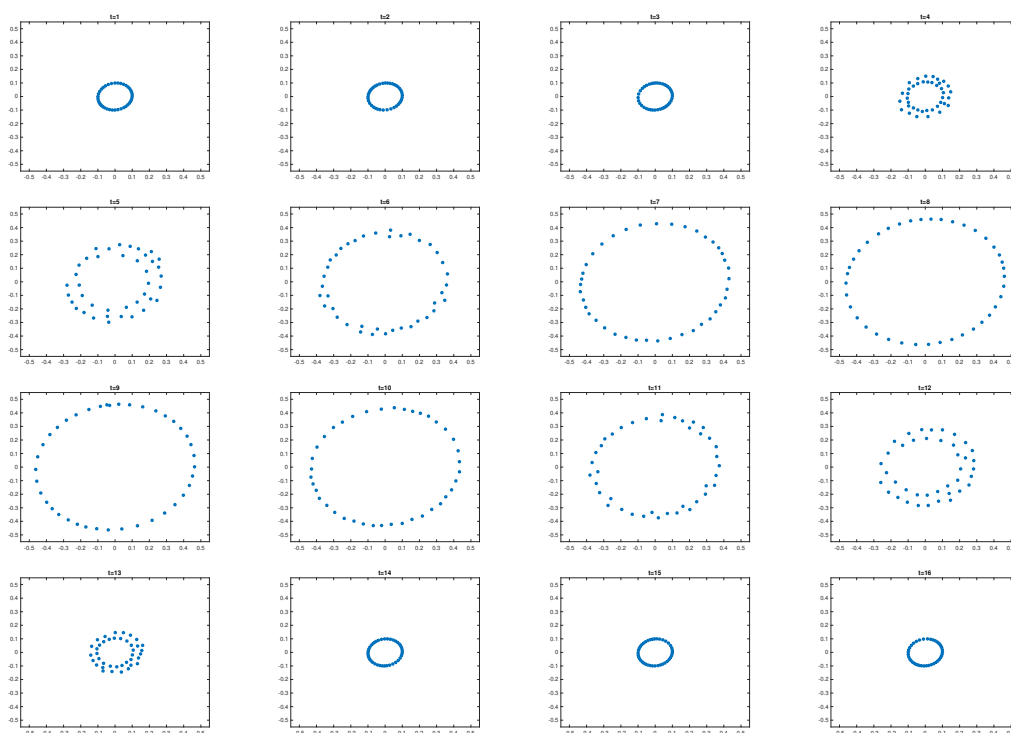


■ **Figure 3** Initial time series (left) and resulting 1st landscape in dimension one (right).

of the time series are detected, but as ϵ grows they vanish and only the bigger hole in the central part of the time series remains.

References

- 1 Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 2015. URL: <http://dl.acm.org/citation.cfm?id=2789275>.
- 2 Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010. doi:10.1007/s10208-010-9066-0.
- 3 Pdraig Corcoran and Christopher B. Jones. Modelling topological features of swarm behaviour in space and time with persistence landscapes. *IEEE Access*, 5:18534–18544, 2017. doi:10.1109/ACCESS.2017.2749319.
- 4 Tamal K. Dey, Woojin Kim, and Facundo Mémoli. Computing Generalized Rank Invariant for 2-Parameter Persistence Modules via Zigzag Persistence and Its Applications. In *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:17, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/opus/volltexte/2022/16042>, doi:10.4230/LIPIcs.SoCG.2022.34.
- 5 Woojin Kim and Facundo Mémoli. Generalized persistence diagrams for persistence modules over posets. *Journal of Applied and Computational Topology*, 5(4):533–581, 2021. doi:10.1007/s41468-021-00075-1.
- 6 Sarah Tymochko, Elizabeth Munch, and Firas Khasawneh. Using zigzag persistent homology to detect hopf bifurcations in dynamical systems. *Algorithms*, 13:278, 10 2020. doi:10.3390/a13110278.
- 7 Oliver Vipond. Multiparameter persistence landscapes. *Journal of Machine Learning Research*, 21(61):1–38, 2020. URL: <http://jmlr.org/papers/v21/19-054.html>.
- 8 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005. doi:10.1007/s00454-004-1146-y.



■ **Figure 4** Time delay embedding of the windowed time series.

Tagged barcodes for the topological analysis of gradient-like vector fields

Clemens Bannwart

Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Italy

Claudia Landi

Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy

Abstract

We present a pipeline that takes as an input a weighted based chain complex, produces a tame epimorphic parametrized chain complex, and encodes it as a tagged barcode. We show how to apply this pipeline to the weighted based chain complex of a gradient-like Morse-Smale vector field on a compact Riemannian manifold in both the smooth and discrete settings.

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

Keywords and phrases parametrized chain complexes, interval complex, interleaving distance, bottleneck distance, Morse-Smale vector fields, Morse complex, combinatorial realization

Related Version A full version of the paper is available at <https://arxiv.org/abs/2401.08466>.

Funding The authors are members of GNSAGA-INdAM (Italy) and carried out this work within the activities of ARCES (University of Bologna).

Claudia Landi: Partially supported by DISMI-UniMORE (FAR2023).

1 Introduction

In topological data analysis (TDA), persistent homology studies a continuous function $f: M \rightarrow \mathbb{R}$ by filtering M by the sublevel sets of f and then applying homology in order to obtain a persistence barcode [12]. Our goal is to develop a persistence theory for gradient-like vector fields on Riemannian manifolds. Such a vector field may not be the gradient of any function, so a difficulty is that there is no canonical filtration of M . Inspired by some recent papers, see e.g. [13, 5, 4, 11], we are decomposing parametrized chain complexes, but with epimorphic rather than the usual monomorphic internal maps that come from filtrations.

2 Tame epimorphic parametrized chain complexes

A **parametrized chain complex** is a functor $X: [0, \infty) \rightarrow \text{Ch}$, where $[0, \infty)$ denotes the poset category of the totally ordered set of non-negative real numbers and Ch is the category of non-negative chain complexes [14] of finite total dimension with coefficients in a fixed field \mathbb{F} . We call X **epimorphic** if all the appearing internal chain maps are degreewise surjective and **tame** if there exist $0 = t_0 < t_1 < \dots < t_r < t_{r+1} = \infty$ such that, for any $i = 0, \dots, r$ and $t_i \leq s \leq t < t_{i+1}$, the internal chain map $X^{s \leq t}: X^s \rightarrow X^t$ is an isomorphism. We write TEPCh for the category of tame epimorphic parametrized chain complexes, where morphisms are given by natural transformations.

This is an abstract of a presentation given at CG:YRF 2024. 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.

Following [6], given $n \geq 1$, the n -**disk** of Ch is the chain complex $D^n = (D^\bullet, \partial_\bullet)$ with

$$D_k^n = \begin{cases} \mathbb{F}, & \text{if } k = n, n-1, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad \partial_k = \begin{cases} \text{Id}_{\mathbb{F}}, & \text{if } k = n, \\ 0, & \text{otherwise,} \end{cases} \quad \text{for all } k \in \mathbb{Z}.$$

Given $n \geq 0$, the n -**sphere** of Ch is the chain complex $S^n = (S^\bullet, \partial_\bullet)$ with

$$S_k^n = \begin{cases} \mathbb{F}, & \text{if } k = n, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad \partial_k = 0, \quad \text{for all } k \in \mathbb{Z}.$$

These chain complexes are indecomposable and every $X \in \text{Ch}$ can be written as a finite direct sum of disks and spheres in a unique way (only true with field coefficients, see e.g. [14]). The only epimorphic chain maps between the disks and spheres of Ch are the identities and $\Psi^n: D^n \rightarrow S^n$, which is the identity in degree n and zero in all other degrees. It follows that the indecomposable objects of TEPCh are of the following form.

► **Definition 1.** *Let $n \in \mathbb{N}$ and $0 \leq s \leq t \leq \infty$. In the case $n = 0$ we additionally demand that $s = 0$. Then we define the **interval complex** $\mathcal{I}^n[0, s, t]$ in TEPCh by*

$$(\mathcal{I}^n[0, s, t])^r = \begin{cases} D^n, & \text{if } 0 \leq r < s, \\ S^n, & \text{if } s \leq r < t, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad (\mathcal{I}^n[u, s, t])^{q \leq r} = \begin{cases} \text{Id}_{D^n}, & \text{if } 0 \leq q \leq r < s, \\ \Psi^n, & \text{if } 0 \leq q < s \leq r < t, \\ \text{Id}_{S^n}, & \text{if } s \leq q \leq r < t, \\ 0, & \text{otherwise.} \end{cases}$$

Our first main result resembles the classical result for parametrized vector spaces, as used in persistent homology [3], however for epimorphic parametrized chain complexes instead. General parametrized chain complexes are of wild representation type (see e.g. [7]), whereas for the monomorphic case see e.g. [5]. The epimorphic case is a contribution of this paper.

► **Theorem 2 (Structure theorem).** *For every $X \in \text{TEPCh}$, there exists a unique multiset $\text{tBar}(X)$, called the **tagged barcode** of X , such that $X \cong \bigoplus_{(n,s,t) \in \text{tBar}(X)} \mathcal{I}^n[0, s, t]$.*

Given $X, Y \in \text{TEPCh}$, we denote by $d_I(X, Y)$ their generalized interleaving distance as defined in [2]. By $d_B(X, Y)$ we denote the generalized bottleneck distance as defined in [10].

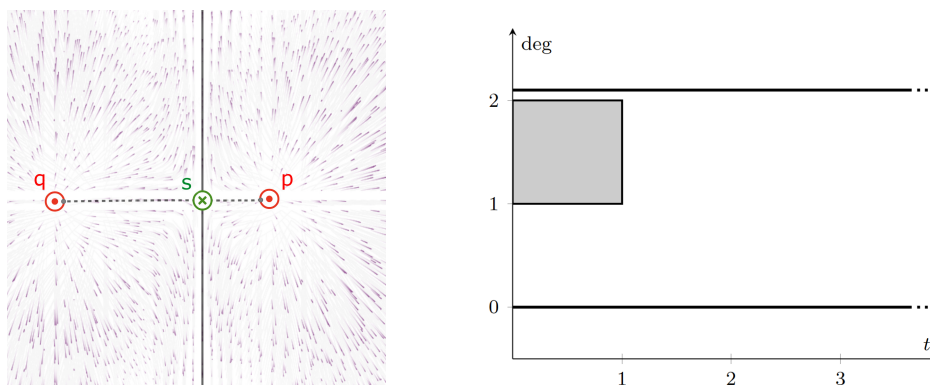
► **Theorem 3 (Isometry theorem).** *For every $X, Y \in \text{TEPCh}$, we have $d_I(X, Y) = d_B(X, Y)$.*

3 Parametrizing based chain complexes

We now describe a method that starts with a chain complex and repeatedly quotients out disks, keeping track of how long they persist before getting simplified. Precisely, we assign a tame epimorphic parametrized chain complex to a **weighted based chain complex**, which is a chain complex C_\bullet , endowed with a basis \mathcal{B}_k of C_k for each k and with **weights** $w(a, b) \in (0, \infty)$ for all $a \in \mathcal{B}_k$ and $b \in \mathcal{B}_{k-1}$. A weighted based chain complex is called **generic** if all the weights are pairwise different.

► **Definition 4.** *For a generic weighted based chain complex $C_\bullet = (C_\bullet, \mathcal{B}_\bullet, w)$, we define the parametrized chain complex $Y = Y(C_\bullet)$ and the **tagged barcode** $\text{tBar} = \text{tBar}(Y)$ as follows.*

- (1) Initialize $\text{tBar} := \emptyset$.
- (2) Find the pair $(a, b) \in \mathcal{B}_k \times \mathcal{B}_{k-1}$, among all k , that has the smallest weight $w(a, b)$ and satisfies the condition: $b \in \partial a$. If there is no such pair, that is $\partial_\bullet = 0$, go to Step (5).



■ **Figure 1** Left: Vector field v on the 2-sphere with singular points p, q of index 2, s of index 1, and x (represented by the whole boundary) of index 0. We have $Y(\text{MC}_\bullet(v)) = \mathcal{I}^2[0, 1, 1] \oplus \mathcal{I}^0[0, 0, \infty] \oplus \mathcal{I}^2[0, 0, \infty]$, assuming p and s have distance 1. Right: Visualization of the tagged barcode of v .

- (3) Update tBar by adding (n, t, t) , where $t = w(a, b)$ and n is such that $(a, b) \in \mathcal{B}_n \times \mathcal{B}_{n-1}$.
- (4) Update C_\bullet by quotienting it by the n -disk of Ch generated by a , i.e.

$$C_n \rightsquigarrow C_n / \text{Span}(a), \quad C_{n-1} \rightsquigarrow C_{n-1} / \text{Span}(\partial a), \quad C_k \rightsquigarrow C_k \text{ for } k \neq n, n-1.$$

The differential on C_\bullet induces a differential on the quotient. Update \mathcal{B}_n and \mathcal{B}_{n-1} by deleting a and b , respectively. Repeat from Step (2).

- (5) For each $n \geq 0$, add as many copies of $(n, 0, \infty)$ to tBar as the dimension of C_n .

- (6) Set $Y := \bigoplus_{(n,t,t) \in \text{tBar}} \mathcal{I}^n[0, t, t] \oplus \bigoplus_{(n,0,\infty) \in \text{tBar}} \mathcal{I}^n[0, 0, \infty] \in \text{TEPCh}$.

4 Application to smooth and discrete vector fields

We study generic enough smooth vector fields (precisely, Morse-Smale with pairwise different distances between singular points), without closed orbits (i.e. gradient-like). Given such v on a Riemannian manifold M , let $\text{MC}_\bullet(v)$ be its Morse complex [1], viewed as a weighted based chain complex with bases and weights given by the singular points of v and distances between them, respectively. Definition 4 then yields a tagged barcode for v (see Figure 1).

For computations, it is important to approximate smooth objects. By [9], there exists a triangulation \overline{M} of M and a discrete vector field V on \overline{M} , such that $\text{MC}_\bullet(v) \cong \overline{\text{MC}}_\bullet(V)$ as based chain complexes, where $\overline{\text{MC}}_\bullet(V)$ is the combinatorial Morse complex [8]. We view $\overline{\text{MC}}_\bullet(V)$ as a weighted based chain complex as well, using the distances between the barycenters of the critical cells, thus Definition 4 applies. By [15], V induces a new discrete vector field $\Delta(V)$ on the barycentric subdivision $\Delta(\overline{M})$ of \overline{M} , with $\overline{\text{MC}}_\bullet(V) \cong \overline{\text{MC}}_\bullet(\Delta(V))$. By iterating this process, we can approximate $Y(\text{MC}_\bullet(v))$ arbitrarily well.

► **Theorem 5.** For every $\varepsilon > 0$ there exists $N \in \mathbb{N}$ such that for all $n \geq N$

$$d_I(Y(\text{MC}_\bullet(v)), Y(\overline{\text{MC}}_\bullet(\Delta^n(V)))) < \varepsilon.$$

References

- 1 Augustin Banyaga and David Hurtubise. *Lectures on Morse Homology*. Texts in the Mathematical Sciences. Springer Netherlands, 2004. URL: https://books.google.it/books?id=AX-_sbMjOK4C.

- 2 Peter Bubenik, Vin de Silva, and Jonathan Scott. Metrics for generalized persistence modules. *Foundations of Computational Mathematics*, 15(6):1501–1531, oct 2014. URL: <https://doi.org/10.1007%2Fs10208-014-9229-5>, doi:10.1007/s10208-014-9229-5.
- 3 Gunnar Carlsson and Afra Zomorodian. Computing persistent homology. *Discrete & Computational Geometry*, 33:249–274, 2005. doi:10.1007/s00454-004-1146-y.
- 4 Wojciech Chachólski, Barbara Giunti, Jin Alvin, and Claudia Landi. Decomposing filtered chain complexes: Geometry behind barcoding algorithms. *Computational Geometry: Theory and Applications*, 109, 2023. doi:10.1016/j.comgeo.2022.101938.
- 5 Wojciech Chachólski, Barbara Giunti, and Claudia Landi. Invariants for tame parametrised chain complexes. *Homology, Homotopy, and Applications*, 23(2):183–213, 2021.
- 6 William G. Dwyer and Jan Spaliński. Homotopy theories and model categories. In I.M. James, editor, *Handbook of Algebraic Topology*, pages 73–126. North-Holland, Amsterdam, 1995. URL: <https://www.sciencedirect.com/science/article/pii/B9780444817792500031>, doi:10.1016/B978-044481779-2/50003-1.
- 7 Emerson G. Escobar and Yasuaki Hiraoka. Persistence modules on commutative ladders of finite type. *Discrete & Computational Geometry*, 55:100–157, 2016. doi:10.1007/s00454-015-9746-2.
- 8 Robin Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90–145, 1998. URL: <https://www.sciencedirect.com/science/article/pii/S0001870897916509>, doi:10.1006/aima.1997.1650.
- 9 Étienne Gallais. Combinatorial realization of the Thom-Smale complex via discrete Morse theory. *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze*, Ser. 5, 9(2):229–252, 2010. URL: http://www.numdam.org/item/ASNSP_2010_5_9_2_229_0/.
- 10 Killian Meehan and David C. Meyer. An isometry theorem for generalized persistence modules. *arXiv: Algebraic Topology*, 2017. URL: <https://api.semanticscholar.org/CorpusID:119596542>.
- 11 Facundo Mémoli and Ling Zhou. Ephemeral persistence features and the stability of filtered chain complexes. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 51:1–51:18, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.SoCG.2023.51>, doi:10.4230/LIPIcs.SoCG.2023.51.
- 12 Steve Y. Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*. Number 209 in *Mathematical Surveys and Monographs*. American Mathematical Society, 2015. URL: <https://inria.hal.science/hal-01247501>.
- 13 Michael Usher and Jun Zhang. Persistent homology and Floer–Novikov theory. *Geometry and Topology*, 20:3333–3430, 2016.
- 14 Charles A. Weibel. *An Introduction to Homological Algebra*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1994. URL: https://books.google.it/books?id=flm-dBXfZ_gC.
- 15 Alena M. Zhukova. Discrete Morse theory for the barycentric subdivision. *Journal of Mathematical Sciences*, 232:129–137, 07 2018. doi:10.1007/s10958-018-3863-4.

The Walk-Length Filtration for Persistent Homology on Weighted Directed Graphs

David E. Muñoz ✉🏠

Computational Mathematics, Science and Engineering, Michigan State University, United States

Elizabeth Munch ✉🏠

Computational Mathematics, Science and Engineering, Michigan State University, United States

Brittany Terese Fasy ✉🏠

Montana State University, United States

Firas A. Khasawneh ✉🏠

Mechanical Engineering, Michigan State University, United States

1 Introduction

Topological data analysis, and particularly persistent homology, has been used in many applications where input data arises as graphs [1]. Recently, particularly with the rise in neuroscience data [6], there is a push for more methods for measuring structure that incorporates directionality [2, 3, 4, 5, 7].

We introduce a new method for analyzing directed graphs with persistent homology via a simplicial filtration, called the *walk-length filtration*, that encodes information about when a walk visits all vertices in a particular set to determine when a simplex should be included. Note that if we asked for a path that visits every vertex just once, this would be equivalent to finding Hamiltonian paths, which is an NP-complete problem, hence we focus on walks. While this filtration is not stable under the network distance used in related work, we define a modification that replaces the maximum with a summation for which stability can be obtained. Finally, we show our results of applying walk-length persistence in cycle networks.

2 Definition

A *weighted directed graph (digraph)* is a triple $D = (V, E, w)$, where V is a finite set of vertices, $E \subseteq V \times V$ is a set of directed edges and there is a weight on the edges given by $w : E \rightarrow \mathbb{R}_{\geq 0}$. We assume that $w(v, v') = 0$ if and only if $v = v'$. A *walk of length n* in D is any sequence of vertices $\gamma = (v_0, \dots, v_n)$ where $(v_i, v_{i+1}) \in E$ for $0 \leq i \leq n - 1$. Here the vertices can be repeated. The *weight of a walk γ* is given by $w(\gamma) := \sum_{i=0}^{n-1} w(v_i, v_{i+1})$. The collection of all complete weighted digraphs $(V, V \times V, w)$ will be denoted by \mathcal{N} .

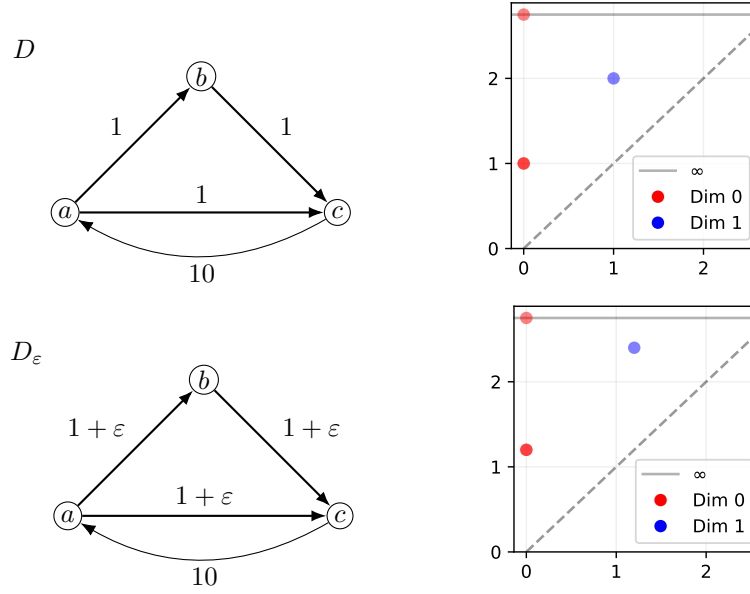
Let $D = (V, E, w)$ be a weighted digraph. For any subset of vertices $\sigma \subseteq V$, define

$$f_D(\sigma) = \inf\{w(\gamma) : \gamma \text{ is a walk in } D \text{ that contains all vertices in } \sigma\}.$$

If there is no such walk, we set $f(\sigma) = \infty$.

For $\delta \in \mathbb{R}$, we can define a simplicial complex $K_\delta = \{\sigma \subseteq V : f(\sigma) \leq \delta\}$. Then, the *walk-length filtration* is the parameterized collection of simplicial complexes $\{K_\delta\}_{\delta \in \mathbb{R}}$ and $\text{Dgm}_k^{\text{WL}}(D)$ is defined as the k -dimensional *walk-length persistence diagram* associated to the corresponding walk-length filtration $\{K_\delta\}_{\delta \in \mathbb{R}}$.

This is an abstract of a presentation given at CG:YRF 2024. 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** A counterexample to stability under an ℓ_∞ -distance. The one-dimensional persistence points are $(1, 2)$ for D and $(1 + \varepsilon, 2 + 2\varepsilon)$ for D_ε .

Given a strongly connected weighted digraph $D = (V, E, w)$, the *shortest-distance function* $\omega_D: V \times V \rightarrow \mathbb{R}$ is given by $\omega_D(u, v) := \min\{w(\gamma) : \gamma \text{ is a walk in } D \text{ from } u \text{ to } v\}$. The complete weighted digraph (V, ω_D) is called the *shortest-distance digraph associated to* D .

► **Proposition 1.** *Let $D = (V, E, w)$ be a strongly connected weighted digraph, and let $\mathcal{X} = (V, \omega_D)$ be the shortest-distance digraph associated to D . Then, the walk-length filtrations for D and \mathcal{X} are the same.*

3 Stability

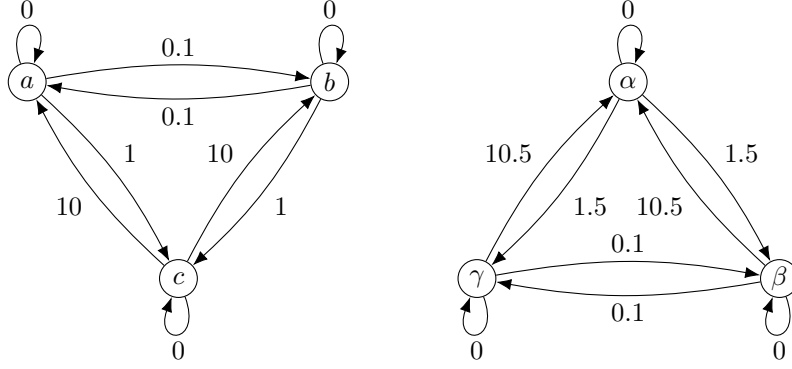
One useful property is that of stability: if the input data is close in some metric, then the resulting representation is closer in its own metric. For this purpose, we work with complete weighted digraphs, where there is an edge for any two distinct vertices; these can also be called *networks*. The standard in the persistence for digraphs literature is to use the network distance $d_{\mathcal{N}}$ [2, 3, 7]. We first point out that our filtration is not stable under this distance. A counterexample is shown in Figure 1. In essence, the issue arises because our filtrating function \mathfrak{f} is a sum in the walk length but the network distance $d_{\mathcal{N}}$ is a maximum. For this reason, we define a summation version of the network distance.

► **Definition 2.** *Let $\mathcal{X} = (X, \omega_X)$ and $\mathcal{Y} = (Y, \omega_Y)$ be two networks. Define the ℓ_1 -distortion of a correspondence $R \subset X \times Y$ as*

$$\text{dis}^1(R) := \sum_{(x,y),(x',y') \in R} |\omega_X(x, x') - \omega_Y(y, y')|.$$

Then the network ℓ_1 -distance $d_{\mathcal{N}}^1: \mathcal{N} \times \mathcal{N} \rightarrow \mathbb{R}$ is defined as $d_{\mathcal{N}}^1(\mathcal{X}, \mathcal{Y}) := \frac{1}{2} \min_R \text{dis}^1(R)$.

► **Proposition 3.** *The network ℓ_1 -distance $d_{\mathcal{N}}^1$ is a pseudometric.*



■ **Figure 2** Networks \mathcal{X} (left) and \mathcal{Y} (right) showing the inequality of Proposition 5 can be strict.

As done in [2], the standard network distance $d_{\mathcal{N}}$ can be reformulated using pairs of maps between networks. Here we define the parallel version in our alternate formulation.

► **Definition 4.** For $\mathcal{X}, \mathcal{Y} \in \mathcal{N}$ and any two maps $\varphi : X \rightarrow Y$ and $\psi : Y \rightarrow X$ on their sets of vertices, the ℓ_1 -distortion and ℓ_1 -codistortion terms are defined respectively as

$$\begin{aligned} \text{dis}^1(\varphi) &:= \sum_{x, x' \in X} |\omega_X(x, x') - \omega_Y(\varphi(x), \varphi(x'))|, \\ \text{codis}^1(\varphi, \psi) &:= \sum_{(x, y) \in X \times Y} |\omega_X(x, \psi(y)) - \omega_Y(\varphi(x), y)|. \end{aligned}$$

Then, define $d_{\mathcal{N}}^{1, \text{map}}(X, Y) := \frac{1}{2} \min_{\varphi, \psi} \{ \max\{\text{dis}^1(\varphi), \text{dis}^1(\psi), \text{codis}^1(\varphi, \psi), \text{codis}^1(\psi, \varphi)\} \}$.

Unlike the ℓ_∞ case, here we have $d_{\mathcal{N}}^{1, \text{map}} \neq d_{\mathcal{N}}^1$. However, for the purpose of stability, it suffices to use the inequality shown in Proposition 5 below.

► **Proposition 5.** Let \mathcal{X} and \mathcal{Y} be two networks. Then, $d_{\mathcal{N}}^{1, \text{map}}(\mathcal{X}, \mathcal{Y}) \leq d_{\mathcal{N}}^1(\mathcal{X}, \mathcal{Y})$.

An example where the inequality in Proposition 5 is strict is shown in Figure 2. We now want to obtain stability for the ℓ_1 -distance:

$$d_B(\text{Dgm}_k^{\text{WL}}(\mathcal{X}), \text{Dgm}_k^{\text{WL}}(\mathcal{Y})) \leq 2 d_{\mathcal{N}}^{1, \text{map}}(\mathcal{X}, \mathcal{Y}) \leq 2 d_{\mathcal{N}}^1(\mathcal{X}, \mathcal{Y}).$$

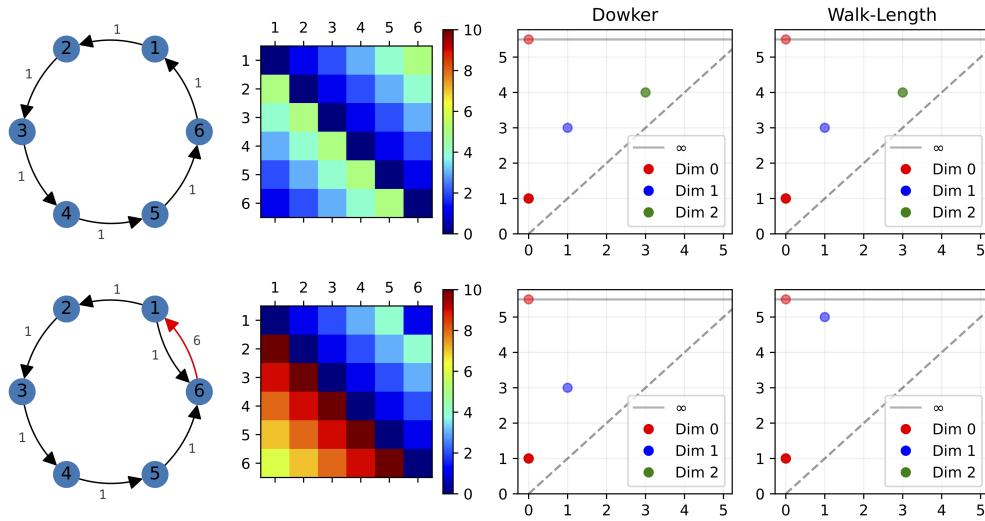
A proof for this inequality will be included in the full version of the paper.

4 Cycle Networks

Lastly, we turn our attention to a specific type of directed graph: cycle networks. For $n \geq 3$, let D_n be the cycle graph given by a full directed cycle with n vertices where all edges have weight 1. Next, we define \tilde{D}_n as a modification: Swap one of the edges, say (x_1, x_2) , to (x_2, x_1) , keeping its weight 1; to maintain strong connectivity, also add the original edge (x_1, x_2) with weight n . We define the *cycle network* (G_n, ω_{G_n}) and *semicycle network* $(\tilde{G}_n, \omega_{\tilde{G}_n})$ as the respective associated shortest-distance digraphs. We now compare results with Dowker persistence [2]. See Figure 3 for an example with 6 vertices.

For cycle networks, we have $\text{Dgm}_1^{\text{WL}}(G_n) = \text{Dgm}_1^{\mathfrak{D}}(G_n) = \{(1, \lceil n/2 \rceil) \in \mathbb{R}^2\}$. For semicycle networks, we conjecture that

$$\text{Dgm}_1^{\mathfrak{D}}(\tilde{G}_n) = \{(1, \lceil (n-1)/2 \rceil) \in \mathbb{R}^2\} \quad \text{and} \quad \text{Dgm}_1^{\text{WL}}(\tilde{G}_n) = \{(1, n-1) \in \mathbb{R}^2\}.$$




■ **Figure 3** Comparison between Dowker and walk-length filtrations. On the top, a cycle with all weights equal to 1. Then, add a vertex $(1, 6)$ with weight 1 and change weight of $(6, 1)$ to 6.

The relevance of these semicycle networks \tilde{G}_n lies in that they hint at an interesting property of the walk-length filtration: a higher sensitivity to directionality in cycles, resulting in non-directed cycles dying later than complete directed cycles.

References

- 1 Mehmet E. Aktas, Esra Akbas, and Ahmed El Fatmaoui. Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1), aug 2019. doi:10.1007/s41109-019-0179-3.
- 2 Samir Chowdhury and Facundo Mémoli. A functorial Dowker theorem and persistent homology of asymmetric networks. *Journal of Applied and Computational Topology*, 2(1):115–175, 2018. doi:10.1007/s41468-018-0020-6.
- 3 Samir Chowdhury and Facundo Mémoli. Persistent path homology of directed networks. In *Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1152–1169. ACM-SIAM, 2018. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611975031.75>, arXiv:<https://epubs.siam.org/doi/pdf/10.1137/1.9781611975031.75>, doi:10.1137/1.9781611975031.75.
- 4 Dejan Govc. Computing homotopy types of directed flag complexes, June 2020. arXiv preprint arXiv:2006.05333. arXiv:2006.05333, doi:10.48550/ARXIV.2006.05333.
- 5 David Méndez and Rubén J. Sánchez-García. A directed persistent homology theory for dissimilarity functions. *Journal of Applied and Computational Topology*, 7(4):771–813, June 2023. doi:10.1007/s41468-023-00124-x.
- 6 Michael W. Reimann, Max Nolte, Martina Scolamiero, Katharine Turner, Rodrigo Perin, Giuseppe Chindemi, Pawel Dlotko, Ran Levi, Kathryn Hess, and Henry Markram. Cliques of neurons bound into cavities provide a missing link between structure and function. *Frontiers in Computational Neuroscience*, 11:48, 2017. URL: <http://journal.frontiersin.org/article/10.3389/fncom.2017.00048>, doi:10.3389/fncom.2017.00048.
- 7 Katharine Turner. Rips filtrations for quasi-metric spaces and asymmetric functions with stability results. *Algebraic & Geometric Topology*, 19(3):1135–1170, August 2019. arXiv:1608.00365, doi:10.2140/agt.2019.19.1135.


A Symmetry-Aware Vietoris-Rips Algorithm

Jordan Matuszewski ✉ 

CUNY Graduate Center, USA

Daniel Hope ✉ 

CUNY College of Staten Island, USA

Mikael Vejdemo-Johansson ✉ 

CUNY College of Staten Island and CUNY Graduate Center, USA

Abstract

The typical topological data analysis pipeline starts with constructing a filtered complex – most often a Vietoris-Rips complex – from a given point cloud. The conventional algorithms tend not to assume any additional structure in the point cloud. We study whether a high degree of symmetry with a known symmetry group could bring improvements to the complex construction algorithms. By finding productive triage within a presentation of the symmetry group, preliminary results show we speed up the complex construction process.

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

Keywords and phrases tda, TDA4j, JavaPlex, Vietoris-Rips

Funding *Mikael Vejdemo-Johansson*: This work was supported by a grant from the Simons Foundation (961833, MVJ)

1 Introduction

We introduce a novel algorithm for faster Vietoris-Rips complex construction. Our algorithm leverages the orbit of points under the group action described by its corresponding symmetry group. These orbits frequently allow us to bypass redundant construction of complexes when creating a VR-complex, resulting in faster runtimes and requiring less memory.

2 Background

Our symmetric developments were made with a focus on point cloud persistent homology with a distance-based filtration. As such, the background provided below reflects this scope.

2.1 Topological Background

We recall that given a set V of vertices, an (abstract) *simplex* is a subset $\sigma \subset V$. A *simplicial complex* is an order ideal of the powerset of the vertices ordered by inclusion – in other words, a simplicial complex is a set of subsets of V such that if $\tau \subset \sigma$ and $\sigma \in \Sigma$ for a simplicial complex Σ , then $\tau \in \Sigma$. Finally, we recall the definition of a *Vietoris-Rips complex* at scale $t \in \mathbb{R}$ of a finite metricspace (V, d) as $VR_t(V) = \{\sigma \in 2^V \mid \text{diam } \sigma \leq t\}$, where the diameter $\text{diam } S$ is the largest distance between points in S .

This is an abstract of a presentation given at CG:YRF 2024. 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.

2.2 Symmetry Tools

The symmetry group G of a geometric object is the group of all transformations under which the object is invariant, endowed with the group operation of composition. Symmetries of geometric objects are often modeled by *group actions*.

► **Definition 1.** A group action by a group G on a set S is a group homomorphism from G into the group of automorphisms of S , or equivalently a map $G \times S \rightarrow S$. We often write the action of a group element g on a set element s as $g(s)$.

► **Definition 2.** The orbit of an element under a group action is the set of all possible images of s under the group action $\text{orbit}_G(s) = G(s) = \{g(s) | g \in G\}$.

The set of all orbits partitions the set S , and two elements of the same orbit can be considered equivalent under the equivalence relation $s \sim_G t$ if $t = g(s)$ for some $g \in G$.

If a group acts on a set S , then that group action induces a group action on the power set 2^S by defining $g(\{s_1, \dots, s_n\}) = \{g(s_1), \dots, g(s_n)\}$. Note that this construction induces a group action on simplexes from a group action on their vertices.

3 Computation

Afra Zomorodian’s incremental algorithm is a common choice for fast Vietoris-Rips complex construction [3]. We’ve adapted Zomorodian’s algorithm to leverage the full description of a group action encoding symmetries of a point cloud input. The key recognition for our improvements is that if a group G acts on point cloud X by *isometries*, when a simplex enters the VR complex at parameter value r , all simplexes in its orbit enter the complex at r .

By imposing a total order on simplexes (per Bauer and Zomorodian [2, 3]), we can describe a *smallest element* in an orbit. Speed-ups in generating and traversing VR complexes come primarily from building the capacity to recognize when a proposed next element of the complex is the smallest element in its orbit. With this discernment, we only process smallest-in-orbit elements.

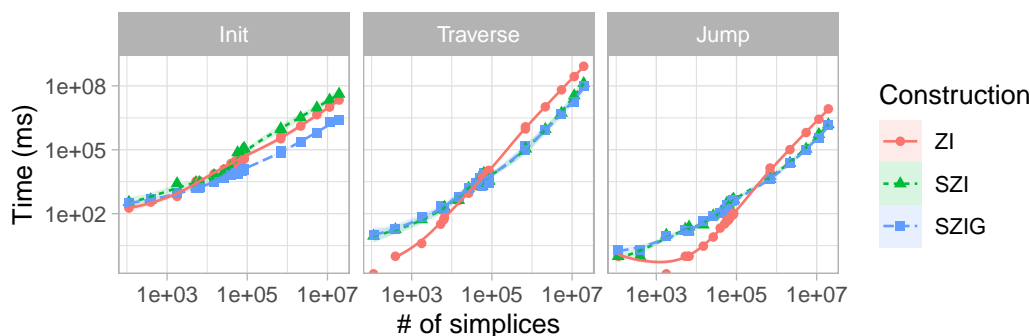
A naive approach to identifying these smallest elements is generating the entire orbit of every element inspected and checking whether it is smallest in its orbit. This approach requires a lot of memory and time. If we instead assume that the group is described by a finite presentation $G = \langle g_1, \dots, g_n | r_1, \dots, r_m \rangle$, we may define a notion of pseudo-minimality.

► **Definition 3.** An element s of a totally ordered set S with an action of the group $G = \langle g_1, \dots, g_n | r_1, \dots, r_m \rangle$ is pseudo-minimal if $s \leq g_i(s)$ and $s \leq g_i^{-1}(s)$ for all generators g_i .

It would be ideal for pseudo-minimality to imply minimality in an orbit. However, our experiments imply that this may not be the case. Even so, this approach reduces the number of elements that require the inspection of an entire orbit dramatically.

For an example of pseudo-minimality, consider the 4-bit hypercube. There is a group action by the symmetric group S_4 , acting by permuting the bits addressing each vertex. The symmetric group has one presentation in which the generators are all the adjacent transpositions: (12), (23), (34). Each of these is its own inverse.

Now, consider the simplex (3, 12, 13) – written as bit-strings this is the simplex (0011, 1100, 1101).



■ **Figure 1** Timings for creating, traversing and jumping to every 100^{th} element of the Vietoris-Rips complexes with maximum filtration value in range $[2,20]$ and maximum dimension in range $[5, 20]$.

It maps under the generators of S_4 :

$$\begin{array}{ll}
 (0011, 1100, 1101) \xrightarrow{(12)} (0011, 1100, 1110) & (3, 12, 13) \xrightarrow{(12)} (3, 12, 14) \\
 (0011, 1100, 1101) \xrightarrow{(23)} (0101, 1010, 1110) & (3, 12, 13) \xrightarrow{(23)} (5, 10, 14) \\
 (0011, 1100, 1101) \xrightarrow{(34)} (0011, 1100, 1110) & (3, 12, 13) \xrightarrow{(34)} (3, 12, 14)
 \end{array}$$

It is clear by inspection that lexicographically the images are all larger or equal. However, $(3, 12, 13) \xrightarrow{(14)(23)} (3, 7, 12)$ – in binary $(0011, 1100, 1101) \xrightarrow{(14)(23)} (0011, 0111, 1100)$ is the actual orbit-minimal representative.

3.1 Results

We can imagine a few interesting point clouds our symmetry-aware VR algorithm applies to (namely symmetric representations of sporadic groups). For now, we run timing experiments using the 4-bit and 5-bit hypercubes Q_4, Q_5 as input, since its difficult-to-compute connectivity sparked the development of our algorithm [1]. We compare the Zomorodian Incremental (ZI) algorithm with one that saves only orbit-minimal elements (SZI) and one that also uses pseudo-minimality to reject candidates for orbit-minimality (SZIG) – see Figure 1. Preliminary data suggests our symmetry-aware approach speeds up traversals and jumps between simplexes within a Vietoris-Rips complex. We are currently performing analyses to determine the asymptotic behavior of our SZI and SZIG algorithms.

References

- 1 Henry Adams and Žiga Virk. Lower Bounds on the Homology of Vietoris-Rips Complexes of Hypercube Graphs, 2023. [arXiv:2309.06222](https://arxiv.org/abs/2309.06222).
- 2 Ulrich Bauer. Ripser: Efficient Computation of Vietoris-Rips Persistence Barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.
- 3 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 347–356, 2004.

On Totally-Concave Polyominoes

Gill Barequet¹ ✉

Dept. of Computer Science, Technion—Israel Inst. of Technology, Haifa, Israel

Noga Keren ✉

Dept. of Computer Science, Technion—Israel Inst. of Technology, Haifa, Israel

Johann Peters ✉

Dept. of Mathematics, Univ. of Waterloo, ON, Canada

Adi Rivkin ✉

Dept. of Computer Science, Technion—Israel Inst. of Technology, Haifa, Israel

Abstract

A polyomino is a connected set of cells on \mathbb{Z}^2 . Every row or column of a *totally-concave* (TC) polyomino consists of more than one sequence of cells. We show that the minimum area of a TC polyomino is 21. We implement and run an algorithm for counting TC polyominoes. Finally, we prove that the associated sequence $(\kappa(n))$ has a finite growth constant λ_κ , and prove that $\lambda_\kappa > 2.4474$.

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems; Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Enumeration

Keywords and phrases Lattice animals, Growth constant

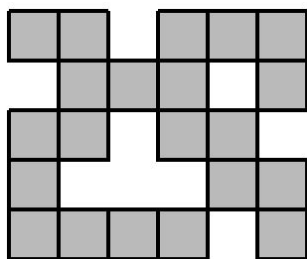
1 Introduction

A *polyomino* of area n is an edge-connected set of n cells on \mathbb{Z}^2 . Polyominoes are equivalent up to translation. Counting polyominoes is a long-standing problem [8, 9]. The sequence $A(n)$, counting polyominoes, is now known up to $n = 70$ [1]. The growth constant of polyominoes has attracted much attention. Klarner [12] showed that $\lambda := \lim_{n \rightarrow \infty} \sqrt[n]{A(n)}$ exists and is finite. Madras [13] proved that $\lim_{n \rightarrow \infty} A(n+1)/A(n) = \lambda$. It is known [4, 5] that $4.0025 \leq \lambda \leq 4.5252$.

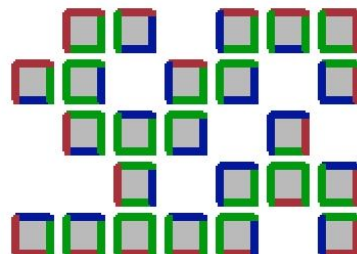
In a *totally-concave* polyomino (TCP), each row/column consists of at least *two* sequences of cells (Fig. 1). It is hinted [7, Prob. 14.5.4] that the minimum TCP has area 21. An algorithm for computing $\kappa(n)$, the number of n -cell TCPs, is also sought [ibid., Prob. 14.5.5].

This is an abstract of a presentation given at CG:YRF 2024. 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.

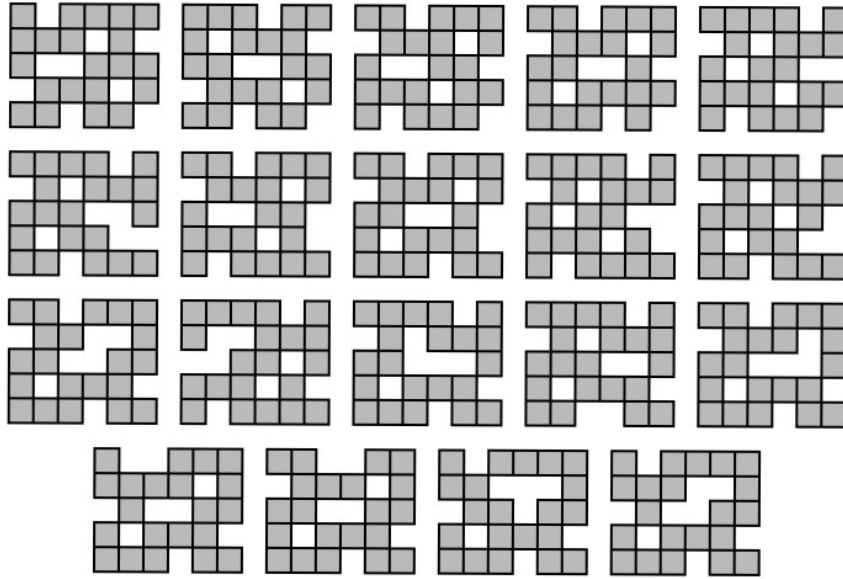


(a) Area 21



(b) Area 24 ($i = 26, o = 24, h = 46$)

■ **Figure 1** TCPs of various areas and flavors.



■ **Figure 2** The 19 TCPs of area 21, up to rotation and mirroring.

2 Minimum Area

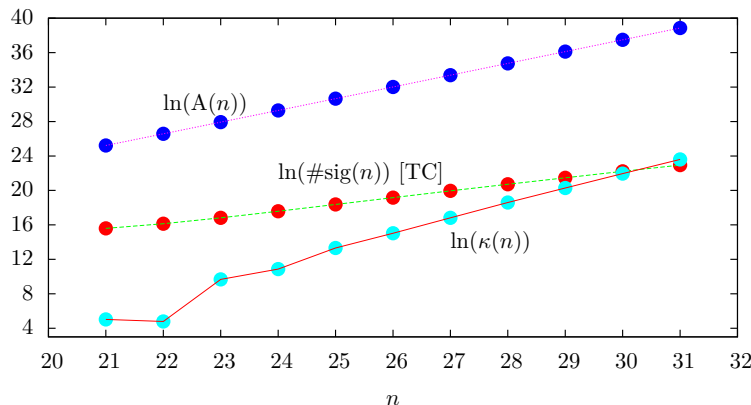
► **Theorem 1.** *The minimum area of a TCP is 21.*

Proof. Denote by n the area of a TCP having an $m \times \ell$ bounding box. A lower bound on n is achieved by classifying the edges as *hidden*, *outside*, and *inside* (Fig. 1(b)). The top (resp., right/bottom/left) edge of a cell c is *hidden* if there is an occupied cell immediately above (resp., right/below/left) c . An edge is *outside* if it is not facing any other edge. An *inside* edge is facing another edge with a gap of at least one cell. Denote by h, o, i the number of hidden, outside, and inside edges, resp. By duplicity of inside and outside edges, we have $o = 2m + 2\ell$ and $i \geq 2m + 2\ell$. We also have $h \geq 2n - 2$ since the TCP is connected. Since $h + o + i = 4n$, we have $n \geq 2m + 2\ell - 1$. For an upper bound on n , assume wlog. that $m \leq \ell$. The TCP is missing at least one cell from each of the ℓ columns, none of which in the top or bottom row, and at least two further cells, one in the top and one in the bottom row. Hence, $n \leq m\ell - \ell - 2$. Altogether, we have $2m + 2\ell - 1 \leq n \leq m\ell - \ell - 2$, with $m \leq \ell$. A case analysis shows that the smallest n satisfying these constraints is 21, with $m = 5$ and $\ell = 6$. Hence, $n \geq 21$. On the other hand, Fig. 1(a) shows TCPs of size 21. The claim follows. ◀

This result was verified by our counting programs (Section 3). See also Figure 2.

3 An Efficient Counting Algorithm

We improved Jensen's algorithm [10, 11] which operates on strips of height h ($1 \leq h \leq \frac{n}{2}$), considering column by column (left to right), and cell by cell (top to bottom) within a column. Each cell is considered as either occupied or empty. Instead of all polyominoes, the algorithm keeps in memory all possible *right boundaries*, maintaining a database whose keys are *signatures*, where a signature consists of a boundary plus all possible connections between boundary cells via cells on the left. Each entry contains statistics of partially-built polyominoes. The counts of polyominoes are updated when the next cell is chosen as occupied.



■ **Figure 3** Number of signatures (while counting TCPs), all polyominoes, and TCPs.

■ **Table 1** Counts of TCPs (values of $\kappa(n)$ for $n \leq 32$ appear in the full version of the paper).

| n | $\kappa(n)$ | n | $\kappa(n)$ | n | $\kappa(n)$ |
|-----|-----------------|-----|-------------------|-----|--------------------|
| 33 | 435,921,253,072 | 34 | 2,113,011,155,472 | 35 | 10,065,872,407,536 |

For TCPs, we ensure that columns and rows consist of more than one sequence of cells. This is easy for columns: At the end of processing a column, we discard all entries of columns that contain one or no sequence of cells. For rows, we split each signature into at most 4^h subsignatures by keeping either 0: First sequence of cells has not been met yet; 1: Inside the first sequence; 2: Between the first and second sequences; or 3: Already entered the second sequence. Then, we count only polyominoes with signatures whose line indicators are all 3.

The running time of the modified algorithm is $\tilde{O}(3.464^n)$, which is still asymptotically much less than the total number of polyominoes. Figure 3 demonstrates this phenomenon.

We implemented the modified algorithm in C++ and ran it on a 12th gen. Intel(R) i9-12900KF with 128GiB of RAM. Table 1 lists the counts obtained after 41 CPU hours.

4 Growth Constant

4.1 Existence

► **Definition 2.** (*lexicographic order*) For cells c_1, c_2 , we say that $c_1 \prec c_2$ if c_1 lies in a column which is to the left of the column of c_2 , or if c_1 lies below c_2 in the same column.

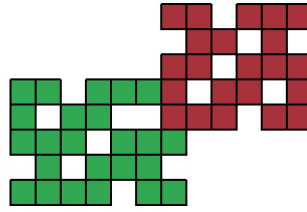
► **Definition 3.** (*concatenation*) Let P_1, P_2 be two polyominoes, and let c_1 (resp., c_2) be the largest (resp., smallest) cell of P_1 (resp., P_2). The concatenation of P_1 and P_2 is the placement of P_2 relative to P_1 , such that c_2 is found immediately on top of c_1 .

► **Theorem 4.** The limit $\lambda_\kappa := \lim_{n \rightarrow \infty} \sqrt[n]{\kappa(n)}$ exists and is finite.

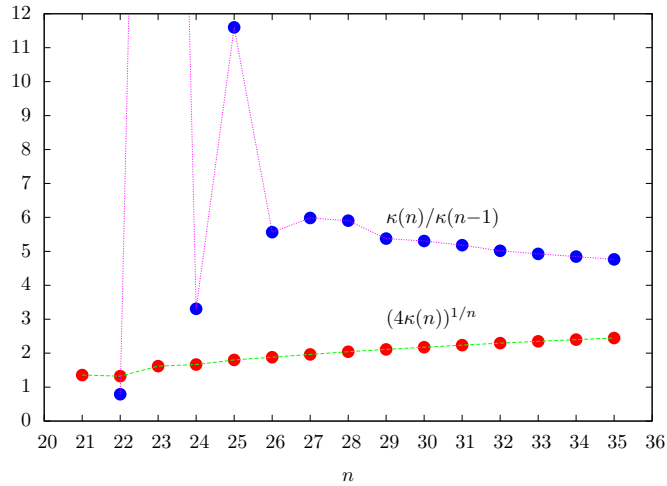
Proof. (Following [12].) First, the sequence $\kappa(n)$ is supermultiplicative, i.e., $\kappa(n)\kappa(m) \leq \kappa(n+m)$ for all $m, n \in \mathbb{N}$. Indeed, all n -cell TCPs can be concatenated with all m -cell TCPs (see Figure 4), yielding distinct $(n+m)$ -cell TCPs. Second, there exists a constant $\mu > 0$ s.t. $\kappa(n) \leq \mu^n$ for all $n \in \mathbb{N}$ (e.g., $\mu = \lambda$). By a lemma of Fekete (cf. [14]), the claim follows. ◀

Fig. 5 shows plots of the known values of $(4\kappa(n))^{1/n}$ (see Section 4.2) and $\kappa(n)/\kappa(n-1)$.

4 On Totally-Concave Polyominoes

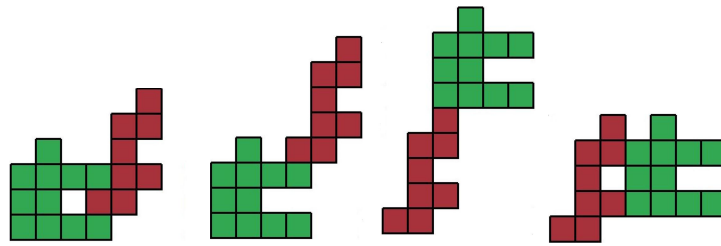


■ **Figure 4** The concatenation of two TCPs is always a TCP.



■ **Figure 5** Plots of known values of $(4\kappa(n))^{1/n}$ and $\kappa(n)/\kappa(n-1)$.

4.2 A Lower Bound on λ_κ

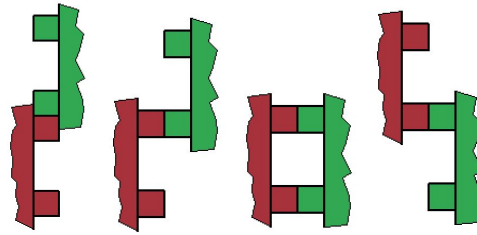


■ **Figure 6** A few compositions of a sample pair of polyominoes.

► **Definition 5.** (*composition*) A composition of two polyominoes is a placement of the two polyominoes, s.t. they touch, possibly in multiple places, but do not overlap. (See Figure 6.)

► **Theorem 6.** $\lambda_\kappa > 2.4474$.

Proof. Since the rightmost and leftmost columns of any TCP have at least two cells, there are at least four lexicographic compositions of any pair of TCPs P, Q that yield TCPs (Fig. 7). Hence, we have that $4(\kappa(n))^2 \leq \kappa(2n)$. Then, Thm. 1(a) in Ref. [2] implies that any term of the form $(4\kappa(n))^{1/n}$ is a lower bound on λ_κ . In particular, $\lambda_\kappa \geq (4\kappa(35))^{1/35} > 2.4474$. ◀



■ **Figure 7** The at least four order-preserving compositions of a pair of TCPs.

References

- 1 G. BAREQUET AND G. BEN-SHACHAR, Counting polyominoes, revisited, *SIAM Symp. on Algorithm Engineering and Experiments*, Alexandria, VA, January 2024 (to appear).
- 2 G. BAREQUET, G. BEN-SHACHAR, AND M.C. OSEGUEDA, Concatenation arguments and their applications to polyominoes and polycubes, *Computational Geometry: Theory and Applications*, 98 (2021), 12 pp.
- 3 G. BAREQUET AND M. MOFFIE, On the complexity of Jensen’s algorithm for counting fixed polyominoes, *J. of Discrete Algorithms*, 5 (2007), 348–355.
- 4 G. BAREQUET, G. ROTE, AND M. SHALAH, $\lambda > 4$: An improved lower bound on the growth constant of polyominoes, *Comm. of the ACM*, 59 (2016), 88–95.
- 5 G. BAREQUET AND M. SHALAH, Improved upper bounds on the growth constants of polyominoes and polycubes, *Algorithmica*, 84 (2022), 3559–3586.
- 6 G. BAREQUET, M. SHALAH, AND Y. ZHENG, An improved lower bound on the growth constant of polyiamonds, *J. of Combinatorial Optimization*, 37 (2019), 424–438.
- 7 G. BAREQUET, S.W. SOLOMON, AND D.A. KLARNER, Polyominoes, *Handbook of Discrete and Computational Geometry*, 3rd ed. (E. Goodman, J. O’Rourke, and C.D. Tóth, eds.), 359–380. Chapman and Hall/CRC Press, 2017.
- 8 S.R. BROADBENT AND J.M. HAMMERSLEY, Percolation processes: I. Crystals and mazes, *Proc. Cambridge Philosophical Society*, 53 (1957), 629–641.
- 9 S.W. Golomb, *Polyominoes*, Princeton University Press, Princeton, NJ, 1965 (2nd ed., 1994).
- 10 I. JENSEN, Enumerations of lattice animals and trees, *J. of Statistical Physics*, 102 (2001), 865–881.
- 11 I. JENSEN, Counting polyominoes: A parallel implementation for cluster computing, *Proc. Int. Conf. on Computational Science*, part III, Melbourne, Australia and St. Petersburg, Russia, *Lecture Notes in Computer Science*, 2659, Springer, 203–212, June 2003.
- 12 D.A. KLARNER, Cell growth problems, *Canadian J. of Mathematics*, 19 (1967), 851–863.
- 13 N. MADRAS, A pattern theorem for lattice clusters, *Annals of Combinatorics*, 3 (1999), 357–384.
- 14 G. PÓLYA AND G. SZEGŐ, Aufgaben und Lehrsätze aus der Analysis Bd. Funktionentheorie. Nullstellen. Polynome. Determinanten. Zahlentheorie (Tasks and theorems from analysis: Vol. on Function theory, zeros, polynomials, determinants, number theory), vol. 2, Springer, 1925.

The Gate-Cover Problem in Thin Polyominoes

Ariel Rosenberg ✉🏠

The Open University of Israel, Israel

Esther Arkin ✉

Stony Brook University, New York, USA

Alon Efrat ✉

University of Arizona, Arizona, USA

Omrit Filtser ✉

The Open University of Israel, Israel

Joseph S. B. Mitchell ✉

Stony Brook University, New York, USA

Abstract

We introduce a new variant of the art gallery problem, namely, the *Gate-Cover Problem in Thin Polyominoes*. We show that the VC-dimension of the problem is 3, describe an efficient greedy algorithm and present selected experimental results. We also discuss some of the open questions that we are considering in this ongoing research project.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases geometric optimization, approximation algorithms, guarding

Related Version A full version of the paper is available at <https://omrit.filtser.com/files/gate-cover-YRF.pdf>.

1 Introduction

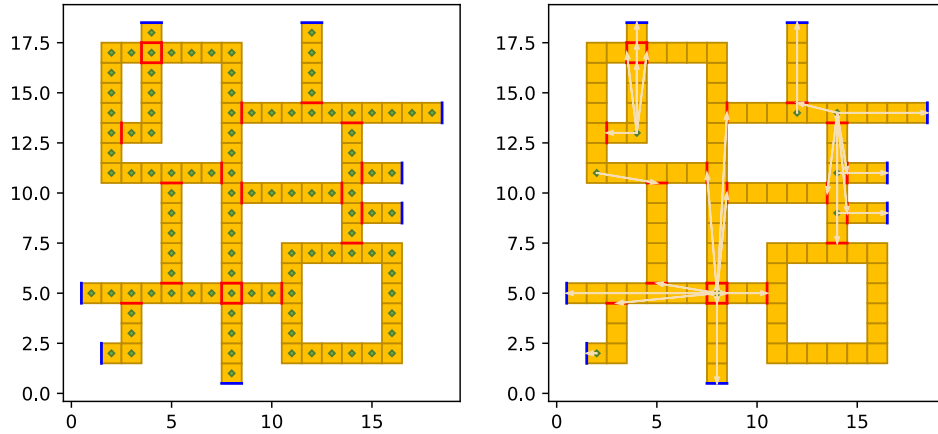
The ART GALLERY PROBLEM (AGP) is a classic problem in computational geometry [4, 5, 6]. In this problem, a polygon P is given, and the goal is to find the smallest set of points to guard P , where a point $p \in P$ is guarded by a point $g \in P$ if and only if the line segment \overline{gp} is contained in P . The motivation for AGP, as implied by its name, is placing a small set of cameras that allow detecting an intruder in an art gallery. However, sometimes guarding the entire gallery is unnecessary, and it is enough to guard a certain set of “gates” in P , so that one can output an approximate location of an intruder, by following it from the moment it enters the gallery. For example, suppose we have a city where a thief roams the streets, and we would like to delimit his location to a certain street, even without knowing his exact location. To represent the city, we use a thin polyomino P . A polyomino P is a polygon formed by joining together $|P| = n$ unit squares on the square lattice, and a polyomino is thin if it does not contain a 2×2 block of unit squares. In this analogy, a street would be a sequence of unit squares in P , such that every pair of consecutive unit squares share an edge. Thus, to be able to tell in which street the thief is located at every moment in time, we simply need to guard all the “junctions” of P , i.e., edges of squares that separates two streets, and “entrances” to the city, which are edges on the boundary of P . We therefore introduce a new variant of AGP in thin polyominoes, which we call *The Gate-Cover Problem in Thin Polyominoes*, and where the goal is to guard a set of gates in a thin polyomino P . We assume that cameras are placed on the center points of unit squares, and that a camera

This is an abstract of a presentation given at CG:YRF 2024. 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 The Gate-Cover Problem in Thin Polyominoes

c sees a gate g if and only if g belongs to a unit square visible to c in the same row/column as c . The range of the camera can be unbounded or within some hop-distance.

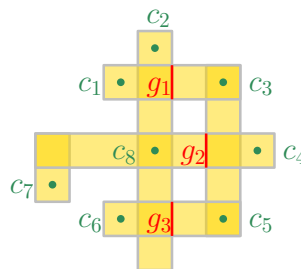
► **Problem 1** (The Gate-Cover Problem in Thin Polyominoes). *Given a thin polyomino \mathcal{P} and a set G of gates, find the minimum set of cameras that cover all gates.*



■ **Figure 1** Right: Center camera in each square. Left: Greedy solution with infinite camera.

2 VC-dimension

We show that the VC-dimension of the gate cover problem in thin polyominoes is 3. Let P be a polyomino and G a set of gates. For a camera (unit square) $c \in P$, denote by $G(c)$ the set of gates that c covers. Consider the set system (G, \mathcal{R}) where $\mathcal{R} = \{G(c) \mid c \in P\}$. The set G is shattered by \mathcal{R} if for every subset of G , there exists a camera c that covers exactly this subset. We show that there exists a set of three gates that can be shattered (Figure 2), and that no set of four gates can be shattered. Therefore the VC-dimension is 3, which implies a $O(\log \text{OPT})$ -approximation algorithm in near linear time by Brönnimann-Goodrich [1].



■ **Figure 2** A polyomino with 3 gates that can be shattered.

3 Exact algorithms

To obtain an exact algorithm, we use orthogonal line separators. Recently, separators have been used to obtain exponential time exact algorithms for various problems (see, e.g., [3]). Carmi et al. [2] show that there exists either a horizontal or vertical line on the grid that

intersect $O(\sqrt{n})$ unit squares of P , and such that on each side of it there are at most $4/5n$ unit squares of P . Given a vertical line separator ℓ , we can divide the problem into $3\sqrt{n}$ subproblems of size at most $4/5n$, as follows. For each unit square edge that lies on the separator, we have three options: (1) there is a camera above it (in the same column), (2) there is a camera below it, and (3) there is no guarantee that a camera is placed in this column. For the third type of subproblems, we do not have to update the polyomino. For the first (resp. second) type of subproblems, we add a single unit square with an entrance gate below (rep. above) the line, and remove all the gates in the column below it (resp. above it).

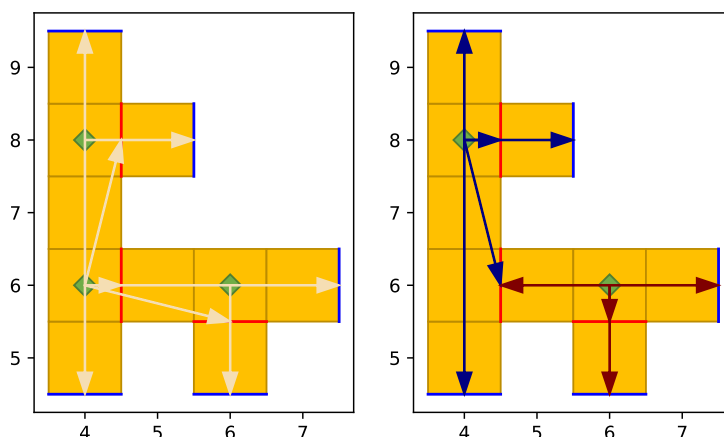
Note that for simply connected thin polyominoes (without holes), this approach gives a polynomial time algorithm, because the dual graph of the polyomino in this case is a tree, and thus there is a single vertex that separates the polyomino.

4 A greedy approximation algorithm

Let P be a thin polyomino with n unit squares, and G a set of gates. Consider the following greedy algorithm: Initialize an empty set S . While G is not empty, find a camera c that maximizes $|G(c)|$ (If there is more than one such camera, choose one arbitrarily). Add c to the set S , and remove $G(c)$ from G . The algorithm stops when all gates are covered, and thus S is a gate cover for G .

Running time. To bound the running time, we first observe that not all centers of unit squares have to be considered for placing cameras. More precisely, we can consider only “dominating” cameras (a camera c is dominating if there is no camera c' for which $G(c) \subseteq G(c')$). Dominating cameras are those placed in the unit squares that contain gates, or in “corner” unit squares that cover at least two gates (see Figure 1). Thus there are $O(|G|)$ such cameras. The running time of the algorithm is therefore $O(n + |G| \log |G|)$.

Approximation factor. In Figure 3 we show a polyomino in which the above algorithm output a set of 3 cameras, while the optimal solution has 2 cameras. In an ongoing work we formalized several claims that we hope would lead to an upper bound of $3/2$ on the approximation factor. Next, we present experimental results that support this direction.



■ **Figure 3** Right: Greedy results. Left: Brute force result - *OPT* solution.

5 Experimental results

We have constructed random “cities” with increasing size and used the following algorithms to solve the gate cover problem: (i) The greedy algorithm described above, (ii) A brute force algorithm (up to a reasonable size for running), and (iii) 0-1 Integer-Linear-Programming.

The results for unbounded-range cameras and 5-hop distance cameras are presented in the full version of this paper. Note that comparing to brute force (in small cities) and the 0 – 1 *ILP* algorithm (in larger cities), the greedy algorithm achieves close to optimal results, and always within a 3/2-factor from the optimum. In our ongoing work, we are trying to prove that 0 – 1 *ILP* is optimal by computing the dual problem.

References

- 1 Hervé Brönnimann and Michael T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discret. Comput. Geom.*, 14(4):463–479, 1995. doi:10.1007/BF02570718.
- 2 Paz Carmi, Man-Kwun Chiu, Matthew J. Katz, Matias Korman, Yoshio Okamoto, André van Renssen, Marcel Roeloffzen, Taichi Shiitada, and Shakhur Smorodinsky. Balanced line separators of unit disk graphs. In Faith Ellen, Antonina Kolokolova, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures - 15th International Symposium, WADS 2017, St. John's, NL, Canada, July 31 - August 2, 2017, Proceedings*, volume 10389 of *Lecture Notes in Computer Science*, pages 241–252. Springer, 2017. doi:10.1007/978-3-319-62127-2_21.
- 3 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for exponential-time-hypothesis-tight algorithms and lower bounds in geometric intersection graphs. *SIAM J. Comput.*, 49(6):1291–1331, 2020. doi:10.1137/20M1320870.
- 4 Joseph O’Rourke. *Art gallery theorems and algorithms*. Oxford University Press, Inc., USA, 1987.
- 5 Thomas C. Shermer. Recent results in art galleries (geometry). *Proceedings of the IEEE*, 80(9):1384–1399, 1992. doi:10.1109/5.163407.
- 6 Jorge Urrutia. Chapter 22 - art gallery and illumination problems. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 973–1027. North-Holland, Amsterdam, 2000. doi:10.1016/B978-044482537-7/50023-1.

Counting Crossing-Free Structures in 2-Page-Book Drawings

Joachim Orthaber ✉ 

Institute of Software Technology, Graz University of Technology, Austria

Javier Tejel ✉ 

Departamento de Métodos Estadísticos and IUMA, Universidad de Zaragoza, Spain

Abstract

There is a long history of counting the maximum number of crossing-free subdrawings in straight-line drawings of a graph. Inspired by the structure of most specific examples and aiming to generalize results to simple drawings, we show as a preliminary result that every 2-page-book drawing contains $\mathcal{O}((2 + \sqrt{3})^n)$ crossing-free perfect matchings and $\mathcal{O}((4 + \sqrt{17})^n)$ crossing-free Hamiltonian cycles.

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

Keywords and phrases simple drawings, counting words, plane subdrawings

Funding *Joachim Orthaber*: Supported by the Austrian Science Fund (FWF) grant W1230.

Javier Tejel: Supported by the Spanish Ministry of Science and Innovation, project PID2019-104129GB-I00 / AEI / 10.13039 / 501100011033.

Acknowledgements We are indebted to Alfredo García for contributing several ideas to this work.

1 Introduction

In a *simple drawing* of a graph vertices are represented by distinct points in the plane and edges by Jordan arcs that connect the two respective end-vertices. In addition, two edges meet in at most one point, which is a crossing or a common vertex. Ajtai, Chvátal, Newborn, and Szemerédi [2] showed in 1982 that a simple drawing of an arbitrary graph on n vertices has at most 10^{13n} different crossing-free subdrawings.

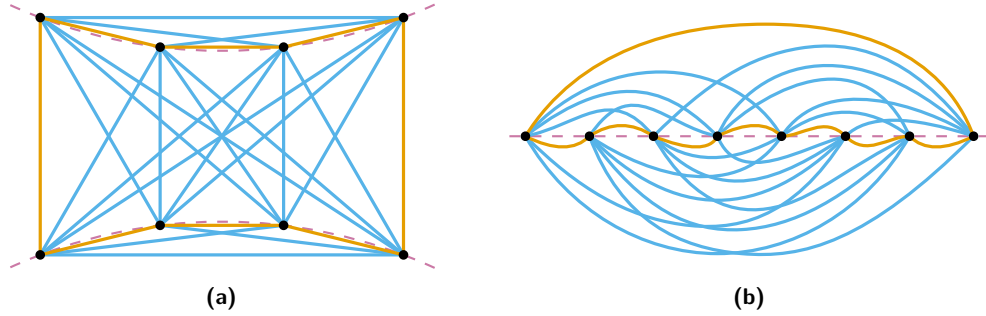
Subsequently, a lot of research has been done on *straight-line drawings*, where edges are the line segments between their end-vertices. For example, Sharir and Welzl [7] showed that such drawings contain $\mathcal{O}(10.05^n)$ different crossing-free perfect matchings and Sharir, Sheffer, and Welzl [6] proved an upper bound of $\mathcal{O}(54.55^n)$ for the number of different crossing-free Hamiltonian cycles. In the other direction, García, Noy, and Tejel [4] introduced double chain drawings (see Figure 1(a) for an example) and proved that they contain $\Omega(4.64^n)$ crossing-free Hamiltonian cycles. Asinowski and Rote [3] analyzed generalizations of these drawings and showed that they contain $\Omega(3.09^n)$ different crossing-free perfect matchings.

Recently, Rutschmann and Wettstein [5] observed in their paper on triangulations that most straight-line drawings known to contain many crossing-free subdrawings follow a specific pattern, which they describe in their definition of a *chain*. Ultimately a chain induces a Hamiltonian cycle of completely uncrossed edges. This leads us to *2-page-book drawings*. They are simple drawings where the vertices lie on a horizontal line, which defines the two pages, and each edge lies either above or below this line; see Figure 1(b) for an example. Equivalently, a simple drawing of the complete graph K_n is a 2-page-book drawing if and

This is an abstract of a presentation given at CG:YRF 2024. 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.

only if it contains a Hamiltonian cycle of completely uncrossed edges. For such drawings we can show upper bounds that lie significantly below the bounds for straight-line drawings.

► **Theorem 1.** *Every 2-page-book drawing on n vertices contains $\mathcal{O}((2 + \sqrt{3})^n)$ different crossing-free perfect matchings and $\mathcal{O}((4 + \sqrt{17})^n)$ different crossing-free Hamiltonian cycles.*



■ **Figure 1** (a) A double chain drawing of K_8 . (b) A crossing minimal 2-page-book drawing of K_8 . The Hamiltonian cycle of completely uncrossed edges is drawn orange in both examples.

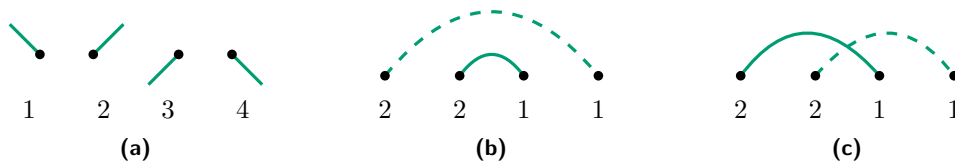
2 Proof sketch of Theorem 1

Let \mathcal{D} be a 2-page-book drawing on n vertices and let us first consider the case of a perfect matching \mathcal{M} in \mathcal{D} . Every vertex v in \mathcal{D} is incident to exactly one edge $e = \{v, w\}$ in \mathcal{M} and w lies either to the *left* or to the *right* of v . Furthermore, every edge in \mathcal{D} lies either on the *top* or the *bottom* page. Therefore we get the following.

► **Lemma 2.** *In a 2-page-book drawing \mathcal{D} on n vertices every crossing-free perfect matching \mathcal{M} corresponds to a unique word of length n over the alphabet $\{1, 2, 3, 4\}$.*

Proof. For each vertex in \mathcal{D} we encode its incident edge in \mathcal{M} going to the *top-left* by 1, *top-right* by 2, *bottom-left* by 3, and *bottom-right* by 4; see Figure 2(a). Then \mathcal{M} clearly gets mapped to a word of length n over the alphabet $\{1, 2, 3, 4\}$.

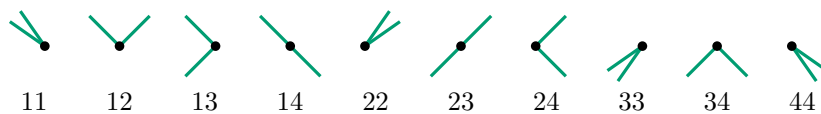
Let \mathcal{W} be a word of length n over the alphabet $\{1, 2, 3, 4\}$. We show that at most one crossing-free perfect matching in \mathcal{D} gets mapped to \mathcal{W} . Reading \mathcal{W} from left to right, every 2 or 4 starts an edge and every 1 or 3 ends an edge of a potential perfect matching. Since the matching needs to be crossing-free, every time we encounter a 1 or 3, we must connect its corresponding vertex w to the closest vertex v to its left that got assigned a 2 or 4, respectively, and that we did not yet connect; see Figure 2(b). Otherwise we would leave some vertex between v and w that later would have to cross $\{v, w\}$; see Figure 2(c). Consequently, the crossing-free perfect matching corresponding to \mathcal{W} , if it exists, is unique. ◀



■ **Figure 2** (a) The 4 letters corresponding to a vertex in a perfect matching. (b) A 1 has to be connected to the closest free 2 to its left so that (c) no later edge (dashed) will cause a crossing.

At this point we would get an upper bound of 4^n for the maximum number of crossing-free perfect matchings in \mathcal{D} . However, for edges between consecutive vertices, which are orange in Figure 1(b), it does not matter whether they are drawn on the top or bottom page because they are completely uncrossed anyway. Hence, we can assume, without loss of generality, that they are all drawn on the top page. Consequently, a word corresponding to a crossing-free perfect matching cannot contain 4-3 as a subword.

Let $f(n)$ be the number of words of length n over the alphabet $\{1, 2, 3, 4\}$ not containing 4-3 as a subword. Then we get the recurrence $f(n) = 4f(n-1) - f(n-2)$ because we can extend all allowed words of length $n-1$ by one of the four letters as a prefix and then subtract all words that start with the subword 4-3 but are fine afterwards. Solving this recurrence leads to the claimed asymptotic bound of $\mathcal{O}((2 + \sqrt{3})^n)$.



■ **Figure 3** The 10 different letters corresponding to a vertex in a Hamiltonian cycle.

For the case of a crossing-free Hamiltonian cycle we can argue similarly. We give the main ideas in the following. Since each vertex is incident to exactly two edges, each crossing-free Hamiltonian cycle corresponds to a word over the alphabet $\{11, 12, 13, 14, 22, 23, 24, 33, 34, 44\}$ of 10 letters; see Figure 3. Assuming again, without loss of generality, that all edges between consecutive vertices are drawn on the top page, we can exclude 16 subwords of length 2 in this case. In addition, the subword 22-11 is not allowed for $n > 2$ because it would form a cycle of length 2. Some of those 17 forbidden subwords can interact with each other. However, if $g(n)$ is the number of allowed words of length n , a careful analysis leads to the recurrence $g(n) = 9g(n-1) - 7g(n-2) - g(n-3)$ and consequently to the claimed asymptotic bound of $\mathcal{O}((4 + \sqrt{17})^n)$.

3 Discussion of the result

The bounds in this submission are ongoing work. We are currently working on improving them and have several ideas on how to also exclude subwords of length greater than 2. Also note that these bounds do not only hold for straight-line chain constructions but also, for example, for crossing minimal 2-page-book drawings (like the one in Figure 1(b)), which have significantly fewer crossings than any straight-line drawing of K_n can have [1].

References

- 1 Bernardo M. Ábrego, Oswin Aichholzer, Silvia Fernández-Merchant, Pedro Ramos, and Gelasio Salazar. The 2-page crossing number of K_n . *Discrete & Computational Geometry*, 49(4):747–777, 2013. doi:10.1007/s00454-013-9514-0.
- 2 Miklós Ajtai, Vašek Chvátal, Monroe M. Newborn, and Endre Szemerédi. Crossing-free subgraphs. *Annals of Discrete Mathematics*, 12:9–12, 1982. doi:10.1016/S0304-0208(08)73484-4.
- 3 Andrei Asinowski and Günter Rote. Point sets with many non-crossing perfect matchings. *Computational Geometry*, 68:7–33, 2018. doi:10.1016/j.comgeo.2017.05.006.
- 4 Alfredo García, Marc Noy, and Javier Tejel. Lower bounds on the number of crossing-free subgraphs of K_N . *Computational Geometry*, 16(4):211–221, 2000. doi:10.1016/S0925-7721(00)00010-9.

4 Counting Crossing-Free Structures in 2-Page-Book Drawings

- 5 Daniel Rutschmann and Manuel Wettstein. Chains, Koch chains, and point sets with many triangulations. *Journal of the ACM*, 70(3):18:1–18:26, 2023. doi:10.1145/3585535.
- 6 Micha Sharir, Adam Sheffer, and Emo Welzl. Counting plane graphs: Perfect matchings, spanning cycles, and Kasteleyn’s technique. *Journal of Combinatorial Theory, Series A*, 120(4):777–794, 2013. doi:10.1016/j.jcta.2013.01.002.
- 7 Micha Sharir and Emo Welzl. On the number of crossing-free matchings, cycles, and partitions. *SIAM Journal on Computing*, 36(3):695–720, 2006. doi:10.1137/050636036.

Comparison of Graph Distance Measures

Maike Buchin ✉

Ruhr University Bochum

Lea Thiel ✉

Ruhr University Bochum

Abstract

In order to compare embedded graphs, many different graph distance measures have been defined. To better understand and use these, we study how several of these measures relate to each other.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Fréchet distance, Graph comparison, Embedded graphs

1 Introduction

For embedded graphs, there are many measures to determine their similarity. Buchin et al. [5] survey several of the measures and which properties of a metric they fulfil. Many of these measures have already been analysed with regard to their properties and their computational complexity, but little attention has been paid to the interrelationships between the different measures. However, in order to decide on a suitable graph measure for an application, it is also interesting to know how these relate to each other. We therefore examine how several of these measures are related, in particular which of these measures are "stronger" in the sense that the distance under a weaker measure is always smaller than that of the stronger one.

In this paper, we consider undirected graphs embedded in \mathbb{R}^2 . Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two such undirected graphs with vertices embedded as points in \mathbb{R}^2 that are connected by crossing free straight-line edges. In the following we first define the distance measures we are interested in, which are the strong and weak graph distance, traversal distance, and contour tree distance. We are interested in these particular measures, because they all take into account both the geometry and the topology of the graph, and are based on similar notions of mapping/matching parts of the graphs. In the next section, we then show relations between these, some of which hold only on paths or trees.

The *strong* resp. *weak graph distance* is based on the (strong resp. weak) Fréchet distance [3]. The idea of this graph distance is to map one of the graphs onto a subgraph of the other that is as similar as possible. For this, a mapping $s: G \rightarrow H$ is called a *graph mapping* if it maps each vertex $v \in V_G$ to a point $s(v)$ on an edge of H , and it maps each edge $\{u, v\} \in E_G$ to an arbitrary simple path from $s(u)$ to $s(v)$ in the embedding of H .

► **Definition 1.** We define the directed [weak] graph distance $\vec{\delta}_{[w]G}$ as

$$\vec{\delta}_{[w]G}(G, H) = \inf_{s: G \rightarrow H} \max_{e \in E_G} \delta_{[w]F}(e, s(e)),$$

where s ranges over graph mappings from G to H , $\delta_{[w]F}$ denotes the [weak] Fréchet distance, and e and its image $s(e)$ are interpreted as curves in the plane. The undirected graph distances are $\delta_{[w]G}(G, H) = \max(\vec{\delta}_{[w]G}(G, H), \vec{\delta}_{[w]G}(H, G))$.

This is an abstract of a presentation given at CG:YRF 2024. 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 Comparison of Graph Distance Measures

The directed graph distances are not symmetric. For example, if G is a subgraph of H , the graph distance from G to H is zero, while the distance from H to G can be arbitrarily large.

Another measure defined by Alt et al. in [2] is the *traversal distance*. It is a natural extension of the weak Fréchet distance, in which two paths are traversed simultaneously. Depending on the application, it can be useful to traverse only one graph completely and only a part of the second (directed traversal distance).

► **Definition 2.** We call a mapping $g: [0, 1] \rightarrow G$ that is surjective and continuous, a traversal of G . A continuous (but not necessarily surjective) mapping $h: [0, 1] \rightarrow G$ is called partial traversal of G . We define the directed traversal distance as

$$\vec{\delta}_T(G, H) = \inf_{g, h} \max_{t \in [0, 1]} \|g(t) - h(t)\|,$$

where g ranges over all traversals of G and h over all partial traversals of H .

There are two different ways to define a symmetric variant of the directed traversal distance. One is to take the maximum of both directed values, as defined in [5], the *symmetric traversal distance* $\delta_T(G, H) = \max\{\vec{\delta}_T(G, H), \vec{\delta}_T(H, G)\}$. We introduce another – arguably more natural – symmetric variant, in which both graphs are traversed completely.

► **Definition 3.** We define the complete traversal distance as

$$\delta_{cT}(G, H) = \inf_{g, h} \max_{t \in [0, 1]} \|g(t) - h(t)\|,$$

where g and h range over all traversals of G and H , respectively.

The *contour tree distance* was motivated by the computation of the Fréchet distance of surfaces and introduced by Buchin, Ophelders and Speckmann [4]. Later it was naturally extended by [5] to a distance on embedded graphs. In it connected areas of the two graphs are matched. For this, we first define two classes of matchings between G and H . A matching $\tau \in M(G, H)$ has the following properties:

1. τ is a connected subset of $G \times H$,
2. $\tau(x)$ is a nonempty, connected subset of H for each $x \in G$, and
3. $\tau^{-1}(y)$ is a nonempty, connected subset of G for each $y \in H$.

We introduce the class of weak matchings $wM(G, H)$, where $\tau(x)$ and $\tau^{-1}(y)$ do not have to be connected. Then the [weak] contour tree distance is defined as the largest distance between two matched points in an optimal [weak] matching.

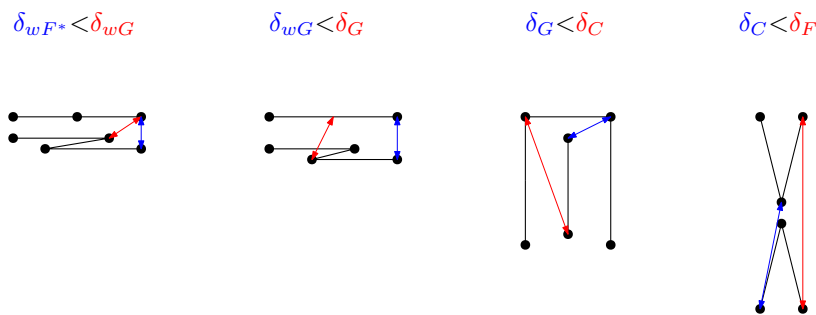
► **Definition 4.** We define the [weak] contour tree distance $d_{[w]C}$ as

$$\delta_{[w]C}(G, H) = \inf_{\tau \in [w]M(G, H)} \sup_{(x, y) \in \tau} \|x - y\|_2,$$

where $\|x - y\|_2$ is the Euclidean distance between the embeddings of x and y .

2 Comparison of the Graph Distance Measures

So far it is known for paths that $\delta_{wF^*} \leq \delta_{wG} \leq \delta_G \leq \delta_F$, where δ_{wF^*} denotes the weak Fréchet distance without boundary restriction [1]. It was also argued for graphs that $\vec{\delta}_T \leq \vec{\delta}_{wG} \leq \vec{\delta}_G$. Here, we show the following relationships, the proofs of which can be found in the full version.



■ **Figure 1** Example of paths where the graph distance measures differ.

► **Theorem 5.** *Let P and Q be paths. Then*

$$\delta_{wF^*} = \delta_T = \delta_{cT} = \delta_{wC} \leq \delta_{wG} \leq \delta_G \leq \delta_C \leq \delta_F.$$

In Figure 1 example paths are given where the graph distances differ for all inequalities.

► **Theorem 6.** *Let G and H be embedded trees, then $\delta_{wG} \leq \delta_G \leq \delta_C$.*

► **Theorem 7.** *Let G and H be embedded graphs, then $\delta_{cT} = \delta_{wC}$.*

3 Conclusion and open questions

We have shown a strict order for the considered distance measures on paths, some of which can be transferred to trees and graphs. An open question is how the two different symmetric variants of the traversal distance relate to each other. Obviously, $\delta_T \leq \delta_{cT}$, since every complete traversal of both graphs is also a traversal of one graph and a partial traversal of the other graph. But it is not yet clear whether they are equivalent. Furthermore, the question of how the graph distance and contour tree distance relate to each other on arbitrary graphs is still open, since the proof for trees uses the uniqueness of paths between matched points.

References

- 1 Hugo A. Akitaya, Maike Buchin, Bernhard Kilgus, Stef Sijben, and Carola Wenk. Distance measures for embedded graphs. *Computational Geometry*, 95:101743, 2021. doi:10.1016/j.comgeo.2020.101743.
- 2 Helmut Alt, Alon Efrat, Günter Rote, and Carola Wenk. Matching planar maps. *Journal of Algorithms*, 49(2):262–283, 2003. doi:10.1016/S0196-6774(03)00085-3.
- 3 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(1-2):75–91, 1995.
- 4 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*, SODA '17, page 2443–2455, USA, 2017.
- 5 Maike Buchin, Erin Chambers, Pan Fang, Brittany Terese Fasy, Ellen Gasparovic, Elizabeth Munch, and Carola Wenk. Distances between immersed graphs: Metric properties. *La Matematica 2*, (1):197–222, 2023. doi:10.1007/s44007-022-00037-8.


A New Separator Theorem for Unit Disk Graphs

Elfarouk Harb ✉

Department of Computer Science, University of Illinois at Urbana-Champaign, USA

Zhengcheng Huang ✉

Department of Computer Science, University of Illinois at Urbana-Champaign, USA

Da Wei Zheng ✉ 

Department of Computer Science, University of Illinois at Urbana-Champaign, USA

Abstract

We introduce a new balanced separator theorem for unit-disk graphs involving two shortest paths combined with the 1-hop neighbours of those paths and two other vertices. This answers an open problem of Yan, Xiang and Dragan [CGTA '12] and improves their result that requires removing the 3-hop neighbourhood of two shortest paths. Our proof uses very different ideas, including Delaunay triangulations and a generalization of the celebrated balanced separator theorem of Lipton and Tarjan [J. Appl. Math. '79] to systems of non-intersecting paths.

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

Keywords and phrases Balanced shortest path separators, unit disk graphs, crossings

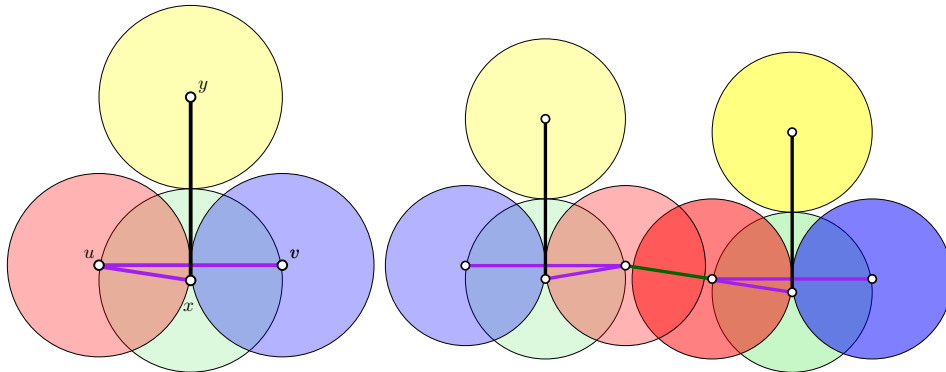
1 Introduction

In graphs, a *separator* is a small set of vertices whose removal splits the graph into smaller components. Separators are very useful for designing divide-and-conquer algorithms. Planar graphs are well-known for admitting good separators. The first separator theorem for planar graphs was due to Lipton and Tarjan [13], who proved that every planar graph on n vertices admits a separator of size $O(\sqrt{n})$ that can be computed in $O(n)$ time. Since then, many variants of separator theorems have been proven for planar graphs [13, 14, 7, 9, 17, 12, 4]. Some of these results can be naturally extended to graphs with bounded genus [6, 8] or to minor-free graphs [2, 10, 18].

Separator theorems for geometric intersection graphs A *geometric intersection graph* is an undirected graph where each vertex corresponds to a geometric object, and edges indicate which pairs of objects intersect each other. One common type of geometric intersection graph is the unit disk graph, which arises in modeling wireless communication. Such graphs also appear in applications such as VLSI design. These graphs have been extensively studied in the computational geometry community. It is natural to ask whether separator theorems apply for geometric intersection graphs. For unit disk graphs, many different separators exist, such as line separators [3] and clique separators [5]. When the unit disks have low *ply* (the number of disks intersecting any given point), good separators are also known to exist [15, 16].

Shortest path separators The usefulness of a separator does not necessarily solely depend on the number of vertices. An excellent example of good separators with large size are *shortest path separators*, *i.e.* separators consisting of a constant number of shortest paths.

This is an abstract of a presentation given at CG:YRF 2024. 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 (Left)** The points $u, v, x, y \in S$ are drawn with circles of radius $1/2$. The unique shortest path tree in G with starting vertex u has a crossing edge. **(Right)** Two reflected copies results in a graph where no planar shortest path tree exists, regardless of starting vertex.

Thorup showed that every planar graph admits a separator consisting of at most two shortest paths [17]. Similar results have been proven for minor-free graphs [1]. Shortest path separators have been used extensively in distance-related problems in planar graphs, such as distance oracles [17, 11] and planar emulators [4].

A natural question is whether shortest path separator theorems can be adapted to unit disk graphs. Naively, such separators cannot exist, as the clique on n vertices is realizable as a unit disk graph for which no such separator can exist. However, we can strengthen the separator by also removing vertices in the k -neighborhood of the shortest path, *i.e.* vertices that are at a distance of at most k from the shortest path. Yan, Xiang and Dragan [19] proved that every unit disk graph admits a shortest path 3-neighborhood separator, that is, by removing two shortest paths and all vertices in the 3-hop neighborhood of any vertex on the shortest path, the remaining graph is disconnected with every component having size at most $2/3$ of the vertices of the original graph. They left open the question of whether there exists a shortest path 1-neighborhood separator.

Our results We answer the open question of Yan, Xiang and Dragan [19] in the affirmative. We show that every graph has a 1-neighborhood separator. In particular, it suffices to only remove the 1-neighborhood of two shortest paths plus the 1-neighborhood of two other vertices. While the proof of Yan, Xiang, and Dragan manipulates crossings in the intersection graph, our proof uses very different ideas involving paths in Delaunay triangulations and a generalization of the shortest path separators of Lipton and Tarjan to sets of *weakly non-crossing paths* in a triangulated planar graph that may be of independent interest.

► **Theorem 1.** *Every unit disk graph admits a shortest path 1-neighborhood separator.*

2 An initial approach and sketch of techniques

To illustrate the difficulty in obtaining our result let us consider one approach to construct shortest path 1-neighborhood separators for a unit disk graph $G = (V, E)$. To do so, we will make two overly wishful assumptions (that would be great if they always held).

Let \mathcal{T} be a shortest path tree of G starting at a fixed vertex $s \in V$. We will wishfully assume that \mathcal{T} is planar (assumption 1); that is, no two edges of \mathcal{T} cross. Next, we will assume that we can triangulate \mathcal{T} to get a graph $G_{\mathcal{T}}$ (assumption 2) such that every edge of

the triangulated graph $G_{\mathcal{T}}$ is an edge in G . Now, we can use the shortest path separator theorem of Lipton and Tarjan [13], on $G_{\mathcal{T}}$ with spanning tree \mathcal{T} to get a Jordan curve C that is a separator for $G_{\mathcal{T}}$. This is because all edges $uv \in E$ have the property that for all other edges crossing the line segment between u and v the crossing edge has at least one end point adjacent to either u or v (we call this property *cross-dominating*). Thus the cycle C is in fact also a shortest path 1-neighborhood separator of G . We address these assumptions:

1. Our first assumption was that we could find a shortest path tree \mathcal{T} of G that is planar. This is not always the case, there are examples (see Figure 1) of unit disk graphs G where no planar shortest path tree exists. Instead, we will construct a non-crossing path system Π consisting of pseudo-shortest paths, i.e. for every vertex $u \in V$ we will find a path $\Pi[u]$ to s such that $\Pi[u]$ consists only of vertices on the shortest path between u and s , as well as 1-neighbors of the shortest path. We show an extension of the planar separator algorithm to find a balanced separator in path systems of planar graphs.
2. Our second assumption was that we could triangulate the tree \mathcal{T} to get a graph $G_{\mathcal{T}}$ such that every edge of the triangulation is an edge in G . We show that instead, we can prove all edges of the Delaunay triangulation has the cross-dominating property as well, and furthermore we show that we can construct Π using only edges of the Delaunay triangulation of the centers of the disks of G .

References

- 1 Ittai Abraham and Cyril Gavoille. Object location using path separators. In Eric Ruppert and Dahlia Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 188–197. ACM, 2006. [PATHdoi:10.1145/1146381.1146411](https://doi.org/10.1145/1146381.1146411).
- 2 Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for nonplanar graphs. *J. Amer. Math. Soc.*, 3(4):801–808, 1990. [PATHdoi:10.2307/1990903](https://doi.org/10.2307/1990903).
- 3 Paz Carmi, Man-Kwun Chiu, Matthew J. Katz, Matias Korman, Yoshio Okamoto, André van Renssen, Marcel Roeloffzen, Taichi Shiitada, and Shakhar Smorodinsky. Balanced line separators of unit disk graphs. *Comput. Geom.*, 86, 2020. [PATHdoi:10.1016/j.comgeo.2019.101575](https://doi.org/10.1016/j.comgeo.2019.101575).
- 4 Hsien-Chih Chang, Robert Krauthgamer, and Zihan Tan. Almost-linear ϵ -emulators for planar graphs. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1311–1324. ACM, 2022. [PATHdoi:10.1145/3519935.3519998](https://doi.org/10.1145/3519935.3519998).
- 5 Mark de Berg, Sándor Kisfaludi-Bak, Morteza Monemizadeh, and Leonidas Theocharous. Clique-based separators for geometric intersection graphs. *Algorithmica*, 85(6):1652–1678, 2023. URL: <https://doi.org/10.1007/s00453-022-01041-8>, [PATHdoi:10.1007/S00453-022-01041-8](https://doi.org/10.1007/S00453-022-01041-8).
- 6 Hristo N Djidjev. A separator theorem. *Compt. Rend. Acad. Bulg. Sci.*, 34(5):643–645, 1981.
- 7 Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.*, 16(6):1004–1022, 1987. [PATHdoi:10.1137/0216064](https://doi.org/10.1137/0216064).
- 8 John R. Gilbert, Joan P. Hutchinson, and Robert Endre Tarjan. A separator theorem for graphs of bounded genus. *J. Algorithms*, 5(3):391–407, 1984. [PATHdoi:10.1016/0196-6774\(84\)90019-1](https://doi.org/10.1016/0196-6774(84)90019-1).
- 9 Michael T. Goodrich. Planar separators and parallel polygon triangulation. *J. Comput. Syst. Sci.*, 51(3):374–389, 1995. URL: <https://doi.org/10.1006/jcss.1995.1076>, [PATHdoi:10.1006/JCSS.1995.1076](https://doi.org/10.1006/JCSS.1995.1076).
- 10 Ken-ichi Kawarabayashi and Bruce A. Reed. A separator theorem in minor-closed classes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 153–162. IEEE Computer Society, 2010. [PATHdoi:10.1109/FOCS.2010.22](https://doi.org/10.1109/FOCS.2010.22).

- 11 Philip N. Klein. Preprocessing an undirected planar network to enable fast approximate distance queries. In David Eppstein, editor, *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA*, pages 820–827. ACM/SIAM, 2002. URL: <http://dl.acm.org/citation.cfm?id=545381.545488>.
- 12 Philip N. Klein, Shay Mozes, and Christian Sommer. Structured recursive separator decompositions for planar graphs in linear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 505–514. ACM, 2013. PATHdoi:10.1145/2488608.2488672.
- 13 Richard J Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.
- 14 Gary L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *J. Comput. Syst. Sci.*, 32(3):265–279, 1986. PATHdoi:10.1016/0022-0000(86)90030-9.
- 15 Gary L. Miller, Shang-Hua Teng, William P. Thurston, and Stephen A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. ACM*, 44(1):1–29, 1997. PATHdoi:10.1145/256292.256294.
- 16 Warren D. Smith and Nicholas C. Wormald. Geometric separator theorems & applications. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 232–243. IEEE Computer Society, 1998. PATHdoi:10.1109/SFCS.1998.743449.
- 17 Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM*, 51(6):993–1024, 2004. PATHdoi:10.1145/1039488.1039493.
- 18 Christian Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 37–46. IEEE Computer Society, 2011. PATHdoi:10.1109/FOCS.2011.15.
- 19 Chenyu Yan, Yang Xiang, and Feodor F. Dragan. Compact and low delay routing labeling scheme for unit disk graphs. *Comput. Geom.*, 45(7):305–325, 2012. URL: <https://doi.org/10.1016/j.comgeo.2012.01.015>, PATHdoi:10.1016/J.COMGEO.2012.01.015.

Extending Hausdorff Distances to Asymmetric Geometries

Tuyen Pham ✉

University of Florida, Gainesville, US

Hubert Wagner ✉

University of Florida, Gainesville, US

Abstract

We propose an extension of the Hausdorff distance from metric spaces to spaces equipped with asymmetric distances, such as the Bregman divergences. This includes the popular Kullback–Leibler divergence (relative entropy). As an experiment we use the new divergence to compare collections of predictions returned by different machine learning models trained using relative entropy.

2012 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems; Theory of computation → Computational geometry; Mathematics of computing → Combinatorial algorithms

Keywords and phrases Bregman divergence, KL divergence, relative entropy, cross entropy, Shannon entropy, Hausdorff distance, Bregman–Hausdorff divergence, Asymmetric Hausdorff

Funding 2022 Google Research Scholar Award in Algorithms and Optimization.

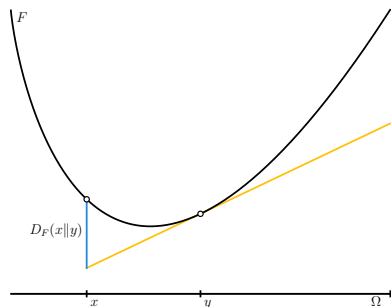
1 Introduction

Given a metric space (Ω, d) , the Hausdorff distance, $d_H(P, Q)$, between two sets $P, Q \subset \Omega$ is

$$\inf \left\{ r \geq 0 : P \subset \bigcup_{p \in P} B(p, r) \text{ and } Q \subset \bigcup_{q \in Q} B(q, r) \right\},$$

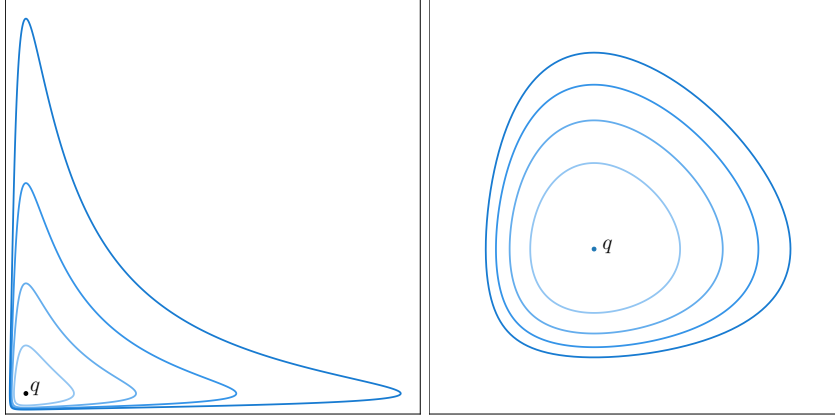
where $B(x, r) = \{\omega \in \Omega : d(x, \omega) \leq r\}$ is the ball of radius r centered at x .

The above defines the distance between these two sets and has been used in many computational applications such as computer vision [3]. We generalize this definition to spaces equipped with asymmetric distances, in particular with Bregman divergences.



■ **Figure 1** Visualization of a Bregman divergence construction for a one-dimensional domain.

This is an abstract of a presentation given at CG:YRF 2024. 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 2** Left: primal Itakura–Saito balls. Right: primal generalized Kullback–Leibler balls.

2 Background on Bregman divergences

Let $\Omega \subset \mathbb{R}^d$ be an open convex set. A **function of Legendre type** [8] is a function $F : \Omega \rightarrow \mathbb{R}$ that is differentiable and strictly convex. We additionally require that $\lim_{x \rightarrow \partial\Omega} \|\nabla F(x)\| = \infty$, provided $\partial\Omega$ is nonempty.

Given a function F of Legendre type, the **Bregman divergence generated** [1] by F is a function $D_F : \Omega \times \Omega \rightarrow \mathbb{R}$. The divergence between x and y is the difference between $F(x)$ and the best affine approximation of F at y evaluated at x , or simply

$$D_F(x||y) = F(x) - (F(y) + \langle \nabla F(y), x - y \rangle). \quad (1)$$

See Figure 1 for an illustration. These divergences are often asymmetric and we refer to $D_F(x||y)$ as the divergence in the **direction** from x to y .

Prominent examples of Bregman divergences are the squared Euclidean, Kullback–Leibler (KL) [5], and the Itakura–Saito (IS) [4] divergences. The KL and IS divergences are information measures connected with Shannon entropy and Burg entropy respectively, and both have seen success as loss functions to be minimized in machine learning [2, 6].

Bregman balls. Due to the asymmetry, one can define two types of Bregman balls [7]. We start with the **primal Bregman ball** of radius $r \geq 0$ centered at q which is defined as

$$B_F(q; r) = \{y \in \Omega : D_F(q||y) \leq r\}. \quad (2)$$

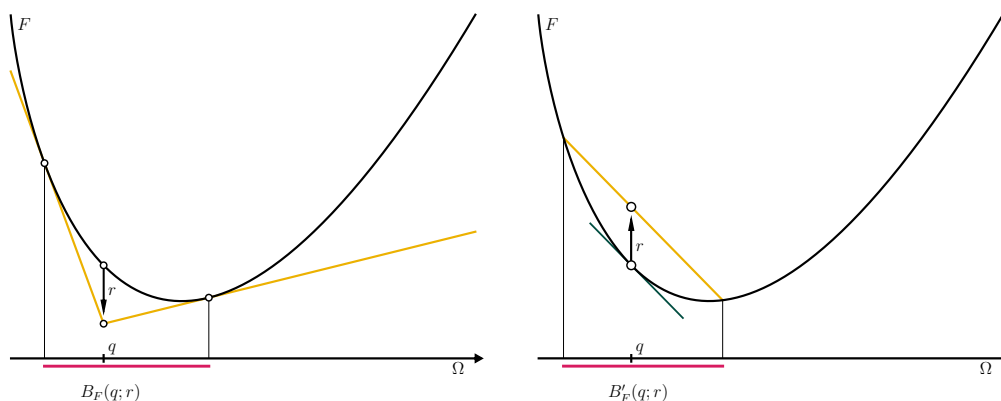
Namely, it is the collection of points with Bregman divergence measured *from* the center not exceeding r . See Figure 2 for illustrations. Primal Bregman balls have a particularly nice geometric interpretation: given a light source at point $(q, F(q) - r)$, the primal ball $B_F(q; r)$ is the *illuminated* part of the graph of F projected vertically onto Ω .

The **dual Bregman ball** of radius $r \geq 0$ centered at q is defined as

$$B'_F(q; r) = \{y \in \Omega : D_F(y||q) \leq r\}. \quad (3)$$

Visualized geometrically, we first shift the tangent plane at $(q, F(q))$ up by r . The dual Bregman ball $B'_F(q; r)$ is the portion of the graph of F below this plane projected vertically onto Ω . Both geometric interpretations are illustrated in Figure 3.

Bregman–Hausdorff divergence. Let us extend the metric definition of the Hausdorff distance to the setting of Bregman divergences. Given $P, Q \subset \Omega$ and a Bregman generator



■ **Figure 3** Geometric interpretation of primal (left) and dual (right) Bregman balls in dimension 1.

F , we define the **primal Bregman–Hausdorff divergence** as

$$H_F(P\|Q) = \inf \left\{ r \geq 0 : Q \subset \bigcup_{p \in P} B_F(p, r) \right\} \quad (4)$$

and the **dual Bregman–Hausdorff divergence** as

$$H'_F(P\|Q) = \inf \left\{ r \geq 0 : P \subset \bigcup_{q \in Q} B'_F(q, r) \right\}, \quad (5)$$

naming them by the type of the balls involved. Despite being a natural extension, the above definitions appear to be new.

As explained in detail in ??, the KL–Hausdorff has an information-theoretical interpretation and is expressed in bits. We are especially interested in comparing the predictions on training and test data arising from two models of different quality. We were also curious about a comparison with the IS–Hausdorff divergence.

3 Experiments

Models and data sets. We train two neural networks, M_1 and M_2 , on a classification task on the CIFAR100 dataset, achieving 80.22%, and 71.74% test accuracy respectively. From each model, we produce two sets of predictions: $(\text{trn}_i, \text{tst}_i)$, for $i \in \{1, 2\}$. The networks are trained to minimize the total KL divergence, and the predictions live in the standard simplex of dimension 99 equipped with the KL divergence. For comparison, we also generate dataset Δ_u randomly distributed on this standard simplex.

Analysis. Table 1 confirms that the KL–Hausdorff divergences capture that M_1 is a better model. Technically a larger loss of coding efficiency is incurred if M_1 were to be approximated by M_2 . Similarly, a much larger loss is incurred if the uniform distribution is used to approximate tst_1 . More generally, the large gap for $H_{IS}(\text{tst}_1\|\Delta_u)$ shows that the choice of loss in training significantly effects the resulting geometry. In contrast, as the standard simplex is bounded under the Euclidean distance but unbounded under the other divergences, the standard Hausdorff distances are all close.

■ **Table 1** Bregman Hausdorff divergences.

| | $(\text{tst}_1 \parallel \text{trn}_2)$ | $(\text{tst}_2 \parallel \text{trn}_1)$ | $(\text{trn}_2 \parallel \text{trn}_1)$ | $(\text{trn}_1 \parallel \text{trn}_2)$ | $(\text{tst}_1 \parallel \Delta_u)$ |
|-------------|---|---|---|---|-------------------------------------|
| H_{KL} | 4.03b | 4.52b | 4.34b | 4.03b | 22.65b |
| H_{IS} | 1,144.24 | 1,998.38 | 2,360.05 | 1,309.63 | 63.77 |
| H'_{IS} | 34,008.59 | 1,739,646,377.75 | 14,801,113.43 | 584,772.40 | 364.21 |
| H_{SqEuc} | 0.26 | 0.24 | 0.23 | 0.27 | 11.84 |

4 Summary

In application to machine learning, the Bregman–Hausdorff divergence is an interpretable measure for comparing model predictions. Prompted by the above experiments, we ask how the choice of divergence affects the estimate of intrinsic dimension of data. We also mention natural usage in measuring the level of noise introduced in the input, by computing the Bregman–Hausdorff divergence between the original and perturbed predictions. The new definition also opens up interesting algorithmic questions.

References

- 1 Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. doi:10.1016/0041-5553(67)90040-7.
- 2 Minh N Do and Martin Vetterli. Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Transactions on Image Processing*, 11(2):146–158, 2002.
- 3 Daniel P. Huttenlocher, Gregory A. Klanderman, and William Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:850–863, 1993. URL: <https://api.semanticscholar.org/CorpusID:8027136>.
- 4 Fumitada Itakura. Analysis synthesis telephony based on the maximum likelihood method. *Reports of the 6th Int. Cong. Acoust.*, 1968.
- 5 Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- 6 Kevin P. Murphy. *Machine learning: A Probabilistic Perspective*. MIT press, 2012.
- 7 Richard Nock and Frank Nielsen. Fitting the Smallest Enclosing Bregman Ball. pages 649–656, 10 2005. doi:10.1007/11564096_65.
- 8 Ralph T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. doi:10.1515/9781400873173.