Noname manuscript No.
(will be inserted by the editor)

Interleaving Distance between Merge Trees

Dmitriy Morozov · Kenes Beketayev · Gunther H. Weber

the date of receipt and acceptance should be inserted later

Abstract Merge trees are topological descriptors of scalar functions. They record how the subsets of the domain where the function value does not exceed a given threshold are connected. We define a distance between merge trees, called an interleaving distance, and prove the stability of these trees, with respect to this distance, to perturbations of the functions that define them. We show that the interleaving distance is never smaller than the bottleneck distance between persistence diagrams.

1 Introduction

Topological data analysis is a young field at the intersection of computational geometry and algebraic topology. It interprets data as functions on topological spaces, detects their salient features, and summarizes their connectivity. The resulting topological descriptors serve many purposes. Some of them allow the user to segment the data into interesting regions. For example, Morse–Smale complexes partition the domain of a scalar function into regions with uniform gradient flow. Others help with rapid exploration of the data set; Reeb graphs let the user quickly label and extract connected components of level sets of a function. Yet others, such as persistence diagrams, present the user with a complete overview of the data, helping her make decisions about the magnitude of noise and recognize significant scales in the data. In all cases, it is crucial for the descriptor to be stable. Stability is the most basic test of robustness: if we perturb the data a little, can the descriptor change a lot? To be reliable, it must not.

In this paper, we are concerned with a specific topological descriptor. One of the basic structures in computational topology, a *merge tree* keeps track

Dmitriy Morozov[1] · Kenes Beketayev[1,2] · Gunther H. Weber[1,2]
[1]Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720
[2]University of California, Davis, 1 Shields Avenue, Davis, CA 95616
E-mail: dmitriy@mrzv.org, KBeketayev@lbl.gov, GHWeber@lbl.gov

of the evolution of connected components in the sublevel sets of a function. It records how new components appear at minima and merge at saddles. To even approach the question of stability in the previous paragraph, we must first define a distance between two trees. We call our definition the *interleaving distance*.

Its introduction has a dual effect. First of all, it lets us prove stability of merge trees with respect to this distance — our main goal. But as important is the resulting transformation of the space of merge trees into a metric space. This construction makes it possible to use merge trees as proxies for function comparison. Often such direct comparison is either too difficult, or too sensitive. For example, directly comparing height functions on two shapes would first require computing a homeomorphism between the shapes that best aligns the two functions, a notoriously difficult proposition. On the other hand, extracting two merge trees is simple and fast.

Despite how significant stability is to topological data analysis, its study has been limited — no proofs exist for most descriptors. The work most closely related to ours is the proof of stability of persistence diagrams [4,1]. In this context, besides purely mathematical developments [3], stability lets us track changes in persistence diagrams of continuously varying functions [5] as well as encourages the use of persistence diagrams as stable signatures of shapes [2], in the spirit of the previous paragraph.

Outline. We define the interleaving distance in Section 3 and check that it is a metric. Theorem 2 in Section 4 ensures that this distance is stable. Theorem 3 in the following section relates interleaving distance to the bottleneck distance between persistence diagrams. It is a quality check: merge trees capture more information than 0–dimensional persistence diagrams, therefore, a distance on merge trees should be more discriminating than the distance on persistence diagrams.

2 Background

We start with a scalar function $f : \mathbb{X} \to \mathbb{R}$, defined on a connected domain $\mathbb{X}$. We say that two points x and y in its domain are equivalent, $x \sim y$, if they belong to the same component of the levelset $f^{-1}(f(x)) = f^{-1}(f(y))$. The quotient space with respect to this equivalence relation, $\mathbb{X}/\!\sim$, is called a *Reeb graph* of f. Informally, it is a continuous contraction of the contours of function f.

Merge trees. An epigraph of the function, denoted by $\operatorname{epi} f$, is the set of points above its graph: $\operatorname{epi} f = \{(x, y) \in \mathbb{X} \times \mathbb{R} \mid y \geq f(x)\}$. We denote the projection from the epigraph onto the range of f by $\bar{f} : \operatorname{epi} f \to \mathbb{R}$; $\bar{f}((x, y)) = y$. Notice that if we project the level sets of $\bar{f}$ back into the domain of our function, we get the sublevel sets of f, which we denote by $F_a = f^{-1}(-\infty, a] = \pi_{\mathbb{X}}(\bar{f}^{-1}(a))$. The Reeb graph of function $\bar{f}$, denoted by T_f, is called the *merge tree* of

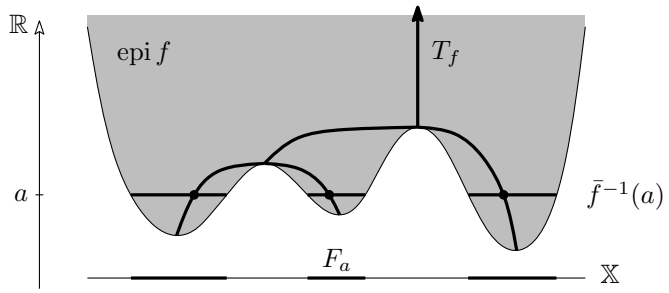

Fig. 1 A graph of function $f : \mathbb{X} \to \mathbb{R}$ together with its merge tree, T_f. The three components of a levelset of the projection $\bar{f} : \operatorname{epi} f \to \mathbb{R}$ are highlighted in bold together with the points of the merge tree that represent them. This levelset projects onto the sublevel set, F_a, highlighted inside the domain, $\mathbb{X}$.

function f; see Figure 1. Intuitively, it keeps track of the evolution of connected components in the sublevel sets of f. A component appears at a minimum and grows until it merges with another component at a saddle. We note that according to our definition, a merge tree extends to infinity. This formulation differs from what usually appears in literature, where the root of the merge tree is taken to be the global maximum of the function. This distinction is minor, but useful to us for technical reasons that will become clear in the next section.

Since the points identified by the equivalence relation in the definition of a merge tree belong to the same level sets of $\bar{f}$, they have the same function value. Therefore, there is a well-defined map $\hat{f} : T_f \to \mathbb{R}$ from the merge tree to the range of $\bar{f}$ — it is the unique map that satisfies $\bar{f} = \hat{f} \circ q$, where $q : \operatorname{epi} f \to T_f$ is defined by $q(x) = y$, where y is the component of the level set $\bar{f}^{-1}(\bar{f}(x))$ that contains x.

We denote by $i^\varepsilon : T_f \to T_f$ the *ε-shift map* in the tree T_f. To define it, recall that $x \in T_f$, with $\hat{f}(x) = a$, represents a connected component X in the sublevel set F_a of function f. The inclusion of sublevel sets $F_a \subseteq F_{a+\varepsilon}$ maps X into a connected component Y of $F_{a+\varepsilon}$. Let y represent this component in the tree T_f. Then $i^\varepsilon(x) = y$. In other words, to find the image of x under i^ε, we simply follow the path from x to the root of T_f until we encounter a point y with $\hat{f}(y) = a + \varepsilon$.

Persistent homology. A 0–dimensional homology group of a space Y, denoted by $\mathsf{H}_0(Y)$, is a group of formal sums of connected components of Y. For simplicity, consider coefficients in $\mathbb{Z}_2$. In this case, an element of $\mathsf{H}_0(Y)$ is a set of connected components of Y; the group operation is the symmetric difference of sets. If space Y is a subset of some space Z, $Y \subseteq Z$, then the inclusion of spaces maps connected components of Y into connected components of Z, and so induces a map between homology groups, $\iota : \mathsf{H}_0(Y) \to \mathsf{H}_0(Z)$.

Given a function $f : \mathbb{X} \to \mathbb{R}$, we can track the evolution of homology groups of its sublevel sets, F_a. We get a sequence of groups, $\mathsf{H}_0(F_a)$, connected by

homomorphisms $\iota_a^b : \mathsf{H}_0(F_a) \to \mathsf{H}_0(F_b)$ induced by the inclusions $F_a \subseteq F_b$, where $a \leq b$. A connected component x is born in this sequence at $\mathsf{H}_0(F_b)$ when it is not in the image of the inclusions from preceding sublevel sets: $x \notin \iota_a^b(\mathsf{H}_0(F_a))$ for all $a < b$. This component dies at $\mathsf{H}_0(F_d)$ if it is in the image of a homology group preceding $\mathsf{H}_0(F_b)$, $\iota_b^d(x) \in \iota_a^d(F_a)$ for some $a < b$, but $\iota_b^c(x) \notin \iota_a^c(F_a)$ for any $b < c < d$.

The collection of all such birth–death pairs (b, d), together with all the points (a, a) on the diagonal taken with infinite multiplicity, is called 0–*dimensional persistence diagram* and is denoted by $\mathrm{Dgm}_0(f)$. A fundamental property of persistence diagrams is their stability. To express it, we need the notion of a bottleneck distance.

Definition 1. *The* bottleneck distance *between two multi-sets of points X and Y is*

$$\mathrm{d_B}(X, Y) = \inf_{\gamma} \sup_{x} \|x - \gamma(x)\|_\infty,$$

where γ goes over all possible bijections between X and Y, and $\|x-\gamma(x)\|_\infty = \max\{|b_x - b_y|, |d_x - d_y|\}$ if $x = (b_x, d_x)$ and $\gamma(x) = (b_y, d_y)$.

Stability was originally proved by Cohen-Steiner et al. [4] for two functions defined on the same domain. Over the years their result was strengthened. In Section 5, we will need the following formulation of the stability theorem for persistence diagrams, which is simplified from the statement due to Chazal et al. [1]. To state it, we need an additional notion of tameness. In our case, it simply means that all the sublevel sets of a function have a finite number of connected components.

Definition 2. *A function $f : \mathbb{X} \to \mathbb{R}$ is called* tame *if the dimension of the 0-dimensional homology group of its every sublevel set is finite, $\dim \mathsf{H}_0(F_a) < \infty$ for all $a \in \mathbb{R}$. In this case, we also call the sequence of homology groups, $\mathsf{H}_0(F_a)$, tame.*

Theorem 1. *Two sequences of homology groups, $\mathsf{H}_0(F_a)$ and $\mathsf{H}_0(G_a)$, are* ε-interleaved *if there are maps*

$$\phi^a : \mathsf{H}_0(F_a) \to \mathsf{H}_0(G_{a+\varepsilon})$$
$$\psi^a : \mathsf{H}_0(G_a) \to \mathsf{H}_0(F_{a+\varepsilon})$$

such that their compositions commute with the maps $\lambda_a^b : \mathsf{H}_0(F_a) \to \mathsf{H}_0(F_b)$ and $\kappa_a^b : \mathsf{H}_0(G_a) \to \mathsf{H}_0(G_b)$ induced by inclusions.

Given two tame sequences of homology groups, $\mathsf{H}_0(F_a)$ and $\mathsf{H}_0(G_a)$, we denote their persistence diagrams by $\mathrm{Dgm}_0(F)$ and $\mathrm{Dgm}_0(G)$. If the sequences are ε-interleaved, then the bottleneck distance between the diagrams does not exceed ε:

$$\mathrm{d_B}(\mathrm{Dgm}_0(F), \mathrm{Dgm}_0(G)) \leq \varepsilon.$$

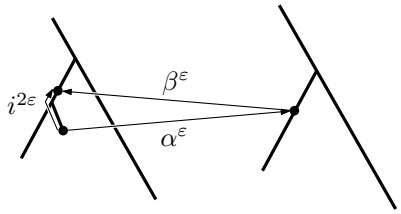

Fig. 2 Compatible maps between two trees.

3 Interleaving Distance

To define the central object of our paper, suppose that we have two merge trees, T_f and T_g, with the corresponding maps $\hat{f} : T_f \to \mathbb{R}$ and $\hat{g} : T_g \to \mathbb{R}$. We begin with an auxiliary notion of ε-compatible maps.

Definition 3. *Two continuous maps $\alpha^\varepsilon : T_f \to T_g$ and $\beta^\varepsilon : T_g \to T_f$ are said to be* ε-compatible*, for some $\varepsilon \geq 0$, if*

$$\hat{g}(\alpha^\varepsilon(x)) = \hat{f}(x) + \varepsilon, \qquad \hat{f}(\beta^\varepsilon(y)) = \hat{g}(y) + \varepsilon,$$
$$\beta^\varepsilon \circ \alpha^\varepsilon = i^{2\varepsilon}, \qquad \alpha^\varepsilon \circ \beta^\varepsilon = j^{2\varepsilon},$$

where $i^{2\varepsilon} : T_f \to T_f$ and $j^{2\varepsilon} : T_g \to T_g$ are the 2ε-shift maps in the respective trees.

In other words, two maps are ε-compatible if they commute with the shift maps in the respective trees. We note that since maps α^ε and β^ε are continuous, the conditions for ε-compatibility extend to the following relations for all $a \geq 0$:

$$\beta^\varepsilon \circ j^a \circ \alpha^\varepsilon = i^{a+2\varepsilon}, \qquad \alpha^\varepsilon \circ i^a \circ \beta^\varepsilon = j^{a+2\varepsilon},$$
$$j^a \circ \alpha^\varepsilon = \alpha^\varepsilon \circ i^a, \qquad i^a \circ \beta^\varepsilon = \beta^\varepsilon \circ j^a.$$

The interleaving distance finds the best ε-compatible maps.

Definition 4. *The* interleaving distance*, $\mathrm{d_I}(T_f, T_g)$, between two merge trees, T_f and T_g, is the greatest lower bound on ε for which there are ε-compatible maps:*

$$\mathrm{d_I}(T_f, T_g) = \inf\{\varepsilon \mid \textit{there are } \varepsilon\textit{-compatible maps } \alpha^\varepsilon : T_f \to T_g, \beta^\varepsilon : T_g \to T_f\}.$$

It is not difficult, but still worthwhile, to verify that the interleaving distance is a metric on the space of merge trees.

Lemma 1 (Metric)**.** *The interleaving distance, $\mathrm{d_I}$, is a metric. In other words, it satisfies the following properties:*

1. $\mathrm{d_I}(T, T) = 0$;
2. $\mathrm{d_I}(T_1, T_2) = \mathrm{d_I}(T_2, T_1)$;
3. $\mathrm{d_I}(T_1, T_3) \leq \mathrm{d_I}(T_1, T_2) + \mathrm{d_I}(T_2, T_3)$.

Proof. The first property is immediate if we take maps α^0 and β^0 to be the identity on tree T. The second property follows from the symmetry of the definition of the interleaving distance.

To show the third property, suppose $\mathrm{d_I}(T_1, T_2) = \varepsilon_1$. Then, for all $\delta > 0$, there are $(\varepsilon_1 + \delta)$-compatible maps, $\alpha_{12}^{\varepsilon_1+\delta} : T_1 \to T_2$ and $\beta_{21}^{\varepsilon_1+\delta} : T_2 \to T_1$. Similarly, suppose $\mathrm{d_I}(T_2, T_3) = \varepsilon_2$. Then, for all $\delta > 0$, there are $(\varepsilon_2 + \delta)$-compatible maps, $\alpha_{23}^{\varepsilon_2+\delta} : T_2 \to T_3$, $\beta_{32}^{\varepsilon_2+\delta} : T_3 \to T_2$. Denote by $i_1^a : T_1 \to T_1, i_2^a : T_2 \to T_2$, and $i_3^a : T_3 \to T_3$ the a-shift maps in the respective trees.

Given $\delta > 0$, let $\varepsilon_3 = \varepsilon_1 + \varepsilon_2$ and define $\alpha_{13}^{\varepsilon_3+\delta} : T_1 \to T_3$ and $\beta_{31}^{\varepsilon_3+\delta} : T_3 \to T_1$ as the compositions:

$$\alpha_{13}^{\varepsilon_3+\delta} = \alpha_{23}^{\varepsilon_2+\delta/2} \circ \alpha_{12}^{\varepsilon_1+\delta/2},$$
$$\beta_{31}^{\varepsilon_3+\delta} = \beta_{21}^{\varepsilon_1+\delta/2} \circ \beta_{32}^{\varepsilon_2+\delta/2}.$$

These two maps are $(\varepsilon_3 + \delta)$-compatible since

$$\begin{aligned} i_1^{2(\varepsilon_3+\delta)} &= i_1^{2(\varepsilon_1+\varepsilon_2+\delta)} \\ &= \beta_{21}^{\varepsilon_1+\delta/2} \circ i_2^{2(\varepsilon_2+\delta/2)} \circ \alpha_{12}^{\varepsilon_1+\delta/2} \\ &= \beta_{21}^{\varepsilon_1+\delta/2} \circ \beta_{32}^{\varepsilon_2+\delta/2} \circ \alpha_{23}^{\varepsilon_2+\delta/2} \circ \alpha_{12}^{\varepsilon_1+\delta/2} \\ &= \beta_{31}^{\varepsilon_3+\delta} \circ \alpha_{13}^{\varepsilon_3+\delta}. \end{aligned}$$

Similarly, $i_3^{2(\varepsilon_3+\delta)} = \alpha_{13}^{\varepsilon_3+\delta} \circ \beta_{31}^{\varepsilon_3+\delta}$.

Therefore, since the statements hold for all $\delta > 0$, $\mathrm{d_I}(T_1, T_3) \leq \varepsilon_3 = \mathrm{d_I}(T_1, T_2) + \mathrm{d_I}(T_2, T_3)$. □

4 Stability

To be a reliable descriptor, merge trees must be stable: if we change a function a little, its tree should only change a little. We show that this is indeed true if we compare trees using the interleaving distance.

Theorem 2 (Stability)**.** *Given two scalar functions $f, g : \mathbb{X} \to \mathbb{R}$, let T_f and T_g denote their merge trees. The interleaving distance between the trees does not exceed the largest difference between the two functions:*

$$\mathrm{d_I}(T_f, T_g) \leq \sup_x |f(x) - g(x)|.$$

Proof. Let $\varepsilon = \sup_x |f(x) - g(x)|$ be the largest difference between the two functions. Recall that $F_a = f^{-1}(-\infty, a]$ and $G_b = g^{-1}(-\infty, b]$ denote sublevel sets of these functions. Since the largest difference between the functions is ε, their sublevel sets include into each other:

$$F_a \subseteq G_{a+\varepsilon} \subseteq F_{a+2\varepsilon}.$$

These inclusions induce maps between the merge trees. A point x in the merge tree T_f with $\hat{f}(x) = a$ corresponds to a component in sublevel set F_a.

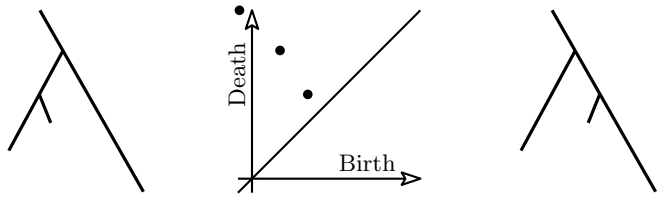

Fig. 3 The interleaving distance between the two trees in the figure is positive — it is equal to half the size of the smallest branch — but the corresponding functions have identical persistence diagrams.

The inclusion $F_a \subseteq G_{a+\varepsilon}$ maps this component to a component in sublevel set $G_{a+\varepsilon}$; let point $y \in T_g$ represent this component in the merge tree of g. Thus the inclusion of the sublevel sets induces a map $\alpha^\varepsilon : T_f \to T_g$, defined via the above construction as $\alpha^\varepsilon(x) = y$. Conversely, we have a map $\beta^\varepsilon : T_g \to T_f$. By construction, if $\hat{f}(x) = a$, then $\hat{g}(\alpha^\varepsilon(x)) = a + \varepsilon$, and vice versa, if $\hat{g}(y) = a$, then $\hat{f}(\beta^\varepsilon(y)) = a + \varepsilon$.

The inclusion of the sublevel sets of a single function produces the shift maps, defined in Section 2. The inclusion $F_a \subseteq F_{a+2\varepsilon}$ induces a map $i^{2\varepsilon} : T_f \to T_f$ that maps a point $x \in T_f$ with $\hat{f}(x) = a$ into its ancestor $y \in T_f$ with $\hat{f}(y) = a + 2\varepsilon$. Similarly, we have a shift map $j^{2\varepsilon} : T_g \to T_g$. Since the maps $\alpha^\varepsilon, \beta^\varepsilon, i^{2\varepsilon}$, and $j^{2\varepsilon}$ are induced by inclusions, they commute:

$$\beta^\varepsilon \circ \alpha^\varepsilon = i^{2\varepsilon} \qquad\qquad \alpha^\varepsilon \circ \beta^\varepsilon = j^{2\varepsilon}.$$

Therefore, by definition, α^ε and β^ε are ε-compatible, and the interleaving distance does not exceed ε, $\mathrm{d_I}(T_f, T_g) \leq \varepsilon$. □

5 Bottleneck Distance between Persistence Diagrams

It is not difficult to construct an example where the bottleneck distance between 0–dimensional persistence diagrams is arbitrarily smaller than the interleaving distance between merge trees; see Figure 3. The main result of this section, stated in Theorem 3, shows that the former can never be larger than the latter.

Theorem 3. *Given two tame functions, $f : \mathbb{X} \to \mathbb{R}$ and $g : \mathbb{Y} \to \mathbb{R}$, the bottleneck distance between their persistence diagrams does not exceed the interleaving distance between their merge trees:*

$$\mathrm{d_B}(\mathrm{Dgm}_0(f), \mathrm{Dgm}_0(g)) \leq \mathrm{d_I}(T_f, T_g).$$

Proof. First of all, notice that the 0–dimensional persistence diagram of the function $f : \mathbb{X} \to \mathbb{R}$ is the same as the persistence diagram of the function $\hat{f} : T_f \to \mathbb{R}$; $\mathrm{Dgm}_0(f) = \mathrm{Dgm}_0(\hat{f})$. This fact follows immediately from the definition of merge trees: collapsing components of sublevel sets to points does not change the 0–dimensional homology groups.

Accordingly, we need to show that $\mathrm{d_B}(\mathrm{Dgm}_0(\hat{f}), \mathrm{Dgm}_0(\hat{g})) \leq \mathrm{d_I}(T_f, T_g)$. Let $\hat{F}_a = \hat{f}^{-1}(-\infty, a]$ and $\hat{G}_a = \hat{g}^{-1}(-\infty, a]$ denote the sublevel sets of the functions on merge trees. Let $\varepsilon = \mathrm{d_I}(T_f, T_g)$. Then, by definition of the interleaving distance, for all $\delta > 0$, there are two maps $\alpha^{\varepsilon+\delta} : T_f \to T_g$ and $\beta^{\varepsilon+\delta} : T_g \to T_f$ that commute with the shift maps. It follows that the two sequences of homology groups, $\mathsf{H}_0(\hat{F}_a)$ and $\mathsf{H}_0(\hat{G}_a)$, are $(\varepsilon + \delta)$-interleaved in the sense of Theorem 1. Therefore, by the same theorem, their persistence diagrams are close, $\mathrm{d_B}(\mathrm{Dgm}_0(\hat{f}), \mathrm{Dgm}_0(\hat{g})) \leq \varepsilon + \delta$. Since the last statement is true for all $\delta > 0$, we have $\mathrm{d_B}(\mathrm{Dgm}_0(\hat{f}), \mathrm{Dgm}_0(\hat{g})) \leq \varepsilon$, and our theorem's claim follows. □

6 Conclusion

In this paper, we have defined an interleaving distance between merge trees. We have proved that this metric is no less sensitive than the bottleneck distance between 0–dimensional persistence diagrams, yet it is still stable to perturbations of the function.

It is not difficult to devise an exponential-time algorithm to find this distance given two merge trees. To do so, one can take advantage of the continuity of the ε-compatible maps in Definition 3. Accordingly, to check existence of ε-compatible maps for a fixed ε, it suffices to try all possible maps on the leaves of the two trees (each leaf has only a finite set of targets, if the trees are finite), extend the corresponding ε-compatible maps continuously and verify their consistency on the saddles.

The next logical step towards using interleaving distance as a metric in applications is to devise an efficient algorithm that calculates it.

Acknowledgements This work was supported by the Director, Office of Science, Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 through the grant "Topology-based Visualization and Analysis of High-dimensional Data and Time-varying Data at the Extreme Scale," program manager Lucy Nowell.

References

1. Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Annual Symposium on Computational Geometry*, pages 237–246, 2009.
2. Frédéric Chazal, David Cohen-Steiner, Leonidas Guibas, Facundo Mémoli, and Steve Oudot. Gromov–hausdorff stable signatures for shapes using persistence. In *Computer Graphics Forum*, volume 28, pages 1393–1403, 2009. Special issue 6th Annual Symposium on Geometry Processing.
3. David Cohen-Steiner and Herbert Edelsbrunner. Inequalities for the curvature of curves and surfaces. *Foundations of Computational Mathematics*, 7:391–404, 2007.
4. David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37:103–120, 2007.

5. David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Annual Symposium on Computational Geometry*, pages 119–126, 2006.