Scalar Field Comparison with Topological Descriptors: Properties and Applications for Scientific Visualization

Lin Yan[1], Talha Bin Masood[2], Raghavendra Sridharamurthy[3], Farhan Rasheed[2], Vijay Natarajan[3], Ingrid Hotz[2], Bei Wang[1]
[1]Scientific Computing and Imaging Institute, University of Utah, USA
[2]Department of Science and Technology (ITN), Linköping University, Norrköping, Sweden
[3]Department of Computer Science and Automation, Indian Institute of Science Bangalore, India

Abstract

In topological data analysis and visualization, topological descriptors such as persistence diagrams, merge trees, contour trees, Reeb graphs, and Morse–Smale complexes play an essential role in capturing the shape of scalar field data. We present a state-of-the-art report on scalar field comparison using topological descriptors. We provide a taxonomy of existing approaches based on visualization tasks associated with three categories of data: single fields, time-varying fields, and ensembles. These tasks include symmetry detection, periodicity detection, key event/feature detection, feature tracking, clustering, and structure statistics. Our main contributions include the formulation of a set of desirable mathematical and computational properties of comparative measures, and the classification of visualization tasks and applications that are enabled by these measures.

1. Introduction

Topological data analysis (TDA) provides fundamental tools for scientific visualization in terms of abstraction and summarization. These tools have great potential for data comparison, feature tracking, and ensemble analysis. For these purposes, a large variety of comparative measures have been proposed targeting different topological descriptors and employed in a variety of visualization applications, which are the focus of this survey. This state-of-the-art report aims to categorize, summarize, and analyze existing approaches that utilize comparative measures and identify open problems and opportunities for future work. We thus are interested in both the mathematical foundations and properties of comparative measures and their use in real-world visualization applications.

Popular topological descriptors for scalar field data, considered in this survey, can be classified into three categories: set-based such as persistence diagrams [ELZ02] and barcodes [Ghr08, CZCG04]; graph-based such as merge trees [BYM*14], contour trees [CSA03], and Reeb graphs [Ree46]; and complex-based such as Morse and Morse-Smale complexes [GP12, EHZ01, EHNP03], respectively. Our work is especially motivated by the following questions:

- Which role do topological methods in comparative analysis and visualization play and what are the typical applications?
- What comparative measures have been proposed and where are they applied?
- What are the desirable properties of a comparative measure for topological descriptors?

Our contributions include:

- We provide a classification of approaches in TDA and visualization relevant to the comparative study of scalar fields.
- We collect a set of desirable properties of a comparative measure concerning metricity, stability, discriminativity, and computational complexity;
- We analyze existing approaches with respect to these properties.
- We derive a list of opportunities and challenges for future work.

We provide three navigation aids that help the reader. Two tables provide an overview of different visualization tasks that are supported by comparative measures over topological descriptors (Table 1) and desirable properties for the various published comparative measures (Table 2). Further, we complement our survey with a visual literature browser (`https://git.io/Jt2Hq`) developed with the SurVis [BKW15] framework for an interactive navigation of the state of the art.

Existing surveys. An organized classification of the literature related to scalar field comparison is a valuable addition that complements existing state of the art on topology-based tool sets. The survey by Heine *et al.* [HLH*16] provides an overview of topology-based visualization, including a classification of such models for scalar fields, vector fields, tensor fields, and multi-fields. Specifically for scalar fields, the survey discusses topological descriptors, including their computation, simplification, visualization, and application. Our paper is different from the above survey because it focuses on comparative measures of topological descriptors for scalar fields, a topic not covered by previous surveys.

Review methodology. To complete the survey, we first gathered papers from a number of visualization, computational topology,

and TDA venues whose title and/or abstract contain keywords relevant to topological descriptors and their comparative measures, such as "persistent homology", "merge trees", "scalar field comparison", etc. The list of venues includes but is not limited to: journals such as IEEE Transactions on Visualization and Computer Graphics (TVCG), Computer Graphics Forum (CGF), Journal of Applied and Computational Topology, Journal of Computational Geometry (JoCG), as well as conferences/workshops such as IEEE Visualization Conference (VIS) and its associated events (*e.g.* IEEE Large Scale Data Analysis and Visualization or LDAV), EG/VGTC Conference on Visualization (EuroVis), IEEE Pacific Visualization Symposium (PacificVis), International Symposium on Computational Geometry (SoCG), etc. Since our primary focus is on topological descriptors and their applications in visualization, we did not survey papers from machine learning venues. We created a virtual index card summarizing each paper under topics such as "summary", "contributions", "topological descriptions proposed/used", "comparative measures proposed/used/parameters", "applications", "properties", "future directions mentioned in the paper", and additional "tags" and "notes". These index cards were then used during the categorization process and each card was checked by two authors. In total, this process resulted in approximately 200 papers that passed the initial screening, ∼100 of which were deemed most relevant and included in this survey.

Overview. This report is organized as follows: after introducing the basic classification categories and the desired properties in Sect. 2, some technical background on scalar field topology is summarized in Sect. 3. Comparative measures developed for topological descriptors with mathematical definition (if applicable) are summarized in Sect. 4. Sect. 5 serves as a reminder of the structure of the survey and provides navigation for the following sections, which focus on visualization application structured by visualization tasks for single fields (Sect. 6), time-dependent fields (Sect. 7), and ensembles (Sect. 8), respectively. A detailed discussion of desirable mathematical and computational properties and a systematic analysis of the surveyed comparative measures can be found in Sect. 9. The report ends with an outlook on future work and opportunities in Sect. 10 and a conclusion in Sect. 11.

2. Literature Research Procedure and Classification

We review representative papers in the field of computational topology, TDA, and visualization that develop or utilize topological descriptors for the comparative analysis and visualization of scalar fields. The annotation of each paper is guided primarily by a set of visualization tasks that are associated with three categories of data, and secondarily by a set of desirable mathematical and computational properties. Our primary categories are loosely inspired by an existing survey [HLH*16] that classifies papers based on the complexity of data types; and our secondary categorization is untreated in previous works.

2.1. Primary Categories Based on Visualization Tasks

During our literature review, we observed that comparative measures were developed with a focus either on a specific topological descriptor or a specific visualization task and application. We therefore identified three categories of data where topological comparison was applied: *single fields*, *time-varying fields*, and *ensembles*.

A single field f is a scalar-valued field defined on a 2D, 3D, or higher-dimensional domain $\mathbb{X}$, $f : \mathbb{X} \to \mathbb{R}$. A time-varying field F is a dynamically changing field, and is defined over the Cartesian product of a spatial domain $\mathbb{X}$ and a time axis $\mathbb{R}$, $F : \mathbb{X} \times \mathbb{R} \to \mathbb{R}$. Time-dependent data is typically available as a discrete set of temporal snapshots. An ensemble refers to a collection $\mathcal{F}$ of fields that are indexed by a collection of parameters, $\mathcal{F} = \{f_i : i \in I\}$ (where I is an index set).

Single fields. Comparative measures help extract, visualize, and highlight similar structures within a single field – broadly referred to as the symmetry detection problem in scalar fields. These measures also enable the comparison of two or more single fields for shape matching and retrieval (*e.g.*, [TN11, TN13, SSW14]).

Time-varying fields. For time-varying fields, comparative measures between successive time steps have been used to detect periodic behavior, key events, or outliers [NTN15, SW17, LWM*17, SMKN20]. Comparative measures also drive explicit feature tracking in time-varying data.

Ensembles. For ensembles, comparative measures help identify similar or dissimilar behavior between members. They help identify clusters of members, outliers, or unique members of the ensemble. More recently, they have been used to compute structure statistics that describe the distribution of the ensemble members in the parameter space [SPCT18, YWM*20, AMY*20].

2.2. Secondary Categories Based on Desirable Properties

We discuss desirable properties of a comparative measure $d = d(\mathcal{A}_1, \mathcal{A}_2)$ between a pair of topological descriptors (of the same type), $\mathcal{A}_1$ and $\mathcal{A}_2$. We focus on four types of properties surrounding *metricity*, *stability*, *discriminativity*, and *computational complexity*. These properties have been studied across scattered literature in TDA and visualization. We systematically investigate these properties and their relations to existing approaches in Sect. 9.

Metricity and Pseudometricity. Requiring d to be a metric is desirable. That is, d satisfies the following metric properties:

1. Non-negativity: $d(\mathcal{A}_1, \mathcal{A}_2) \geq 0$;
2. Identity: $d(\mathcal{A}_1, \mathcal{A}_2) = 0$ iff $\mathcal{A}_1 = \mathcal{A}_2$;
3. Symmetry: $d(\mathcal{A}_1, \mathcal{A}_2) = d(\mathcal{A}_2, \mathcal{A}_1)$;
4. Triangle inequality: $d(\mathcal{A}_1, \mathcal{A}_2) \leq d(\mathcal{A}_1, \mathcal{A}_3) + d(\mathcal{A}_2, \mathcal{A}_3)$.

If the triangle inequality (item 4) above is not required, d becomes a *dissimilarity measure* instead. If the identity is not required, d becomes a *pseudometric*, replacing item 2 above by:

- $d(\mathcal{A}_1, \mathcal{A}_1) = 0$ (but possibly $d(\mathcal{A}_1, \mathcal{A}_2) = 0$ for some distinct $\mathcal{A}_1 \neq \mathcal{A}_2$).

Stability. Many definitions of stability for a distance metric d with respect to the underlying scalar field have been proposed. Stability can refer to whether d is stable with respect to simplification or perturbation of the underlying function. For example, given two

scalar fields f_1 and $f_2 : \mathbb{X} \to \mathbb{R}$ that give rise to a pair of topological descriptors $\mathcal{A}_1$ and $\mathcal{A}_2$, d is L^∞-stable if for some constant $C > 0$,

$$d(\mathcal{A}_1, \mathcal{A}_2) \leq C \cdot ||f_1 - f_2||_\infty.$$

Discriminativity. Discriminativity also has various definitions. For instance, using a comparative measure d_0 as a baseline, d is considered to be more discriminative than d_0 if for some constant $c > 0$,

$$d_0(\mathcal{A}_1, \mathcal{A}_2) \leq c \cdot d(\mathcal{A}_1, \mathcal{A}_2)$$

and there exists no constant $c' > 0$ such that $d_0 = c' \cdot d$ (that is, d is not a scaled version of d_0).

Computational complexity. We investigate the computational complexity of d in terms of the time and space complexity, scalability, and parallel computing. We investigate whether d is *easily implementable*, referring to whether an algorithmic solution has been proposed which affects its practicality.

The above properties are particularly desirable for analysis and visualization tasks that are supported by a comparative measure. They lead to theoretically sound, interpretable, robust, reliable, and practical methods for comparative visualization.

3. Technical Foundations on Scalar Field Topology

In this section, we review the technical foundations for scalar field topology, including the definitions of Morse functions and topological descriptors; see [Zom05, EH10] for computation-oriented and [Tie17] for visualization-oriented introduction to scalar field topology. We review set-based (persistence diagrams, barcodes), graph-based (merge trees, contour trees, Reeb graphs), and complex-based (Morse and Morse-Smale complexes) topological descriptors and their variants. The graph-based descriptors are largely based on contours of a function, whereas complex-based ones are primarily based on its gradient. We also briefly mention relevant topological descriptors for multivariate functions (Jacobi sets, Reeb spaces, joint contour nets), as they are natural extensions of their univariate counterparts.

3.1. Morse Functions and Morse Theory

Most of the topological descriptors described in this section are rooted in Morse theory [Mil63]. We give a high-level review here; see [Mat02] for a friendly introduction and [Mil63] for the original treatment.

Morse functions. Let $\mathbb{M}$ be a smooth manifold and $f : \mathbb{M} \to \mathbb{R}$ a smooth function on $\mathbb{M}$. A point $x \in \mathbb{M}$ is a *critical point* of f if and only if the partial derivatives at x are zero; otherwise, it is a *regular point*. The image of a critical point is a *critical value* of f. A critical point x is *non-degenerate* if the Hessian (the matrix of second derivatives) at x is non-singular. f is a *Morse function* if all its critical points are non-degenerate and have distinct function values. Fig. 1 gives two examples of Morse functions with a 1- and a 2-dimensional domain, respectively. Critical points are always displayed as red (for local maxima), blue (for local minima), and white (for saddles) circles or spheres.

Morse theory. For a Morse function $f : \mathbb{M} \to \mathbb{R}$, let $\mathbb{M}_t := f^{-1}(-\infty, t] = \{x \in \mathbb{M} \mid f(x) \leq t\}$ denote sublevel sets of f. A basic result of Morse theory states that almost all functions are Morse functions. Technically speaking, the set of Morse functions forms an open dense subset of the space of smooth functions. In practice, a non-Morse function can be made into a Morse function by resolving degenerate conditions via the simulation of simplicity [EM90]. We assume all functions discussed in this paper to be Morse.

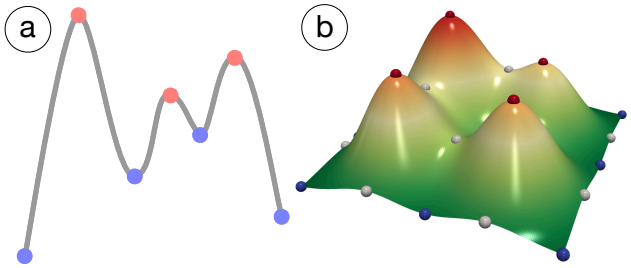

Figure 1: *Morse functions with (a) a 1-dimensional and (b) a 2-dimensional domain, respectively.*

The Morse lemma states that a function looks extremely simple near a non-degenerate critical point. Two fundamental theorems of Morse theory study how sublevel sets of a function changes topologically *w.r.t.* its critical points. A number of theoretical properties relevant to topological descriptors described in this section can be traced back to these two fundamental theorems. We refer interested readers to [Mil63, Theorems 3.1 and 3.2] in their original forms. In a nutshell, these theorems describe if and when the topology of sublevel sets $\mathbb{M}_t$ change as t varies, in particular, when t passes a critical value. Topological descriptors such as persistent diagrams and merge trees are related with one another via theorems of Morse theory as both are defined over the sublevel sets of a function.

In practice, we rarely find smooth functions. Instead, we are given samples of such functions, represented as a function on a point cloud sample of $\mathbb{M}$. Oftentimes, we impose a combinatorial structure (*i.e.*, a simplicial complex K) on the sample as an approximation of $\mathbb{M}$. Let K be a simplicial complex with real values specified on its vertices; $|K|$ represents its *underlying space*. We obtain a piecewise linear (PL) function $f : |K| \to \mathbb{R}$ using linear extension over the simplices, where $f(x) = \sum_i b_i(x) f(u_i)$ (u_i are vertices of K and $b_i(x)$ are the barycentric coordinates of x) [EH10, page 135]. We can then apply Morse-theoretical ideas to this PL approximation. This application is justifiable according to the *Simplicial Approximation Theorem* [EH10, page 56], which states that every continuous function on a triangulable topological space can be approximated by a PL function.

As described in subsequent sections, in some instances, features that form parts of topological descriptors are used in the comparative measures, in particular, critical points and their attributes, level sets (contours, or isosurfaces) defined as $f^{-1}(t)$ for some $t \in \mathbb{R}$.

3.2. Persistence Diagrams and Barcodes

Persistent homology is a widely used tool for TDA and visualization. Algebraically, it takes the form of a persistence mod-

ule [CDSGO16]. In this paper, we are mostly concerned with persistence homology that arises from *sublevel set filtrations* of Morse functions. We refer the reader to [EH10, CdS10, BEMP13] for different ways to study persistent homology.

Persistence diagrams. Let $f : \mathbb{M} \to \mathbb{R}$ be Morse and $\mathbb{M}_t := f^{-1}(\infty, t]$ its sublevel sets. Assuming $\mathbb{M}$ is also compact, then a Morse function f on a compact manifold contains finitely many critical points (as a consequence of the Morse lemma). Let n be the (finite) number of critical values of f. Let $a_0 < \cdots < a_n$ be a sequence of regular values of f such that each interval (a_i, a_{i+1}) contains exactly one critical value of f. A sublevel set filtration of f is a sequence of sublevel sets connected by inclusions,

$$\mathbb{M}_{a_0} \to \mathbb{M}_{a_1} \to \cdots \to \mathbb{M}_{a_n}.$$

Persistent homology studies the topological changes of sublevel sets by applying k-dimensional homology ($k \geq 0$) to this sequence,

$$\mathsf{H}_k(\mathbb{M}_{a_0}) \to \mathsf{H}_k(\mathbb{M}_{a_1}) \to \cdots \to \mathsf{H}_k(\mathbb{M}_{a_n}).$$

Given a topological space $\mathbb{X}$, the 0-, 1-, and 2-dimensional homology groups, denoted as $\mathsf{H}_0(\mathbb{X})$, $\mathsf{H}_1(\mathbb{X})$, and $\mathsf{H}_2(\mathbb{X})$, respectively, capture the connected components, tunnels, and voids of $\mathbb{X}$. We give an example of 0-dimensional persistence homology based on the sublevel set filtration of a 1-dimensional Morse function in Fig. 2.

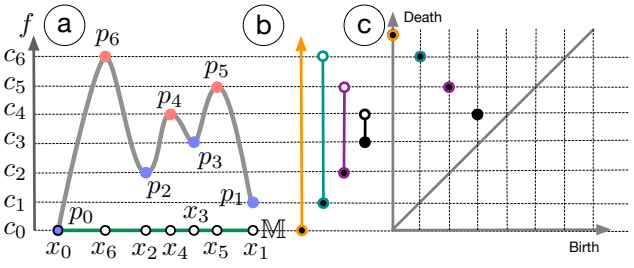

Figure 2: *(a) The graph of $f : \mathbb{M} \to \mathbb{R}$, where each point $p_i = (x_i, c_i)$ for $c_i = f(x_i)$; together with (b) the 0-dimensional barcode and (c) 0-dimensional persistence diagram of f based on its sublevel set filtration.*

Formally, a k-dimensional persistence diagram $\mathcal{D}$ is the disjoint union of a multi-set of off-diagonal points $\{(b,d) \mid b \neq d, b, d \in \mathbb{R}_{\geq 0}\}$ on the Euclidean plane $\bar{\mathbb{R}}^2$ (where $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$) and the diagonal $\Delta = \{(b,b) \mid b \in \mathbb{R}_{\geq 0}\}$ counted with infinite multiplicity. As illustrated in Fig. 2a, let c_i denote the critical values of a Morse function f restricted to an interval $\mathbb{M} \subset \mathbb{R}$, $f : \mathbb{M} \to \mathbb{R}$, where $c_0 < c_1 < \cdots < c_6$. Let x_i denote the critical points of f. Assume f is Morse, then $c_i = f(x_i)$. For simplicity, we set $c_0 = 0$, $c_1 = 1$, and $c_i = i$, etc. Let $a_0 < a_1 < \cdots < a_7$ be a sequence of regular values of f such that each interval (a_i, a_{i+1}) contains exactly one critical value c_i. The 0-dimensional persistent homology captures how connected components in the sublevel sets $\mathbb{M}_t$ changes as t varies from a_0 to a_7. At $t = a_0 < 0$, $\mathbb{M}_t = \emptyset$. At $t = 0$, a single (connected) component appears in the sublevel set $\mathbb{M}_t$ containing the global minimum x_0, we call this a *birth* event at $\mathbb{M}_0$. At $t = 1, 2$, and 3, a 2nd, 3rd, and 4th component appears in $\mathbb{M}_t$ containing local minima x_1, x_2, and x_3, respectively. At $t = 4$, the component containing x_3 merges with the component containing x_2 as per the *Elder Rule* [EH10, Page. 150], referred to as a *death* event: the component containing x_3 disappears (dies) while the component containing x_2 remains. At $t = 5$, the component containing x_2 merges with the component containing x_1 and dies. At $t = 6$, the component containing x_1 merges with the component containing x_0 and dies. Persistent homology pairs the birth and death events either as a set of intervals (called *barcode*), or a multi-set of points in the plane (called *persistence diagram*).

Barcodes. A barcode is shown in Fig. 2b. The component containing x_0 never dies, giving rise to a bar $[0, \infty)$ in the barcode that begins at 0 and goes to ∞. The component containing x_1 is born at $t = 1$ and dies at $t = 6$, which corresponds to a bar $[1, 6)$. Similarly, the birth and death events of components containing x_2 and x_3 give rise to two additional bars $[2, 5)$ and $[3, 4)$, respectively. The *persistence* of a bar $[b, d)$ in a barcode is defined to be $|d - b|$, which captures the life span of a component in the filtration. A persistence diagram is shown in Fig. 2c, where each bar $[b, d)$ is mapped to a point (b, d) on the plane.

Other variants exist, mostly derived from persistence diagrams or barcodes. The *persistence landscape* [Bub15] is a function-based representation of a persistence diagram. It maps a persistence diagram into a function space, which allows it to be easily integrated with tools from statistics and machine learning [BD17, Bub20]. Formally, for a birth-death pair (b, d) in a persistence diagram, assuming b and d are finite, we define a piecewise-linear function $f_{(b,d)} : \mathbb{R} \to [0, \infty]$ as

$$f_{(b,d)} = \begin{cases} 0, & \text{if } x \notin (b,d) \\ x - b, & \text{if } x \in (b, \frac{b+d}{2}] \\ -x + d, & \text{if } x \in [\frac{b+d}{2}, d) \end{cases}.$$

The *persistence landscape* of the birth-death pairs $\{(b_i, d_i)\}_{i=1}^n$ in a persistence diagram is the sequence of functions $\lambda_k : \mathbb{R} \to [0, \infty]$, where $\lambda_k(x)$ is the k-th largest value of $\{f_{(b_i,d_i)}(x)\}_{i=1}^n$ (for $k = 0, 1, 2, \ldots$). $\lambda_k(x) = 0$ if the k-th largest value does not exist. In other words, the *persistence landscape* is a function $\lambda : \mathbb{N} \times \mathbb{R} \to [0, \infty]$, where $\lambda(k, t) = \lambda_k(t)$ [BD17]. Intuitively, consider the points with finite birth and death times in a persistence diagram (Fig. 3a). We construct a persistence landscape in Fig. 3b by rotating the points by 45° and building three linear functions, λ_1 (blue), λ_2 (red), and λ_3 (green), with these points.

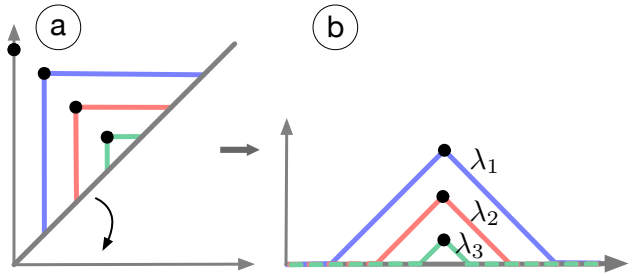

Figure 3: *Rotating a persistence diagram in (a) to create a functional representation – a persistence landscape in (b).*

Another descriptor widely used in machine learning is the *persistence image* [AEK*17]. It is a vector-based representation of a

persistence diagram. It can be informally considered as a heat map, which is generated from a weighted sum of Gaussian centered at each point (b,p), where b is the birth and $p=d-b$ is the persistence of a point in the persistence diagram.

Betti curves also summarize the information of persistent homology (*e.g.*, [Rob02, GMK04, RSL20b, CL20]). Recall the k-th Betti number is informally the number of k-dimensional holes (homology) of a topological space. For a filtration parameter t, the Betti curves at t are the Betti numbers of the associated complex. Betti curves are arguably the simplest function-based representation of a persistence diagram (*cf.*, the persistence landscape). Turner *et al.* [TMB14] introduced a summary statistic from persistence diagram, called the persistent homology transform (PHT), to model surfaces in $\mathbb{R}^3$ and shapes in $\mathbb{R}^2$. Li *et al.* [LWA*17] proposed another persistence-based feature vectorization of a persistence diagram using a 1-dimensional density function to compare neuronal trees; their feature vectorizations can be considered as a 1-dimensional version of the persistent images [AEK*17]. Rieck *et al.* [RSL17] developed an inter-level set persistence hierarchy (ISPH) to capture the spatial relationship between features in persistence diagram.

3.3. Merge Trees, Contour Trees, and Reeb Graphs

Topological descriptors such as merge trees, contour trees, and Reeb graphs capture topological changes of (sub)level sets of scalar fields, which are real-valued smooth functions.

Merge trees. Given a Morse function $f:\mathbb{M}\to\mathbb{R}$ defined on a connected domain $\mathbb{M}$, a merge tree records the connectivity of its sublevel sets. Two points $x,y\in\mathbb{M}$ are *equivalent* (*w.r.t.* f), $x\sim y$, if they have the same function value, that is, $f(x)=f(y)=t$, and if they belong to the same connected component of the sublevel set $\mathbb{M}_t$, for some $t\in\mathbb{R}$. A *merge tree* is the quotient space $\mathbb{M}/\sim$ obtained by gluing together points in $\mathbb{M}$ that are equivalent under the relation $\sim$. It keeps track of the evolution of connected components in $\mathbb{M}_t$ as t increases; see Fig. 4 for an example. In the abstract view of a merge tree in Fig. 4b, each leaf corresponds to a local minimum of f that represents the birth of a connected component; each internal node corresponds to the merging of components; and the root represents the entire space as a single component. Fig. 4b also visualizes the branches of the merge tree based on its branch decomposition. The connection between a merge tree and the barcode is apparent, *cf.* Fig. 2(b) and Fig. 4(b-c), where a merge tree decomposes into a barcode following a branch decomposition process; and bars in a barcode can be used to assemble a (non-unique) merge tree following a gluing process. See [CCF*20, Cur18, KGH20] for references for the relation between a merge tree and a barcode. Note that the notions of join and split trees [CSA03] are the two forms of merge trees; a join tree is the merge tree of f and a split tree is the merge tree of $-f$.

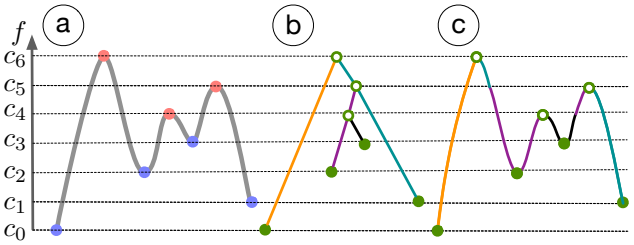

Figure 4: *(a) The graph of a 1-dimensional Morse function f restricted to an interval, $f:\mathbb{M}\to\mathbb{R}$; (b) the merge tree of f shown abstractly, where branches are colored based on its branch decomposition; (c) the graph of f is colored based on the branch decomposition in (b).*

Reeb graphs and contour trees. A Reeb graph, on the other hand, relies on equivalence relations among points in the *level sets* of a Morse function $f:\mathbb{M}\to\mathbb{R}$. Two points $x,y\in\mathbb{M}$ are *equivalent*, $x\sim y$, if $f(x)=f(y)=t$, and if they belong to the same connected component of the level set $f^{-1}(t)$, for some $t\in\mathbb{R}$. The *Reeb graph* $\mathcal{G}_f:=\mathbb{M}/\sim$ is the quotient space obtained by identifying equivalent points; see Fig. 5. Nodes in the Reeb graph have a one-to-one correspondence with the critical points of f, while arcs connect the nodes. A point on an arc represents a connected component of a level set (i.e., a *contour*) in $\mathbb{M}$. Intuitively, as t increases within the range of f, a Reeb graph captures the topological changes in the level sets of f, in particular, the appearances, disappearances, splitting, and merging among the connected components (contours) of $f^{-1}(t)$; see [EH10, section VI.4] for a formal treatment. Bauer *et al.* [BFL16] worked with the notion of a *labeled Reeb graph*, where the vertices of $\mathcal{G}_f$ are labeled by the function $l_f:V(\mathcal{G}_f)\to\mathbb{R}$ induced by restricting $f:\mathbb{M}\to\mathbb{R}$ to its critical points. Then, $(\mathcal{G}_f,l_f)$ is the labeled Reeb graph of the data $(\mathbb{M},f)$, see Fig. 5c.

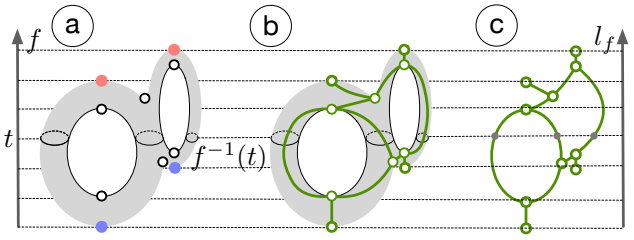

Figure 5: *(a) A height function $f:\mathbb{M}\to\mathbb{R}$ defined on a double torus, (b) its Reeb graph embedded in the domain $\mathbb{M}$, and (c) its Reeb graph shown in an abstract view. If the Reeb graph in (c) is further equipped with a function l_f defined on its vertices, where l_f is the restriction of f to V, then we obtain a labeled Reeb graph.*

A contour tree is a special type of Reeb graph when the domain $\mathbb{M}$ is simply connected. Then, $\mathbb{M}/\sim$ gives rise to a tree; see Fig. 6 for an example involving a "deformed" spherical domain. The main difference between a contour tree and a merge tree is that the former captures the connectivity among level sets, while the latter encodes the connectivity among sublevel sets of a Morse function.

Mapper constructions and mapper graphs. Given a point cloud $\mathbb{X}\subset\mathbb{R}^d$, we construct the *nerve of a covering*. Let I be an index set. A *cover* of $\mathbb{X}$ is defined as a set of open sets in $\mathbb{R}^d$, $\mathcal{U}=\{U_i\}_{i\in I}$ such that $\mathbb{X}\subset\cup_{i\in I}U_i$. The *nerve* complex of $\mathcal{U}$ is a simplicial complex, $\mathcal{N}(U):=\{J\subset I\mid\cap_{j\in J}U_j\neq\emptyset\}$. The 1-dimensional nerve of $\mathcal{U}$, denoted as $\mathcal{N}_1(\mathcal{U})$, is a graph. Each node $i\in I$ in $\mathcal{N}_1(\mathcal{U})$ rep-

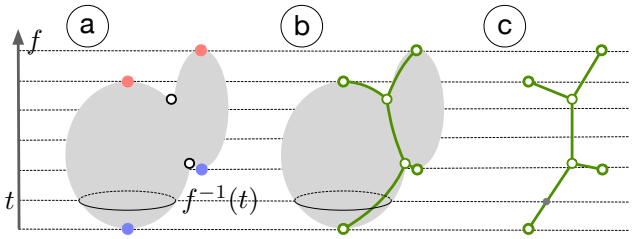

Figure 6: *(a) A height function $f : \mathbb{M} \to \mathbb{R}$ defined on the surface of two (solid) balls glued together; (b) its contour tree embedded in the domain $\mathbb{M}$; and (c) its contour tree shown in an abstract view.*

resents a cover element U_i, and there is an edge between $i, j \in I$ if $U_i \cap U_j \neq \emptyset$.

Given a real-valued function $f : \mathbb{X} \to \mathbb{R}$, we start with a finite cover of $f(\mathbb{X}) \subset \mathbb{R}$ using intervals, that is, a cover $\mathcal{V} = \{V_k\}$ such that $f(\mathbb{X}) \subseteq \cup_k V_k$. We obtain a cover $\mathcal{U}$ of $\mathbb{X}$ by considering the clusters induced by points in $f^{-1}(V_k)$ for each V_k as cover elements. The nerve of $\mathcal{U}$ is a simplicial complex, and is referred to as the *mapper* (or mapper construction) of f. The 1-dimensional nerve of $\mathcal{U}$, $\mathcal{N}_1(\mathcal{U})$, is the *mapper graph* of $(\mathbb{X}, f)$.

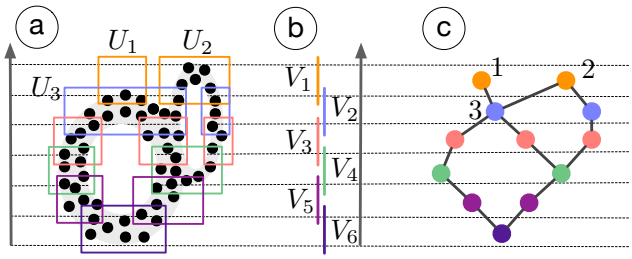

Figure 7: *(a) A height function $f : \mathbb{X} \to \mathbb{R}$ defined on a point cloud sampled from a double annulus together with a cover, (b) the cover of $f(\mathbb{X})$ with intervals, and (c) its mapper graph.*

Take as an example a 2-dimensional point cloud $\mathbb{X}$ sampled from a double annulus that is equipped with a height function in Fig. 7a. Six intervals form a cover $\mathcal{V} = \{V_1, V_2, \cdots, V_6\}$ of the image of f, that is, $f(\mathbb{X}) \subset \bigcup_k V_k$ $(1 \leq k \leq 6)$ in Fig. 7b. For each k, $f^{-1}(V_k)$ induces some clusters of points that are subsets of $\mathbb{X}$; each cluster forms cover elements of $\mathbb{X}$. For instance, $f^{-1}(V_1)$ induces two clusters of points that are enclosed by the orange cover elements U_1 and U_2, and $f^{-1}(V_2)$ induces two clusters enclosed by the blue cover elements, one of which is U_3. The mapper graph shows that there is an edge between node 1 and node 2 in Fig. 7c since $U_1 \cap U_3 \neq \emptyset$.

Other contour-based topological descriptors have been studied in recent years. Zhang *et al.* introduced the *dual contour tree*, which is constructed from the contour tree of a volume by dividing its functional range into segments such that the connected contour tree edges within a segment become a node in the dual tree [ZBB04]. The dual contour tree shares many resemblances with the mapper graph; see Sect. 6 and Sect. 8 for its applications in visualization.

A *branch decomposition tree* (BDT) is derived from a contour tree [PCMS04] or a merge tree [SSW14]. A BDT represents the branch decomposition of a tree, with the nodes representing the branches and the edges representing their hierarchy. Saikia *et al.* [SSW14] further introduced an *extended branch decomposition graph* (eBDG), which represents a forest of BDTs, where each of the BDTs is computed from a subtree of the merge tree.

In addition to mapper graphs, Reeb graphs have several variants, many of which have not been utilized in scientific visualization. The α-*Reeb graph* [CHS15] defines the equivalence relation between points using open intervals of length at most α. The *extended Reeb graph* [BB14] uses cover elements from a partition of the domain without overlaps. The *enhanced mapper graph* [BBMW21] considers inverse images of intersections among the cover elements and encodes function values on its vertices and edges. Several variants of mapper constructions exist, as discussed in Sect. 3.5.

3.4. Morse and Morse-Smale Complexes

Let $f : \mathbb{M} \to \mathbb{R}$ be a Morse function, ∇f its gradient. At a regular point x, an *integral line* is a maximal path whose tangent vectors agree with the gradient [EHZ01]. An integral line begins and ends at critical points. The *stable manifold* surrounding a critical point p includes p itself and all regular points whose integral lines end at p. This is also referred to as the *descending manifold* of p since $f(p) \geq f(x)$ for all points x in the stable manifold of p [EH10, Page 131]. For instance, the stable manifold of the local maximum p in Fig. 8a corresponds to the red "bump". The *unstable manifold* (*ascending manifold*) of a critical point p is the point itself together with all regular points whose integral lines originate at p [EH10, Chap. VI, page 131], see Fig. 8b. Symmetrically, an unstable manifold (*ascending manifold*) of p in f is a stable manifold of p in $-f$. A Morse function f is a *Morse-Smale function* if the stable and unstable manifolds intersect transversally.

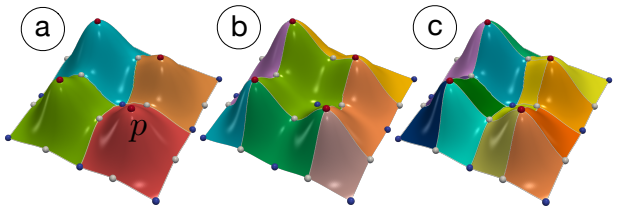

Figure 8: *Given the 2-dimensional function f from Fig. 1, (a) shows the Morse complex of f (with stable manifolds), (b) shows the Morse complex of $-f$ (with unstable manifolds), and (c) is the Morse-Smale complex of f.*

Given a Morse-Smale function f defined on a 2-dimensional domain, its stable manifolds surrounding local maxima decompose the domain into 2-cells (colored regions in Fig. 8a), whereas integral lines connecting the critical points are the 1-cells, and critical points are the 0-cells. These cells form a complex called a *Morse complex* of f. Intersecting the stable and unstable manifolds of f (equivalently, intersecting the Morse complex of f and $-f$) gives rise to a refinement of the two complexes called the *Morse-Smale complex* (MSC) of f, see Fig. 8c. Its 0-cells are the critical points,

and its 1- and 2-cells are the components of the unions of integral lines with a common origin and a common destination [EH10, Chap. VI, page 134]. 3D Morse and Morse-Smale complexes of $f : \mathbb{M} \subset \mathbb{R}^3 \to \mathbb{R}$ are defined similarly based on the gradient behavior of points in its domain [EHNP03]. These complexes can be approximated in high dimensions for data analysis and regression [GBPW10, GP12]. Persistent homology can be used to simplify a MSC [EHZ01]; see [GRSW14] for a discussion.

Subsets of Morse-Smale Complexes. An *extremum graph*, introduced by Correa *et al.* [CLB11], is a sparse subset of the MSC. It connects critical points along steepest ascending (or descending) lines, which join adjacent extrema [CLB11]. It is designed to retain (some) important structural information of a MSC without visual clutter from the entire complex. A maximum graph contains maximum-saddle connections, whereas a minimum graph contains minimum-saddle connections. Thomas and Natarajan [TN13] augmented the extremum graph with topological and geometric information to facilitate the efficient detection of geometric symmetry in the electron microscopy data.

Feng *et al.* [FHJB13] introduced *feature graphs* to represent non-rigidly deformed surfaces. A feature graph is derived from the MSC of the Auto Diffusion Function (ADF), a solution to the heat equation. Nodes in a feature graph are critical points of a persistence-simplified MSC, which are connected by integral lines. Thus, a feature graph is the 1-dimensional skeleton of a simplified MSC.

3.5. Topological Descriptors of Multivariate Functions

We briefly describe topological descriptors of multivariate functions, although they are not the focus of this paper. We specifically focus on these multivariate descriptors as many of them are the direct extensions of their univariate counterparts. Given a multivariate function $F = (f_1, f_2, \cdots, f_k) : \mathbb{M} \to \mathbb{R}^k$ ($k \geq 2$), we have three types of descriptors, those based on (a) the gradient behaviors of components f_i (Jacobi sets), (b) the contours of F (Reeb spaces, multivariate mapper constructions), and (c) multi-parameter persistent homology.

Reeb spaces, multivariate mapper constructions, joint contour nets. Reeb spaces [EHP08] are high-dimensional analogs of Reeb graphs. Given a multivariate function $F : \mathbb{M} \to \mathbb{R}^k$, the *Reeb space* is the quotient space obtained by identifying equivalent points, that is, $\mathbb{M}/\sim$, where $x \sim y$ if $F(x) = F(y) = t \in \mathbb{R}^k$ and x and y belong to the same connected component of the pre-image of t.

Following the mapper construction for a scalar field (Fig. 7), the filter function f may be generalized to be a multivariate function, that is, $F : \mathbb{M} \to \mathbb{R}^k$ ($k \geq 2$). For instance, when $k = 2$, the corresponding cover elements of $F(\mathbb{M}) \subset \mathbb{R}^2$ become rectangles. We call this the multivariate mapper construction in this paper to differentiate it from its univariate (scalar field) version.

Since a mapper graph is considered as a discrete approximations of a Reeb graph, the mapper construction for multivariate data $F : \mathbb{X} \to \mathbb{R}^k$ is a discrete approximation of the Reeb space [MW16]. There are other variants of such approximations, noticeably the *joint contour nets* (JCNs) [CD13]. The JCN applies quantizations to the cover elements by rounding the function values. The *multi-scale mapper* [DMW16] is a sequence of mapper constructions connected by linear maps by varying the granularity of the cover elements. The *multi-nerve mapper* [CO18] computes the multi-nerve [ECdVGG12] of a cover. For comparing time-varying and multi-fields (see Sect. 7), Agarwal *et al.* [ARC20] introduced a multi-resolution Reeb Space (MRS), which is approximated as a series of JCNs at various levels of discretization.

Jacobi sets. The relation between two Morse functions $f, g : \mathbb{M} \to \mathbb{R}$ can be studied in terms of their Jacobi set [EH04], $J(f,g)$. The Jacobi set is the collection of points in $\mathbb{M}$ where the gradients of the functions align, that is, for some $\lambda \in \mathbb{R}$,

$$J(f,g) = \{x \in \mathbb{M} \mid \nabla f(x) + \lambda \nabla g(x) = 0 \text{ or } \lambda \nabla f(x) + \nabla g(x) = 0\}.$$

The Jacobi set has been used to derive local and global comparison measures of multiple scalar functions [EHNP04]. Several techniques have been developed for its topological simplification [NN11, BWN*15]. A relevant concept is *Pareto sets* [HG15].

Multi-parameter persistence is an active area of research, where previous results surrounding the indecomposables of multi-parameter persistence modules have been largely theoretical (see [CZ09, Les12] for relevant readings). Multi-parameter versions of barcodes and their variants are actively researched, see recent results on multi-parameter persistence landscapes [Vip20] and persistence images [CB20] respectively. Noticeably, the software RIVET [The20] computes barcodes from “slices" from 2-dimensional persistence modules.

4. Comparative Measures for Topological Descriptors

Comparing scalar fields using their topological descriptors is an important tool in the study of scientific data. Defining and computing these comparative measures give rise to interesting problems both in theory and in practice. In this section, we review various definitions of comparative measures for topological descriptors before discussing their applications in visualization in Sect. 6, Sect. 7, and Sect. 8. We defer the discussion on their mathematical and computational properties to Sect. 9. We give formal definitions in the forms of equations for some of the well-known comparative measures. We give informal descriptions for their variants. We defer detailed discussions to later sections for comparative measures designed specifically for visualization tasks, which oftentimes are coupled with heuristics and/or data-dependent modifications.

Before diving into the technical descriptions of these comparative measures, we would like to discuss the different origins and motivations behind these developments. For instance, comparative measures for persistence diagrams, such as the bottleneck and Wasserstein distances, are related to optimal transport [Vil03]. Functional distortion distances for Reeb graphs are the continuous version and a constant factor approximation of the extended Gromov-Hausdorff distances, a classic tool from the study of metric spaces; while interleaving distances originate from the algebraic study of persistence modules. Kernels for persistence diagrams interface with kernel methods for machine learning. The persistence scale-space kernel takes inspirations from the scale-space theory in signal processing, while persistence Fisher kernel is derived from

information theory. Each comparative measure enjoys a set of desirable properties (Sect. 9) and is suited for a specific collection of analysis and visualization tasks (Sect. 6, Sect. 7, and Sect. 8), which motivated its development in the first place.

In the following sections, $\mathcal{D}$ represents a persistence diagram and its variants (persistence landscape and persistence image), $\mathcal{T}$ represents a tree-based descriptor, $\mathcal{G}$ represents a graph-based descriptor, and $\mathcal{M}$ represent complex-based descriptors, including Morse and Morse-Smale complexes. We emphasize the function as labels when a comparative measure explicitly encodes information from the function (*e.g.,* $\mathcal{T}_f$, $\mathcal{T}_g$), and we use numeric labels (*e.g.,* $\mathcal{D}_1, \mathcal{D}_2$) otherwise.

4.1. Comparing Persistence Diagrams and Their Variants

We review classic distances between persistence diagrams, namely, bottleneck and p-Wasserstein distances, as well as distances between their variants, such as p-landscape distances. We also include kernels defined on persistence diagrams that interface with machine learning.

Bottleneck and Wasserstein distances. To compare persistence diagrams, the bottleneck distance [CSEH07,EH08] and the Wasserstein distance [CSEHM10] are well established and widely used, for instance, in similarity estimation [HZLR20] and machine learning tasks [Bub15, ZW19].

Definition 1 [EH08, Bottleneck distance] Given two persistence diagrams $\mathcal{D}_1$, $\mathcal{D}_2$ and a bijection $\eta : \mathcal{D}_1 \to \mathcal{D}_2$, the *bottleneck distance* between $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined as

$$d_\infty(\mathcal{D}_1, \mathcal{D}_2) = \inf_{\eta:\mathcal{D}_1\to\mathcal{D}_2} \sup_{x\in\mathcal{D}_1} ||x - \eta(x)||_\infty. \tag{1}$$

Definition 2 [CSEHM10, p-Wasserstein distance] The *p-Wasserstein distance* is defined as

$$d_p(\mathcal{D}_1, \mathcal{D}_2) = \left[\inf_{\eta:\mathcal{D}_1\to\mathcal{D}_2} \sum_{x\in\mathcal{D}_1} ||x - \eta(x)||_\infty^p \right]^{\frac{1}{p}} \tag{2}$$

While Eq. 2 is a typical notion in the literature, Turner *et al.* [TMMH14] discuss a more general formulation by introducing a second parameter (i.e., q) to Eq. 2 that specifies the degree of the point-wise norm; that is, by replacing L^∞ norm in Eq. 2 with a L^q norm; where $q = 2$ in [TMMH14].

Kernels for persistence diagrams. Since persistence diagrams do not have the structure of an inner product space (*i.e.* Hilbert space), various kernels have been introduced to interface persistence diagrams with kernel-based machine learning models such as kernel support vector machines (SVMs). An intuitive way to think about kernels for SVMs is that kernels are similarity functions for a pair of objects. A number of kernels exist for persistence diagrams, such as the persistence scale-space kernel [RHBK15], the persistence weighted Gaussian kernel [KFH17], the sliced Wasserstein kernel [CCO17], and the persistence Fisher kernel [LY18], denoted as K_S, K_G, K_W and K_F, respectively.

Let $\mathcal{D}_1$ and $\mathcal{D}_2$ denote two k-dimensional persistence diagrams. The *persistence scale-space kernel* [RHBK15] K_S is defined as

$$K_S(\mathcal{D}_1, \mathcal{D}_2, \sigma) = \frac{1}{8\pi\sigma} \sum_{p\in\mathcal{D}_1, q\in\mathcal{D}_2} e^{\frac{\|p-q\|}{8\sigma}} - e^{\frac{\|p-\overline{q}\|}{8\sigma}}, \tag{3}$$

where $\forall\ q = (b,d) \in \mathcal{D}_2$, we define $\overline{q} = (d,b)$, that is, $\overline{q}$ is a reflection of q along the diagonal Δ; σ is bandwidth of the Gaussian kernel.

The *persistence weighted Gaussian kernel* (PWGK) [KFH17] K_G is defined as

$$K_G(\mathcal{D}_1, \mathcal{D}_2, \sigma) = \sum_{p\in\mathcal{D}_1, q\in\mathcal{D}_2} w(p)w(q)e^{-\frac{\|p-q\|^2}{2\sigma^2}}, \tag{4}$$

where $w(p)$ is the weight assigned to the point p. Kusano *et al.* [KFH17] suggest $w(p) = \arctan(C(d-b)^t)$ as the weight for $p = (b,d)$, where C is a positive constant for practical purposes, and t is assumed to be greater than the dimension of the underlying space.

Given a unit vector θ in $\mathbb{R}^2$, let $L(\theta) = \{\lambda\theta \mid \lambda \in \mathbb{R}\}$ denote the line and $\pi(\theta, p)$ denote the orthogonal projection of point p on the line $L(\theta)$. To compute the *sliced Wasserstein kernel* [CCO17], we first augment persistence diagram $\mathcal{D}_1$ with the orthogonal projection of points in $\mathcal{D}_2$ onto the diagonal (denoted as $\mathcal{D}_1^\Delta$) and vice versa (denoted as $\mathcal{D}_2^\Delta$) to obtain two new sets $\mathcal{D}_1^*$ and $\mathcal{D}_2^*$. That is, $\mathcal{D}_1^* = \mathcal{D}_1 \cup \mathcal{D}_2^\Delta$ and $\mathcal{D}_2^* = \mathcal{D}_2 \cup \mathcal{D}_1^\Delta$. The *sliced Wasserstein distance* between these two sets is approximated as

$$SW(\mathcal{D}_1^*, \mathcal{D}_2^*, M) = \frac{1}{\pi} \sum_{j=1}^{M} \|V(\mathcal{D}_1^*, \theta_j) - V(\mathcal{D}_2^*, \theta_j)\|_1, \tag{5}$$

where M is the number of directions, $\theta_j = j\pi/M - \pi/2$ and $V(\mathcal{D}_1^*, \theta_j)$ is the vector of dot products $< p, \theta_j >$ of all points $p \in \mathcal{D}_1^*$. The *sliced Wasserstein kernel* is then computed as

$$K_W(\mathcal{D}_1^*, \mathcal{D}_2^*, M) = e^{\frac{-SW(\mathcal{D}_1^*, \mathcal{D}_2^*, M)}{2\sigma^2}}, \tag{6}$$

Given an k-dimensional persistence diagram $\mathcal{D}$ and a bandwidth $\sigma > 0$, we can define a smooth, normalized measure

$$\rho_\mathcal{D} = \left[\frac{1}{Z} \sum_{u\in\mathcal{D}} N(x; u, \sigma I) \right]_{x\in\Theta} \tag{7}$$

over a given set Θ, where I is the identity matrix, N is a Gaussian function, and $Z = \int_\Theta \sum_{u\in A} N(x; u, \sigma I) \mathrm{d}x$. Note that if Θ is the entire Euclidean space $\mathbb{R}^2$, then $\rho_\mathcal{D}$ is a probability distribution similar to the case of persistence images [AEK*17]. Given two k-dimensional persistence diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$, we obtain two new sets $\mathcal{D}_1^*$ and $\mathcal{D}_2^*$ by augmenting $\mathcal{D}_1$ with the orthogonal projection of points of $\mathcal{D}_2$ on the diagonal and vice versa. For these two sets, the persistence Fisher kernel [LY18] is defined as

$$K_F(\mathcal{D}_1, \mathcal{D}_2) = e^{-t d_F(\mathcal{D}_1^*, \mathcal{D}_2^*)} \tag{8}$$

where $t > 0$ is a scalar parameter and d_F is the Fisher information metric defined as follows:

$$d_F(\mathcal{D}_1^*, \mathcal{D}_2^*) = \arccos\left(\int \sqrt{\rho_{\mathcal{D}_1^*}(x)\ \rho_{\mathcal{D}_2^*}(x)} \mathrm{d}x \right) \tag{9}$$

Comparing variants of persistence diagrams. Both persistence

landscapes and persistence images (as well as the persistence-based feature vectorizations [LWA*17]) can be used in machine learning algorithms such as SVMs under a Euclidean metric (*e.g.*, L^2 or L^p).

Definition 3 [Bub15, *p*-landscape distance] If λ_1 and λ_2 are the persistence landscapes corresponding to persistence diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$, the *p-landscape distance* is

$$\Lambda_p(\mathcal{D}_1,\mathcal{D}_2) = ||\lambda_1 - \lambda_2||_p. \tag{10}$$

Rieck *et al.* [RSL20b] defined a family of distances for Betti curves (also called the *persistence indicator functions*), as well as corresponding kernels in order to use Betti curve in machine learning algorithms. Zhao and Wang [ZW19] introduced a weighted-kernel for persistence images (WKPI), its induced distance, and a metric-learning framework to learn the weights (and kernel) from labeled data. The persistent homology transform (PHT) introduced by Turner *et al.* [TMB14] comes with a distance measure, referred to as the *PHT distance*, which captures similarity between shapes in shape classification. The inter-level set persistence hierarchies (ISPHs) [RSL17, RSL20b] are directed trees, whose similarity can be measured by the edit distance (see Sect. 4.2).

4.2. Comparing Reeb Graphs and Their Variants

A number of metrics have been proposed for Reeb graphs and their variants such as merge trees, including functional distortion distance [BGW14, BMW15], edit distance [BFL16, BLM20, SMKN20], interleaving distance [CCSG*09, MBW13, DSMP16, MS19], distances based on branch decompositions and matching [BYM*14, SSW14], and metrics for phylogenetic trees [CMR*13].

Functional distortion distances. Inspired by the Gromov-Hausdorff (GH) distance for measuring metric distortions, Bauer *et al.* [BGW14] introduced the *function distortion distance* for Reeb graphs. Let f and g be two real-valued functions on topological spaces $\mathbb{X}$ and $\mathbb{Y}$ (the technical requirements are tame functions), together with maps $\varphi : \mathbb{X} \to \mathbb{Y}$ and $\psi : \mathbb{Y} \to \mathbb{X}$. Let $\mathcal{G}_f$ and $\mathcal{G}_g$ be the two Reeb graphs. Define

$$C(\varphi,\psi) = \{(x,\varphi(x)) \mid x \in \mathcal{G}_f\} \cup \{(\psi(y),y) \mid y \in \mathcal{G}_g\}, \tag{11}$$

$$D(\varphi,\psi) = \sup_{(x,y),(x',y') \in C(\varphi,\psi)} \frac{1}{2}|d_f(x,x') - d_g(y,y')|. \tag{12}$$

$C(\varphi,\psi)$ captures the set of correspondences between $\mathcal{G}_f$ and $\mathcal{G}_g$ induced by maps φ and ψ.

Definition 4 [BGW14, Functional distortion distance] The *functional distortion distance* between two Reeb graphs, $d_{FD}(\mathcal{G}_f,\mathcal{G}_g)$ is defined to be

$$d_{FD}(\mathcal{G}_f,\mathcal{G}_g) = \inf_{\varphi,\psi} \max\{D(\varphi,\psi), ||f - g\circ\varphi||_\infty, ||g - f\circ\psi||_\infty\}. \tag{13}$$

Here, φ and ψ are all continuous maps between $\mathcal{G}_f$ and $\mathcal{G}_g$.

Edit distances. We begin with edit distances for trees, since contour trees and merge trees are inherently tree-based representations. Inspired by the edit distance from computational linguistics [RY98] that quantifies dissimilarities between strings, Zhang and Shasha [ZS89] introduced edit distance for ordered labeled trees by computing the minimum-cost of node operations (*i.e.* "relabel", "delete", and "insert") that transform one tree into another. Zhang *et al.* [ZSS92] extended edit distance to unordered labeled trees. Tree edit distances have been used in many applications [ZS89, RR92, KTSK00], including comparing topological structures such as merge trees. Rieck *et al.* [RSL17] proposed persistence hierarchies to related points in persistence diagrams, where tree edit distance-based dissimilarity is used to compare these hierarchies. Sridharamurthy *et al.* [SMKN20] extended constrained tree edit distance [Zha96] based on dynamic programming with suitable modifications applicable to merge trees and showed its implementation in a feature-driven analysis of scalar fields.

Definition 5 [SMKN20, Edit distance between merge trees] The *edit distance* between merge trees $\mathcal{T}_1$ and $\mathcal{T}_2$ is defined as

$$d_E(\mathcal{T}_1,\mathcal{T}_2) = \min_S\{\gamma(S)\}, \tag{14}$$

where S is a tree edit operation sequence from $\mathcal{T}_1$ to $\mathcal{T}_2$ that include edit operations such as "relabel", "delete", and "insert"; and γ is a cost function that assigns a non-negative real number to each operation.

Recently, Lohfink *et al.* [LWL*20] adapted the graph-theoretic notion of the *tree alignment*, which is similar to the edit distance mapping and is used to jointly visualize the contour trees from members of an ensemble.

Bauer *et al.* [BFL16, BLM20] introduced an *edit distance* between labeled graphs, and applied it to Reeb graphs.

Definition 6 [BFL16, Edit distance between labeled Reeb graphs, Definition 3.8] The *edit distance between labeled Reeb graphs* $(\mathcal{G}_f,l_f)$ and $(\mathcal{G}_g,l_g)$ is defined as

$$d_{EG}((\mathcal{G}_f,l_f),(\mathcal{G}_2,l_g)) = \inf_S\{\gamma(S)\}, \tag{15}$$

where S varies in a set of arbitrarily long sequences of edit operations necessary to transform $(\mathcal{G}_f,l_f)$ into $(\mathcal{G}_g,l_g)$, and $\gamma(S)$ is the cost of an edit sequence.

Interleaving distances. Algebraically, the interleaving distance arises from ε-interleavings of persistence modules; see [CCSG*09] for technical details. For topological descriptors, Morozov *et al.* [MBW13] defined an analog as the interleaving distance between merge trees. Inspired by L^∞-cophenetic metric introduced by Cardona *et al.* [CMR*13], Munch *et al.* [MS19] introduced an interleaving distance between labeled merge trees, which is the L^∞-distance between their induced matrices. Gasparovic *et al.* further studied interleaving distance intrinsic properties for the space of labeled and unlabeled merge trees, and used it to construct metric 1-centers for collections of labeled merge trees [GMO*19, YWM*20]. We describe the interleaving distance between merge trees as defined in [GMO*19], which was shown to be equivalent to the original in [TW19].

Given two merge trees $\mathcal{T}_f$ and $\mathcal{T}_g$ that arise from functions $f : \mathbb{X} \to \mathbb{R}$ and $g : \mathbb{Y} \to \mathbb{R}$, a δ-*good map* $\alpha : (\mathcal{T}_f,f) \to (\mathcal{T}_g,g)$ is a continuous map on the metric trees such that the following properties hold:

I. $\forall x \in |\mathcal{T}_f|$, $g(\alpha(x)) - f(x) = \delta$, where $|\mathcal{T}_f|$ denote the support (*i.e.* underlying space) of the tree;

II. $\forall w \in \mathrm{Im}(\alpha)$ with $x' := LCA(\alpha^{-1}(w)), f(x') - f(u) \leq 2\delta$ for all $u \in \alpha^{-1}(w)$;
III. $\forall w \notin \mathrm{Im}(\alpha)$, $\mathrm{depth}(w) \leq 2\delta$.

$LCA(v,w)$ denotes the *lowest common ancestor* of v and w in a tree. Intuitively, the δ-goodness means that the points that are mapped from $\mathcal{T}_f$ to $\mathcal{T}_g$ via α do not change their function values much.

Definition 7 [GMO*19, Interleaving distance between merge trees] The *interleaving distance* between merge trees $\mathcal{T}_f$ and $\mathcal{T}_g$ is defined as

$$d_I(\mathcal{T}_f, \mathcal{T}_g) = \inf\{\delta \mid \exists \delta\text{-good } \alpha : \mathcal{T}_f \to \mathcal{T}_g\} \tag{16}$$

Definition 8 [MS19, Interleaving distance between labeled merge trees] Given two labeled merge trees $(\mathcal{T}_f, \pi)$ and $(\mathcal{T}_g, \pi')$, where maps $\pi : [n] \to V(\mathcal{T}_f)$ and $\pi' : [n] \to V(\mathcal{T}_g)$ assign labels $[n] := [1, 2, \ldots, n]$ to the nodes of $\mathcal{T}_f$ and $\mathcal{T}_g$, their *interleaving distance* is

$$d_{IL}((\mathcal{T}_f, \pi), (\mathcal{T}_g, \pi')) = ||M(\mathcal{T}_f, \pi) - M(\mathcal{T}_g, \pi')||_\infty. \tag{17}$$

We use $M(\mathcal{T}_f, \pi)$ to denote the *induced matrix* of a labeled merge tree $(\mathcal{T}_f, \pi)$, which is the symmetric matrix $M \in R^{n \times n}$, and $M_{ij} = f(LCA(\pi(i), \pi(j)))$.

Additionally, Silva *et al.* [DSMP16] defined a sheaf-theoretic interleaving distance between a pair of Reeb graphs as the interleaving distance between their cosheaves, and proved that this distance is stable under perturbations of the input data; see [DSMP16] for technical details.

Distances based on branch decompositions or subtrees. Beketayev *et al.* [BYM*14] defined a distance d_{BR} between merge trees based on branch decompositions. They considered all branch decompositions of merge trees and found a minimum cost matching between them.

Definition 9 [BYM*14, Distance between merge trees based on branch decomposition] Given two merge trees $\mathcal{T}_f$ and $\mathcal{T}_g$, and all of their possible branch decompositions $B_{\mathcal{T}_f} = \{R_1^f, \ldots, R_k^f\}$ and $B_{\mathcal{T}_g} = \{R_1^g, \ldots, R_k^g\}$, the distance between $\mathcal{T}_f$ and $\mathcal{T}_g$ can be defined as

$$d_{BR}(\mathcal{T}_f, \mathcal{T}_g) = \min_{R_i^f \in B_{\mathcal{T}_f}, R_j^g \in B_{\mathcal{T}_g}} (\varepsilon_{\min}(R_i^f, R_j^g)), \tag{18}$$

where $\varepsilon_{\min}(R_i^f, R_j^g)$ is the lower bound for non-negative matching and removal costs of branch decompositions R_i^f and R_j^g.

Saikia *et al.* [SSW14] defined a comparative measure for the *extended branch decomposition graph* (eBDG) based on minimizing the cost of matching between sequences of trees formed by branch decomposition of merge trees. In other words, they compared all subtrees of a merge tree. Both the descriptor (eBDG) and the comparative measure are computed by dynamic programming. The authors also extended their work on eBDG to define a simple, histogram-based comparative measure for merge trees [SSW15]. Instead of overlaying branch decomposition trees obtained from the subtrees, they described every subtree with a feature vector, referred to as a histogram [SSW15]. To compare two histograms, they used the L^2-norm of the log-scaled bin values [SSW15]. Subsequently, Saikia and Weinkauf [SW17] proposed a global similarity measure for feature tracking in time-varying fields. Their similarity measure is an extension of [SSW14] that involves a combination of spatial overlaps and histogram comparisons.

Thomas and Natarajan [TN11] defined a comparison measure based on constructing and comparing hierarchical descriptors of the subtrees of contour trees, and claimed that such a descriptor is stable in the presence of noise.

Comparative measures for variants of Reeb graphs. A number of comparative measures are based on features or attributes derived from Reeb graphs and their variants.

Saggar *et al.* [SSGC*18] used the mapper graph to study similarities among time-varying fMRI data. Each time frame of the fMRI data is interpreted as a point in a high-dimensional space; and two time frames are considered similar if they are connected in the mapper graph. Hilaga *et al.* [HSKK01] constructed a multi-resolutional Reeb graph (MRG) based on geodesic distance, and designed a coarse-to-fine strategy to measure similarity between MRGs using the attributes of nodes in the MRGs. Biasotti *et al.* [BMM*03] defined a similarity measure based on error tolerant graph isomorphism on extended Reeb graph (ERG). Later, Barra and Biasotti [BB13] developed a similarity measure for ERGs by applying a Gaussian kernel to vertex and edge attributes. Wu and Zhang [WZ13] attached measures of similarities to contour tree branches for comparative analysis; such a measure is quantified based on contour overlaps.

Graph-based or tree-based comparative measures. Finally, comparative measures developed in biology or graph theory may be applicable for topological descriptors. Cardona *et al.* [CMR*13] defined a family of cophenetic metrics for comparing phylogenetic trees, which can be adopted as comparative measures for merge trees (*e.g.,* [MS19, GMO*19]). Tools developed for pairwise comparisons of graphs may be used for Reeb graphs and their variants; see surveys on graph distances [TITP19, WM20] and references therein. On the other hand, comparative measures for topological descriptors can be extended for general graphs as well. Dey *et al.* [DSW15] compared graphs via the persistence distortion distance. They compared a set of persistence diagrams constructed by defining scalar fields from various base points from the graphs. The sets are compared by Hausdorff distance, and individual persistence diagrams are compared by bottleneck distance.

4.3. Comparing Morse and Morse-Smale Complexes

A few papers have focused on comparative measures for Morse complexes and Morse-Smale complexes, most of which compare graphs derived from these complexes. This focus is not too surprising as a general form of stability for these complexes appears to be elusive.

Comparing graphs derived from complexes. Feng *et al.* [FHJB13] studied the problem of computing feature correspondences between two non-rigidly deformed surfaces using feature graphs, which are 1D skeletons of simplified Morse-Smale complexes. Feature graphs are compared using a minimum-cost graph matching algorithm. The authors observed (without proof) that such a feature graph is stable for surfaces differing by topology or by significant deformation.

Thomas and Natarajan [TN13] focused on detecting symmetry in scalar fields using *augmented extremum graphs*. They used the *geodesic distances* between extrema, and the *earth mover's distance* between histograms of selected seed regions.

In order to compare a pair of extremum graphs that may differ in the number of extrema and their adjunct relationships, Narayanan *et al.* [NTN15] introduced the notion of a *complete extremum graph*, which allows edges between all pairs of extrema in the graph. They then defined a distance between extremum graphs based on computing the maximum distortion of the vertex sets and edge sets between the graphs. Specifically, Narayanan *et al.* represented a complete extremum graph as an attributed graph. Each vertex $v \in V$ of the graph – identical to a vertex from an extremum graph – is assigned its persistence $p(v)$. Each edge $(u,v) \in E$ of the graph is assigned a cost $c(u,v)$ such that $c(u,v) \leq \min(p(u), p(v))$. The persistence of the global maximum is set to 1. A scalar function is normalized to have a range of $[0,1]$ to ensure that $0 \leq p(v), c(u,v) \leq 1$. The distance between extrema graphs is defined based on these vertex and edge attributes [NTN15]. Let $\mathcal{G}_f = (V, E_V)$ denote a complete extremum graph of f with vertex set V and edge set E_V. Given two complete extremum graphs $\mathcal{G}_f = (V, E_V)$ and $\mathcal{G}_g = (U, E_U)$, a map $h : V \to U$ is called ρ-*valid* for $\rho \in [0,1]$ if it is bijective and the edge distortion of corresponding edges is bounded by ρ. For a map h, let $D_h(\mathcal{G}_f, \mathcal{G}_g)$ denote the maximal distortion between the vertex set and edge set, that is,

$$D_h(\mathcal{G}_f, \mathcal{G}_g) = \max_{v \in V} |p(v) - p(h(v)| + \max_{(u,v) \in E} |c(u,v) - c(h(u,v))|. \tag{19}$$

Definition 10 [NTN15, Distance between extremum graphs] For a fixed ρ, the distance d_ρ between extremum graphs $\mathcal{G}_f$ and $\mathcal{G}_g$ is the minimum over all possible ρ-maps,

$$d_\rho(\mathcal{G}_f, \mathcal{G}_g) = \min_h \{D_h(\mathcal{G}_f, \mathcal{G}_g) \mid h \text{ is } \rho\text{-valid}\}. \tag{20}$$

5. Navigating the State of the Art in Visualization

The comparative measures introduced in Sect. 4 have enabled a wide variety of visualization tasks. We categorize these tasks based on whether they are applied primarily to single scalar fields, time-varying scalar fields, or ensembles. The visualization tasks are described in Sect. 6, Sect. 7, and Sect. 8. The definitions of several comparative measures have been motivated for the most part by specific tasks associated with visualization or interactive exploration. We discuss these comparative measures with a focus on their roles in enabling the visualization tasks. Table 1 presents a guide for navigating the state-of-the-art described in these sections. We have also released the list of all references covered in this survey via SurVis [BKW15], the visual literature browser, which is available at `https://git.io/Jt2Hq`.

6. Visualization Tasks for Single Fields

We begin by considering comparative measures between single scalar fields. The comparison of single fields has many applications in scientific visualization, and in many cases serves as a building block in comparing time-varying scalar fields and ensembles. It plays an important role in tasks such as symmetry detection and shape matching. The former finds applications in the visual analysis of biomolecules, and the latter is essential for comparative visualization and computer vision. In the context of single fields, comparative measures may support the comparison between two different fields or between sub-structures within a single field (*i.e.*, self-comparison).

An important application of self-comparison is symmetry detection, discussed in Sect. 6.1. Applications of comparisons between two fields include shape matching and retrieval, followed by matching shapes that are not represented by meshes, such as neuronal trees, see Sect. 6.2. We also discuss other visualization applications with single field comparisons in Sect. 6.3, such as parameter tuning for ray casting algorithms, graphs, and social networks. For comparative measures already summarized in Sect. 4, we focus on their applications in specific visualization tasks. For other comparative measures designed mainly for visualization, we introduce the measures on a high-level before discussing their associated tasks.

6.1. Symmetry Detection

Symmetry detection refers to the identification of repeating structures within a single scalar field f. The repeats are identified based on a comparative measure, which is applied to compare a scalar field with itself. A typical pipeline would first construct a topological descriptor $\mathcal{A}$ from f, simplify $\mathcal{A}$ to remove noise, explicitly or implicitly enumerate sub-structures of $\mathcal{A}$, and compare pairs of these sub-structures. A refinement step may be incorporated to reduce the number of such comparisons, by selecting a specific sub-structure as a query or by applying spatial overlap criteria. Symmetry detection is a first step in many visualization tasks such as query-based exploration, transfer function design, and linked volume editing, to name a few [TN14, MTN13]. Applications in molecular biology and allied fields have been demonstrated, where detecting repeating sub-structures in biomolecules is crucial to understand their shape and function. In Fig. 9, we see various symmetries identified within the Buckyball and vortex simulation datasets. In some cases, such as the vortex simulation data, the symmetries are not perceivable using a casual visual inspection.

6.1.1. Merge Trees

A rich set of techniques is available to detect symmetry by comparing merge trees and their variants, some of which are successful in discovering visually hidden symmetries. All papers described below showcase the utility of the method on CryoEM datasets, which consist of electron microscopy density maps of biomolecules.

Saikia *et al.* [SSW14] computed the extended branch decomposition graph (eBDG) using a (simplified) merge tree $\mathcal{T}$ as input. An eBDG contains a union of all branch decomposition trees (subtrees) of $\mathcal{T}$. To construct an eBDG, similarity scores are precomputed for all sub-trees against all other sub-trees using a combination of (normalized) volume and function differences. To detect self-similarity in a dataset, the eBDG is compared with itself. Since similarity scores are precomputed for all comparisons among subtrees, given a region of interest, the method can report all similar sub-structures and hence support real-time exploration of the data.

	Single scalar field			Time-varying scalar fields			Scalar field ensemble			
Topological structure	Symmetry detection	Shape retrieval	Other tasks	Feature tracking	Global structure changes	Space-time structures	Clustering and classification	Summarization	Uncertainty visualization	Interactive exploration
Critical points/ contours	[SWC*08] [TN14]			[SBS02] [WTGP10] [KRHH11] [RKG*11] [KHNH12] [KZHH12] [RKWH12] [DNN13] [LWM*17] [SPCT18] [VMN*19] [EMB*20] [NEFH20]			[FFST18]	[FFST18]	[MW14a] [GST14]	
Persistence diagram		[TMB14] [LWA*17] [ZW19] [HZLR20]			[RSL17] [SPCT18] [SPD*19]		[KVT19] [VBT20]			
Merge tree	[SSW14] [SSW15] [SMKN20]	[SMKN20]	[BYM*14]	[SW17]	[SSW14] [SMKN20] [LPYW21]			[YWM*20]	[GST14] [YWM*20]	[PDT*15] [YWM*20]
Contour tree/ Reeb graph	[TN11]	[HSKK01] [BMM*03] [ZBB04] [BB13]		[EHMP04] [SB06]		[BWP*10] [WBD*11]	[HSKK01] [ZBB04]	[LWL*20]	[WZ13]	
Morse-smale complex/ Extremum graph	[TN13]	[FHJB13]		[KRHH11] [RKWH12] [NTN15] [KEF*17] [SHC*19] [SHD*20] [RSL20a]	[NTN15]				[AMY*20]	
Other methods		[AC07] [DSW15]	[HHC*13] [RL16a]	[ARC20]	[EH04] [EHNP04] [SNN11] [SSGC*18]	[SSGC*18]				

Table 1: *Navigating surveyed papers based on topological descriptors vs. visualization tasks.*

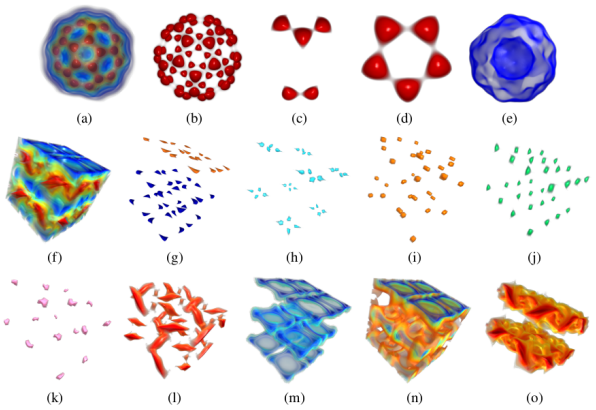

Figure 9: *Symmetric regions identified within two datasets: Buckyball (top row) and vortex simulation (middle and bottom row). (a) and (f) are full volume renderings, whereas the rest correspond to symmetric regions. Image reproduced from Thomas and Natarajan [TN11], cropped to show two datasets.*

This approach also addresses the problem of seed selection [TN13] and supports slicing the 3D field and isolating symmetries that are not visually evident in 3D.

In a follow-up work, Saikia *et al.* [SSW15] provided an alternate method for symmetry detection. Each sub-tree is augmented with a feature vector, namely the histogram given by intensity distribution among voxels of the sub-tree. This augmentation is computed together with the merge tree, and the resulting histograms (bin size 100) are compared using an L^2 norm. The similarity scores are computed by comparing all pairs of sub-trees and a distance matrix is then constructed. A user can pick a voxel and then select the corresponding feature as the region of interest. Entries from the corresponding row in the distance matrix are picked, with an option to vary the distance threshold to refine the matches based on how close they are to the query.

Sridharamurthy *et al.* [SMKN20] used tree edit distance (Definition 5) to detect symmetric structures. After computing the merge tree of the scalar field, a set of sub-trees is selected based on persistence rank, and the edit distance is calculated by comparing all pairs of sub-trees to construct a distance matrix. The method has limited utility as it does not support query-based similarity search, but can be used to detect symmetric structures by explicitly extracting sub-trees and comparing them. However, the method achieves results similar to those of Thomas and Natarajan [TN11], who used the entire contour tree.

All three methods suffer from instabilities. Saikia *et al.* [SSW14] provided examples for false negatives and performed a perturbation analysis. The authors suggested a combination of volume and function differences as edge weights to alleviate the instability issue. Although histogram-based approaches [SSW15] are robust for small perturbations, they cannot be a substitute for complicated branching. A case for which two trees with identical histograms was provided in the discussion. The number of bins used for the histogram can affect the final results. The authors suggested the possible use of non-linear binning in future work. Sridharamurthy *et al.* [SMKN20] achieved stability by merging saddle points into a multi-saddle based on an approach proposed earlier [TN11]. The merging is directed by a stabilization threshold that determines

which critical points are merged. This approach works in practice, but with no theoretical guarantees.

6.1.2. Contour Trees and Reeb graphs

Thomas and Natarajan [TN11] defined a similarity measure for symmetry detection based on constructing and comparing hierarchical descriptors constructed from sub-trees of contour trees. The comparison is based on the max-weight matching of the hierarchical descriptors. The algorithm computes groups of symmetric regions and refines them in a post-processing step. The measure can be used to detect many kinds of symmetries that are not apparent via visual inspection. A stabilization parameter is used to handle instabilities. The method does not consider geometric information and thus cannot capture symmetries based on geometry. The method is used in applications such as symmetry-aware transfer function design and isosurface extraction.

In a subsequent work, Thomas *et al.* [TN14] exclusively used geometric information to extract symmetric structures in multiple scales. Schneider *et al.* [SWC*08] presented a similarity browser for comparing scalar fields, where similarity is defined as the relative overlap of the largest contours, and relevant contours are extracted by querying edges of a contour tree.

6.1.3. Extremum Graphs and Morse Complexes

Thomas and Natarajan [TN13] augmented a simplified extremum graph with edges that directly connect saddles, providing a good approximation of geodesic distance between pairs of extrema. Seed points are chosen from which the symmetric regions are grown using geodesic distance between extrema. Iteratively, the seeds are combined to form super-seeds, and symmetric regions are extracted via a region growing process. The method avoids computing matches between sub-trees or sub-structures and instead relies on geodesic distance; thus, it is robust in handling noise when compared to [TN11]. The method is used in applications such as proximity-aware volume visualization, linked volume editing and multi-mode volume rendering. Seed selection is a critical step, and symmetry detection depends on selection of a meaningful set of seeds. Seed selection and simplification depend on user-defined thresholds.

6.2. Shape Matching and Retrieval

Shape matching and retrieval is another important problem studied within the fields of computer graphics, computer vision, and visualization that can be addressed by comparative analysis of topological descriptors. Shape matching deals with comparing 3D shapes that are stored in the form of meshes. The solutions should be able to detect similarity/dissimilarity between shapes irrespective of view direction, scale, orientation, pose variation, and other transformations. The models often contain multiple attributes in addition to geometry and topology. Quantifying similarity is necessary to explore large databases of shapes. Biasotti *et al.* [BCBB14] surveyed existing methods from the perspective of maps between spaces. We restrict the discussion here to methods for which the comparative measure is based on topological descriptors. Although recent advances in learning-based methods do provide better results, topological descriptors provide scope for improvement as additional feature vectors. Shape matching applications depend crucially on the choice of the scalar field/Morse function defined on the shapes. For example, the average geodesic distance from a randomly chosen set of vertices of the mesh is often used in the literature. Once the scalar field is computed, the rest of the pipeline is similar to symmetry detection. For each shape, a suitable topological descriptor is constructed, and the descriptors are compared to get a similarity score. The scores are used in a query-based system to match and retrieve shapes. Fig. 10 shows one application of shape matching where correspondences of various parts of the shape mesh are found.

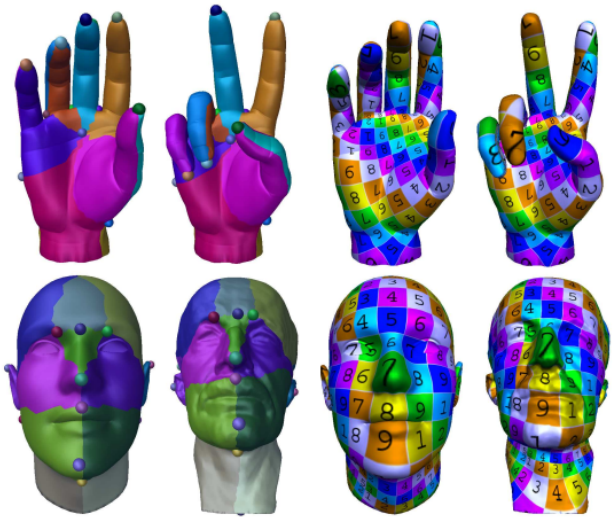

Figure 10: *Shape-matching application: Point-and-patch correspondences followed by the texture transfer results of the hand and head data in different poses are shown. Image reproduced from Feng et al. [FHJB13] with different alignment.*

6.2.1. Critical Points and Persistence diagrams

To study mesh similarity, Hajij *et al.* [HZLR20] used persistence diagrams computed on the lower star filtration of the eigenfunctions of the Laplacian that store important geometric information. They alleviated the need for higher-dimensional persistence diagrams and showed that using a single eigenfunction itself has more discriminatory power than metric-based approaches. They used only the 0-dimensional persistence diagrams, which are easy to compute. They compared the diagrams using bottleneck distance followed by 2D t-SNE projection of the distance matrix. They showcased their results on 60 meshes divided into 6 categories, available from Sumner and Popović [SP04, SP21]. The results were shown for Fielder's vectors. They planned to combine signatures from multiple eigenfunctions and extend their method from triangulated meshes to point clouds and graphs.

Li *et al.* [LWA*17] compared neuronal tree shapes by vectorizing neuron structures based on topological persistence, unlike traditional shape matching where the shapes are meshes. They proved that such a persistence-based signature is more effective in capturing the global and local structure than simple statistical summaries. They also proved, using a certain descriptor function, that

a persistence-based signature contains more information than the classical Sholl analysis. The persistence diagram of the trees is computed with the descriptor function as the scalar field. The points in the diagram are then converted to a 1D density function by first assigning appropriate weights followed by converting the set of 1D points using a kernel estimate. The density function is then vectorized, which can be compared using standard norms like L^1 or L^2. Li *et al.* performed experiments on neuronal trees, used geodesic descriptor as the scalar field, showcased various tasks such as comparison and clustering, and analyzed results using classification accuracy. The space of the neurons was visualized. Future work involves building a database of descriptors and experiments using multiple descriptor functions.

Zhao and Wang [ZW19] further used the Weighted Persistence Image Kernel (WKPI) to compare neuronal trees and classify them in a similar fashion as Li *et al.* [LWA*17]. They contrasted the classification accuracy using existing learning approaches and provided additional results for other graph data.

Persistence homology transform (PHT) was introduced by Turner *et al.* [TMB14] as a topological descriptor of surfaces in 3D or curves in 2D. A unit vector induces a height field on the surface, for which a set of d persistence diagrams can be computed. PHT is then defined as a map from all possible direction vectors (points on a unit sphere) to space of persistence diagrams. The authors proved that this transform is injective, and thus a metric defined on the space of persistence diagrams can be used to define a metric on the set of shapes. In practice, Wasserstein distance is used for comparing individual persistence diagrams. The distance is approximated by sampling the unit sphere to get a finite number of directions. The technique is demonstrated by applying it for classification and clustering of a set of shapes from MPEG-7 shape silhouette database [Sik01]. Seven class of objects with 20 examples each totaling 1400 objects are chosen, and 0-th PHT is computed using 64 evenly spaced directions. Then, distance computation considers various rotations and takes the minimum. The objects are then projected into 2D or 3D using multi-dimensional scaling. Turner *et al.* reported that the classes are well separated.

6.2.2. Merge Trees

Sridharamurthy *et al.* [SMKN20] used tree edit distance (Definition 5) to showcase shape matching using TOSCA non-rigid world dataset [BBK21]. The shapes are in different poses and consist of both humanoid and non-humanoid shapes. Average geodesic distance field is calculated on the meshes, followed by a persistence simplification using a threshold of 1% of the scalar field range. A distance matrix (DM) is computed by comparing all pairs of shapes. Each collection appears within a block of the DM, comprising low distance values irrespective of the pose. Some similarity across the blocks is also observed for humanoid shapes.

6.2.3. Contour Trees and Reeb Graphs

Hilaga *et al.* [HSKK01] defined multi-resolution Reeb graphs (MRG) and a similarity measure to compare them. An approximation of geodesic distance based on Dijkstra's algorithm on the edge lengths is used as the scalar function upon which MRG is built. The similarity is computed by matching attributes defined on the MRGs. The MRGs are constructed on the shapes using a continuous scalar function, and a coarse-to-fine strategy is used to compare the graphs and compute the similarity. The similarity is used to find the best matches for a given query shape. The experimental setup of Hilaga *et al.* used 230 models collected from three sources: Viewpoint Models, 3DCAFE, and Stanford University models. They chose an object as the key and reported similar objects as retrieved by the method. The method depends on the resolution and two other parameters, range (μ_n) and weight (w), which are typically set to 0.5. They reported running times and mentioned that incorporating geometric information, extension to handle morphing, and application to pose estimation as potential future work.

Inspired by the work of Hilaga *et al.* [HSKK01], Zhang *et al.* [ZBB04] presented an algorithm to match volumetric functions based on multi-resolution dual contour trees. The matching is again based on weighted sum of attributes, normalized volume, function range, and Betti numbers of bounded contours. Electrostatic potential and electron density distributions within biomolecules (PDB data repository) are represented as scalar fields and compared. The method is robust with respect to both rigid body transformations and small perturbations to the scalar field. The paper reported the results of applying the matching algorithm on 242 protein chains assembled from different families. A clustering extension helps distinguish between different protein families. The authors proposed the use of sophisticated shape attributes and combined both electrostatic potential and electron density to improve the classification accuracy. They also described the scope for improving the tree matching algorithm.

Biasotti *et al.* [BMM*03] defined a similarity measure based on error-tolerant graph isomorphism on an extended Reeb graph (ERG). An ERG is defined with respect to Euclidean distance from a point or with respect to integral geodesic distance. The measure depends on the choice of the function used to construct ERG and is shown to be a metric. Barra and Biasotti [BB13] used a kernel-based comparison of ERGs for shape retrieval. Kernels are defined on both vertex and edge attributes and computed by comparing all paths stemming from the two graphs being compared. Both papers discussed various applications on shape matching and retrieval contest (SHREC) datasets, with detailed analysis of precision and recall.

6.2.4. Extremum Graphs and Morse Complexes

Feng *et al.* [FHJB13] studied the problem of computing corresponding features between two non-rigidly deformed surfaces, which is a key component in any shape-matching application. They compared their approach with other feature correspondence detection methods that involve geodesic and diffusion distances on various shapes with different values of weighting parameter. They also showcased the utility of their method in matching surfaces with non-zero genus, and showed examples of cross-parameterization and texture transfer.

6.2.5. Other Descriptors

Allili and Corriveau [AC07] provided a method to compare shapes by comparing the Morse shape descriptors (MSDs), which are topological descriptors defined on smooth manifolds. The MSD

for an n-dimensional manifold is defined as a set of $(n+1)$ matrices with appropriate discretization of the scalar function. The similarity is measured as the weighted sum of distances between collections of MSD associated with contours. The main limitation is that the MSD is dependent on the Morse function used. No tools are available to select appropriate Morse functions for a given class of shapes. The descriptor allows multi-scale analysis due to discretization. The method is applied to 2D shapes only. The authors discussed that there is room to improve precision and recall, and to develop theoretical foundations to facilitate the choice of appropriate Morse functions.

Dey *et al.* [DSW15] used persistence distortion distance to compare surface meshes of different geometric models, some of them being the same models but in different poses. They used the geodesic distance from base points as the scalar field. The input is a δ-sparse subsample of the 1-skeleton of the meshes constructed by randomized decimation. They proved that the error in the estimation of the distance is at most 12δ. Although the distance can be non-zero between a graph with itself (this depends on the choice of base points), the distances are small, confirming the stability.

6.3. Other Visualization Tasks for Single Fields

Beketayev *et al.* [BYM*14] used distances between merge trees (Definition 9) to analyze the tuning of a ray tracing algorithm on a multi-core system as studied by Bethel *et al.* [BH12]. Focusing on three parameters, work block width, height, and concurrency level, the authors recorded the performance for two datasets with the same parameter space but slightly different algorithms based on a ray selection method. The similarity of the datasets implies no significant difference is caused by the choice of ray selection method, which they confirmed by the small distance between them.

Rieck and Leitte [RL16a] proposed a measure for comparing different clusterings of multivariate data and for studying the clustering quality. Given a multivariate dataset, the method first computes the Vietoris-Rips complex and defines a suitable shape descriptor (a scalar field) on the point cloud. For a given clustering, the global version of the proposed measure is defined as the ratio of the total persistence sum of all clusters to the total persistence of the scalar field. This ratio captures the loss of features due to the clustering. They defined a local variant as well to capture the quality of the cluster. Although the measure is not a true comparative measure, it does enable a comparative analysis. The local measure is visualized as an attribute overlaid on the cluster, and the global measure for all clusterings is visualized as a network where similar clusterings are located close to each other.

Topological descriptors have also been used for comparative analysis and visualization of discrete structures such as graphs and social networks [RFLL17, RL16b].

7. Visualization Tasks for Time-Varying Fields

Time plays a fundamental role in many processes. Prominent examples are physical models that simulate phenomena such as clouds formations in climate or weather modeling [DNN13, KEF*17, EMB*20], vortex shedding in flow simulations [KZHH12, RSL20a], the simulation of combustion and burning structures [BWP*10, SNN11], or molecular dynamics simulations [SB06]. An example from medicine is time-varying measurements of the brain activity [SSGC*18]. In all these examples, efficient analysis of the resulting dynamic data plays an increasing role. Key visualization and analysis tasks are the identification and tracking of features to understand the evolution of structural properties, find periodicity, or detect explicit events. Comparative measures play a primary role in this process. To this end, topological data analysis has proven to be a fundamental tool, and a large number of related publications are available, which will be discussed in this section along with topological descriptors being used.

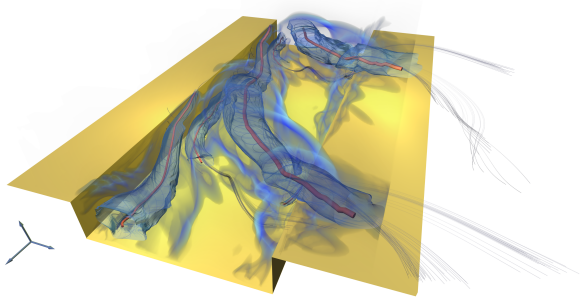

Figure 11: *Time-dependent vortex core lines (red) and their associated vortex regions shown as transparent surfaces (blue) obtained from tracking minima and extremum structures in the acceleration magnitude filed of the 2D flow over a cavity. Volume rendering of the acceleration and a few path lines provide the context. Image reproduced from [KRHH11].*

In our context, the underlying assumption is that *time* is a continuous variable. However, typically, time-dependent data is available as a set of temporal snapshots. Generally, methods analyzing such data can be categorized depending on the treatment of the temporal dimension [Pos03]. The first possible approach is to analyze the data per time-slice and then compare the results. One can thereby find methods that explicitly track local features by solving an explicit correspondence problem (Sect. 7.1) and methods that consider a global distance between the topological structures in one time-slice as a whole (Sect. 7.2). The third group of methods defines features as entities in the space-time domain where no explicit tracking is necessary (Sect. 7.3). In this report, we make an explicit distinction between time-varying fields and ensembles of scalar fields, where we consider ensembles to be a collection of scalar fields that arise from different parameter settings. Ensembles that are collections of time-varying fields are mainly discussed in the current section.

7.1. Feature Tracking and Event Detection

Tracking is mostly a two-step process in which topological features are extracted in each time slice and then matched solving a correspondence problem. Therefore, feature tracking deals not only with the evolution of features but also with the identification of structural changes (events) between time steps, which are appearance

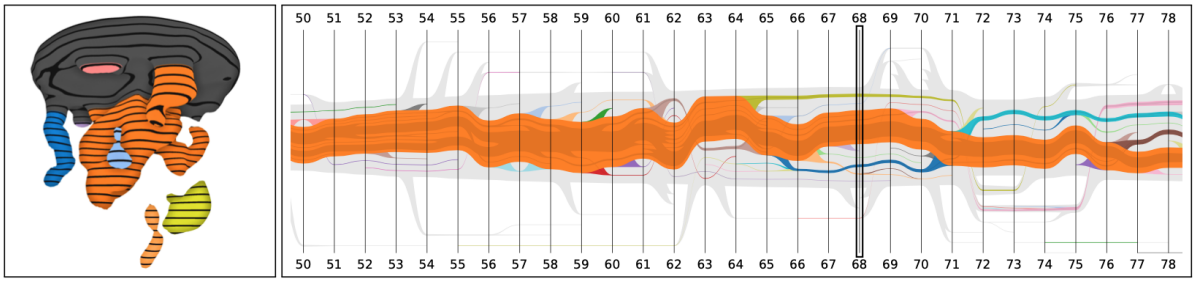

Figure 12: *Nested tracking graph for one ensemble member of the viscous finger dataset. The graph illustrates the nesting hierarchy of contours across levels. The x-axis represents time and the y-axis is used to minimize edge crossings. The image shows the interface of the visual analytics framework consisting of a DVR window (left) and the interactive nested tracking graph (right). Image reproduced from [LWM*17].*

(birth), disappearance (death), merging, or splitting of features. An optional simplification step using topological persistence is applied before computing the matching in the case of large data.

The most frequently used topological descriptors (Sect. 3) are critical points [RKWH12, SPCT18] or contours [LWM*17], especially when it comes to real-world applications. However, some approaches have also been proposed for tracking the contour trees [SB06], Reeb graphs [EHMP04], Morse cells [SHD*20], or extremum graphs [KEF*17]. Correspondence criteria use distance measures in the spatial domain or attribute space (Sect. 4). These are often based on application-specific heuristics or feature overlap. Some methods also establish a correspondence building on an explicit temporal interpolation or optical flow [VMN*19]. The respective papers are described in the following sections.

7.1.1. Critical Point Tracking

Weinkauf *et al.* [WTGP10] described the dynamic behavior of critical points as streamlines of a higher-dimensional vector field in space-time, so-called *feature flow fields*. The method relies on a continuous temporal interpolation of the field and applies numerical integration to generate the tracks. Originally, the method was applied to critical points in vector fields, but the concept is also applicable to critical points in scalar fields. Reininghaus *et al.* [RKWH12] tracked critical points in scalar fields to facilitate a discrete feature flow field. The feature tracks are gradient lines in the discrete field. Practically, feature tracking establishes a correspondence between extremal points in consecutive time steps using a forward and backward *Morse matching*. This means an extremal point from time-step t falls into the descending (resp. ascending) manifold of an extremal point in time-step $t+1$ and the other way round. The method explicitly keeps track of merge and split events. By tracking the persistence value over time, Reininghaus *et al.* introduced a temporal importance measure for the tracked features. The tracking itself is inherently local and fast. This method has been successfully applied to vortex tracking by Kasten *et al.* [KHNH12, KZHH12], thus providing an abstract tracking graph. Reininghaus *et al.* [RKG*11] also used the approach to track critical points in the scale-space. In an extension of this work, the descending manifolds of the tracked minima enclosed by a subset of the extremum structures are tracked and interpreted as vortex regions [KRHH11] (Fig. 11). The correspondence of the extremum structure is inherited from the extremal points. A similar approach was used by Engelke *et al.* [EMB*20, NEFH20] to track multi-centered cyclones defined as sets of critical points in a pressure field.

Soler *et al.* [SPCT18] introduced an extremal point tracking algorithm that solves the correspondence problem by minimizing the sum of the distances between matched pairs. The distance between two extrema is defined as a weighted sum of (i) the difference of the corresponding persistence pairs as used in the Wasserstein distance (Definition 2) and (ii) their geometric distance in the domain. The authors call the resulting metric *lifted Wasserstein metric*. For an efficient computation of their tracking, they proposed a modification of the Kuhn-Munkres algorithm [Mun57]. They demonstrated the usefulness of the approach on a few analytical datasets and examples from flow visualization (vortex tracking) (Fig. 13). As a future extension, they proposed to also include other attributes in the distance function assigned to the topological features. In a follow-up paper [SPD*19], the method was adapted to the analysis of the viscous finger by describing the dynamic process observed at the interface between fluids of different viscosity.

Valsangkar *et al.* [VMN*19] used optical flow to study the temporal evolution of cyclones. Cyclonic centers are defined as local minima of the mean sea level pressure field. Candidate tracks are computed from an optical flow field, which then is clustered in a moving time window to obtain the final tracks. The tracks are visualized as a smooth curve interpolating the vertices of a track. The method supports the identification of merge and split events.

7.1.2. Contour, Contour Trees and Merge Tree Tracking

Shamir *et al.* [SBS02] proposed a progressive isosurface tracing algorithm that predicts the contour at time step $t+1$ based on the contour at time step t. In a later paper, Sohn and Bajaj [SB06] extended this approach by proposing a contour-tree based feature tracking method. The method has some similarities with the methods introduced in Sect. 4.2. Specific to this method, Sohn and Bajaj keep track of the contours (points on the contour tree) for all isovalues, which is summarized in a topology change graph (TCG).

A temporal correspondence between contours is established in the case of *significant* overlap over both their sublevel and superlevel sets. To be able to track contours for iso-values, they facilitated the coherence of contours when changing iso-values. Additional "critical points" are added to the contour trees where two super level sets of the two trees "collide" and a new overlap region is created. The same process is done for sublevel sets. The significance value is a conceptual parameter in this approach. They demonstrated the utility of their method in tracking turbulent vortex structures and the binding of oxygen to hemoglobin molecule using time-dependent electron density maps.

Doraiswamy *et al.* [DNN13] described a framework for the exploration of cloud systems at various scales in space and time based on infrared (IR) brightness temperature images. The framework automatically extracts cloud clusters as contours for a given temperature threshold. To identify the threshold, the persistence diagram is used. The movement of the cloud system is tracked using the optical flow between the pair of IR brightness temperature images.

Lukasczyk *et al.* [LWM*17] introduced nested tracking graphs as a hierarchical representation of feature tracks highlighting temporal events. Features are defined as contours and correspondence is established by contour overlap. They applied the approach to data from finite point set methods, computational fluid dynamics, and cosmology. The method has been integrated into a visual analytic approach [LAS*17] for tracking of viscous fingers (Fig. 12).

Saikia and Weinkauf [SW17] used subtrees of the merge tree for temporal feature tracking. Subtree similarity is based on a combination of spatial overlap and χ^2-distance between local histograms. The result is a directed acyclic graph with nodes that represent features and the edges that record the similarity information. The final feature track is then computed as the shortest path in a graph, and thus considers not only the similarity between consecutive time steps but also global similarity. They applied their method to track primary and secondary vortex structures in the 3D time-dependent flow behind a cylinder (Fig. 14) by considering the Okubo-Weiss criterion as the scalar field.

Time-varying Reeb graphs for the analysis of continuous space-time scalar data were proposed by Edelsbrunner *et al.* [EHMP04]. The nodes of the Reeb graphs for consecutive time steps, which correspond to the critical points of the scalar field, are connected using Jacobi curves in space-time (Sect. 3).

7.1.3. Extremum Graph and Morse Complex Tracking

Schnorr *et al.* [SHD*20] presented a two-step tracking approach for space-filling features (objects whose union covers the domain). The work was motivated by the analysis of dissipation elements, which are defined as Morse-Smale cells (Sect. 3.4). This problem is especially challenging since the number of overlapping cells in consecutive time steps can be very high. In a first step, the solution of a weighted (normalized volume overlap) bi-partite matching problem generates a 1:1 assignment of cells. This mapping is assumed to result in the most plausible matches. In a second step, additional edges are generated to allow for merge and split events by solving a maximum-weight independent-set problem. Since the method is computationally expensive, the authors later published an algorithm that approximates the tracking achieving a substantial speed-up [SHC*19].

Tracking of extremum structures (Sect. 3.4) or skeletons is an unstable process since the resulting structures typically are sensitive to small variations in the data. Kuhn *et al.* [KEF*17] dealt with this challenge by proposing a space-time clustering approach for the extremum graphs from the individual time steps. The clusters are visualized as a distance field isosurface of the extremum graphs in space-time. The method has been developed in an application dealing with ash cloud tracking in a 2D domain. Rieck *et al.* [RSL20a] analyzed the evolution of skeletons, which are extracted applying iterative thinning to binary images. To improve the temporal coher-

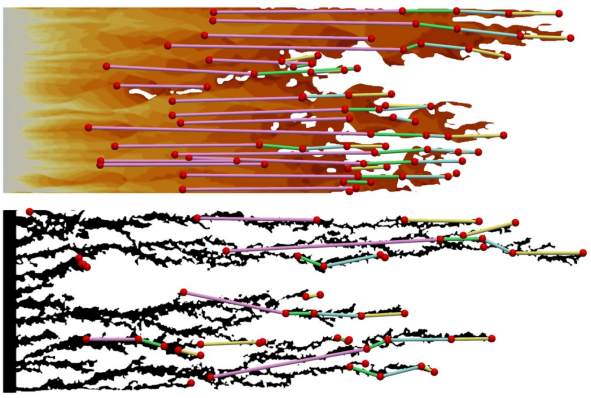

Figure 13: *Critical point trajectories based on optimal matchings. The geometrical coherence of fingers allows to use a lifted version (that gives importance to the y-coordinate of fingers) of the Wasserstein metric to correctly track the evolution of persistence pairs. Image reproduced from [SPD*19].*

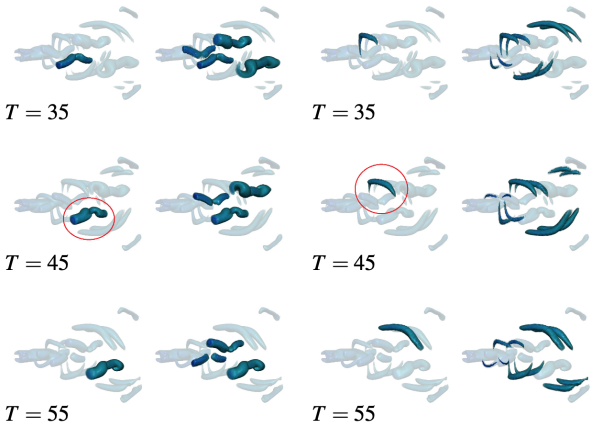

Figure 14: *A primary and secondary vortex structure have been selected at $T = 45$ and tracked backwards and forwards in time. Their tracks have been used to find spatio-temporally similar structures in the entire dataset. Image reproduced from [SW17].*

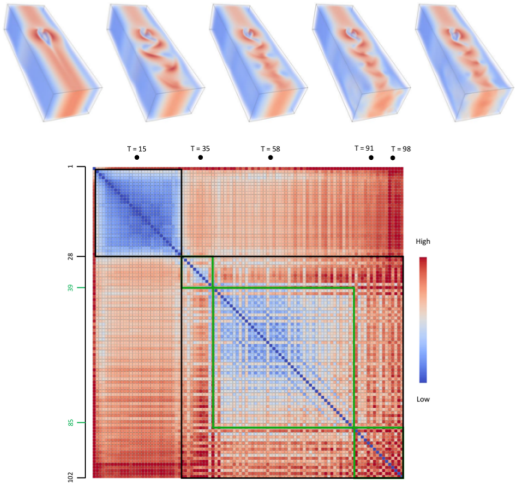

Figure 15: *Tree edit distance matrix for all time steps of the 3D von Kármán vortex street. Columns corresponding to the selected time steps (images in the top row) are highlighted in the matrix. Interesting patterns are highlighted using black and green boxes. Image reproduced from [SMKN20].*

ence, they introduce a novel persistence concept. They applied the method to the viscous fingers dataset.

Narayanan *et al.* [NTN15] used the distance d_ρ (Eq. 10) to compare extremum graphs based on maximum common sub-graphs and to track turbulent vortex structures.

7.2. Structural Change Detection

A global comparison of datasets derives a pair-wise distance measure between time steps, which often are displayed in a distance matrix or similarity plots over time (Fig. 15). Typical visualization tasks are outlier, periodicity, or event detection. Periodicity detection refers to identifying structures that repeat over time. Event detection in this context refers to finding sudden changes in the overall structure of the data, which results in specific patterns in the distance matrix. Principally, all distance measures introduced in Sect. 4 can be used for this purpose. However, only a few of those have been used in visualization applications concerned with time-varying data.

Saikia *et al.* [SSW14] used eBDG, which is computed from merge trees, and compared these trees using a dynamic programming algorithm to detect periodicity in the 2D Benard-von Kármán vortex street dataset. A similar application is showcased by Narayanan *et al.* [NTN15].

Sridharamurthy *et al.* [SMKN20] utilized the tree edit distance in applications such as detecting periodicity (similar to [SSW14, NTN15]) and computing temporal summaries. They used a distance matrix to detect key events and summarized the Benard-von Kármán vortex street dataset as shown in Fig. 15.

Edelsbrunner *et al.* [EHNP04] introduced a global comparison measure for a set of $k \leq d$ scalar functions defined on a common d-dimensional manifold, and applied it to the study of time-varying functions. The comparison measure is defined as an integral of the norm of the wedge product of the k derivatives. For the case of $k = d = 2$, the measure is related to the Jacobi set of the two scalar functions (Sect. 3.5) and equals the integral of the persistence of all critical points of one function restricted to the level sets of the other. The measure can be computed in linear time. It has been applied to combustion simulation data. A time plot of the comparison measure between the fuel and a variable representing the progress of the combustion helps identify the key stages of ignition, burning, and extinction. The paper also describes a local version of the measure obtained by restricting the computation to an infinitesimally small subdomain. The measure is symmetric but does not satisfy the triangle inequality. The measure is also not scale-invariant and is therefore not suitable to compare the similarity of pairs of functions. An extension of the local measure to compare gradients of $k > d$ scalar functions was applied to climate science and combustion studies [SNN11].

In a few applications, the comparison of persistence diagrams was also applied for the analysis of temporal data (Sect. 4.1). Rieck *et al.* [RSL17] identified periodicity in time-varying temperature data from a climate application by comparing all pairs of time steps using inter-level set persistence hierarchies (ISPHs) and visualizing the resulting distance matrix. In an application paper, Soler *et al.* [SPD*19] compared time-varying viscous finger datasets from ensemble simulation runs based on time-varying persistence diagrams. Hajij *et al.* [HWSR18] employed the distances between persistence diagrams to visually detect structural changes in time-varying graphs. To quantify the structural difference between two time instances of a social network, they computed the bottleneck and Wasserstein distances between their persistence diagrams. The same framework is potentially applicable to the study of time-varying scalar fields.

Agarwal *et al.* [ARC20] proposed a similarity measure for the analysis of time-varying multi-fields. Each multi-field is represented as a Multi-resolution Reeb Space (MRS) that is approximated as a series of joint contour nets (JCNs) at various levels of data-range discretization. Between the nodes of adjacent resolutions, a parent-child relationship is introduced. The computation of the similarity measure between two MRSs is a two-step process. In the first step, a list of matching pairs from the nodes of the respective MRSs is established, moving from the coarser to the finer resolution Reeb spaces. Two nodes are matched if they have the same resolution level, the parents have been matched, and they are topologically consistent. In the second step, the similarity of the MRSs is computed as a sum of an attribute-based similarity between the nodes following ideas from Zhang *et al.* [ZBB04] and Hilaga *et al.* [HSKK01] (Sect. 4.2). The visualizations show the abstract graphs and plots of the similarity measure over time. The paper focuses on the introduction of the method. Its use is demonstrated for time-varying multi-field data from computational physics. The similarity measure is not specific to the analysis of time-varying data. The correspondence principle could also be used for explicit feature tracking.

Given a set of merge trees $\mathbb{T} = \mathcal{T}_1, \ldots, \mathcal{T}_k$, Li *et al.* [LPYW21] were interested in finding a basis set $\mathbb{S}$ of merge trees such that

each tree in $\mathbb{T}$ can be approximately reconstructed from a linear combination of merge trees in $\mathbb{S}$. A set of high-dimensional vectors can be sketched via matrix sketching techniques such as principal component analysis and column subset selection. Li *et al.* developed a framework for sketching a set of merge trees that combines the Gromov-Wasserstein framework [CN20] with techniques from matrix sketching. They demonstrated the applications of the framework in sketching merge trees that arise from time-varying data in scientific simulations (*e.g.*, flow simulations, and the Red Sea eddy simulation from the IEEE Scientific Visualization Contest 2020), where their method is used to find good ensemble representatives and to identify outliers *w.r.t.* a chosen basis set. The same framework is potentially applicable for the ensembles of scalar fields discussed in Sect. 8.

7.3. Space-Time Structures

An alternative to feature tracking is to define topological structures directly in the space-time domain, in this case, the correspondence must not be established explicitly and no distance measures are required. Instead, one must assume an explicit temporal interpolation.

Weber *et al.* [WBD*11] tracked subsets of isosurfaces using the Reeb graph. The work was motivated by the analysis of burning regions in simulated flames. Such regions are defined as parts of the temperature isosurfaces, where the fuel consumption rate is above a certain threshold. Boundary surfaces separating burning and non-burning regions are extracted in a two-phase contouring operation: a temperature iso-volume in 4D space-time is computed, from which the fuel-consumption-rate isosurface is constructed. Using time as a filter function, the Reeb graph of this boundary surface represents the evolution of burning regions. In a follow-up paper [BWP*10], this graph was embedded in an exploration framework and augmented with statistical attributes (Fig. 16).

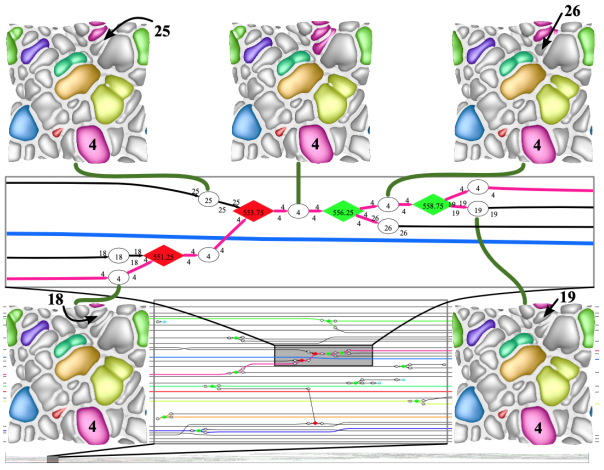

Figure 16: *Reeb-graph based tracking graph of burning cells with corresponding segmentations. Round nodes correspond to cells explicitly segmented by the Morse complex, diamonds to topological events between time steps. Red signifies a merge, green a split, and turquoise a birth/death event. Image reproduced from [BWP*10].*

Edelsbrunner *et al.* [EH04] proposed to compare individual time steps using Jacobi sets (see Sect. 3). Although this approach provides a solid theoretical framework, the results are often very complex and hard to interpret when applied to real-world data.

Saggar *et al.* [SSGC*18] proposed a pipeline for the analysis of time-series of fMRI (functional magnetic resonance) data to explore the dynamical organization of the brain. Each timeframe is interpreted as a point in a high-dimensional space (number of pixels). After a dimension-reduction step using t-SNE [vdMH08], mapper construction (See Sect. 3) is applied to build a shape graph, where the nodes represent sets of timeframes. The results are visualized as an abstract graph augmented with aggregated node attributes linked to special visualizations. A distance between time frames can be defined via distances in the shape graph. The authors stated that future work is needed to reduce the computational costs of the approach.

8. Visualization Tasks for Ensembles

In this section, we focus on topological descriptors applied to scalar field comparisons for ensembles. An ensemble of scalar fields, in the context of this report, is a collection of scalar fields indexed on a set of parameters with no temporal relationship between them. Such a collection of scalar fields often results from multiple simulation runs obtained by varying the input parameters.

The typical overall analysis task in an ensemble is to relate the observed features in the output scalar fields to the input parameter space. Topological descriptors play an important role and thus can reduce the task of understanding the space of scalar fields to a more tractable task of exploring the space of their corresponding topological descriptors. For example, instead of searching for patterns in the high-dimensional space of scalar fields, the job simplifies to finding patterns in the space of merge trees. Key tasks include clustering, classification, outlier detection, feature detection, summarization, and computation of structural statistics.

8.1. Clustering and Classification

Applying and adapting techniques from pattern recognition and statistical analysis, such as clustering and classification to analyze ensembles, has seen active research in recent years. However, these ideas have been around for a long time. The work by Hilaga *et al.* [HSKK01], which used Reeb graph for shape matching and retrieval, is an early example of how topological approaches can work in the context of information retrieval, clustering, and classification. Another early topology-guided comparison for classification is the

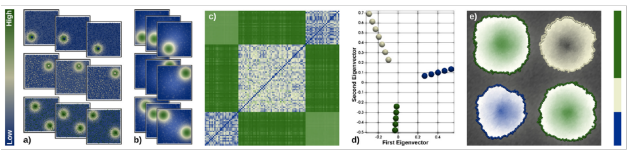

Figure 17: *Persistence atlas for an ensemble of a synthetic scalar fields. (a) The input scalar fields. (b) Persistence maps. (c) Distance matrix for the persistence maps. (d) Projection of persistence maps to 2D. (e) The persistence atlas. Image reproduced from [FFST18].*

use of multi-resolution dual contour trees proposed by Zhang *et al.* [ZBB04]. They demonstrated applications to electrostatic potential and electron density fields for protein structures, resulting in a classification of proteins into categories. In both papers, a similarity matrix was used to capture distances between members of the ensemble; refer to Sect. 6 for a more detailed discussion of these papers.

Favelier *et al.* [FFST18] described a framework called *persistence atlas* that uses a *persistence map* for clustering and trend variability analysis of ensembles. The persistence map is a derived scalar field that captures the spatial distribution and density of extrema in the input field weighted by their importance. Persistence is used as the importance measure. This derived map is then used as a signature of the scalar field. Eigenanalysis-based dimensionality reduction is applied to project the ensemble members to 2D where clusters are identified. Trend and spatial variability analyses are done per cluster by computing mandatory critical points [GST14]. The utility of the framework is demonstrated using simple synthetic data (Fig. 17), multiple flow simulation ensembles, and ensembles originating from climate simulations.

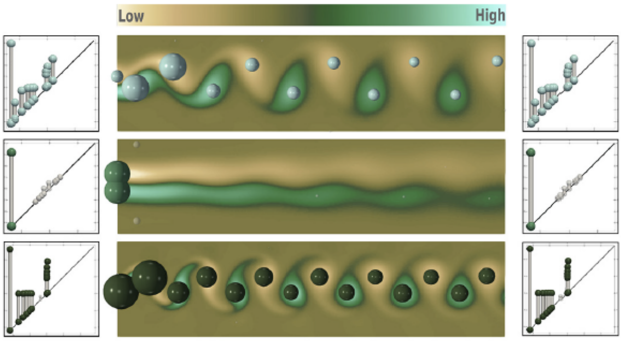

Figure 18: *Three clusters identified in the flow data ensemble correspond to different viscosity regimes. The Wasserstein persistence diagram barycenters computed using interruptible approach are shown on the left and the exact barycenters on the right. There is little difference between the diagrams, and the performance difference between these approaches is huge. Image reproduced from [VBT20], cropped to show three of the five clusters.*

Vidal *et al.* [VBT20] pursued a more direct approach to clustering of persistence diagrams, adapting the k-means algorithm based on Wasserstein distance. k-means is an iterative algorithm that proceeds in two steps. First, the point assignment step assigns each point to one cluster based on its distance to the cluster centroids. The second step updates the centroids. The mean persistence diagram of a set of persistence diagrams is computed as the Fréchet mean under the Wasserstein distance metric $d_{p=2}$. This requires repeated computation of pairwise distances between diagrams and the computation of Fréchet mean. If done naively, the computation is prohibitively costly and not time efficient for realistically sized data ensembles. The authors proposed a progressive algorithm for computing the Fréchet mean or discrete Wasserstein barycenters in a time-interruptible manner. This approach leads to improvements in runtime while producing similar quality results as shown via applications to ensembles studied earlier by Favelier *et al.* [FFST18]. Fig. 18 shows their results for a flow data ensemble.

The drawback of using k-means is the requirement of the number of expected clusters k in the ensemble. Kontak *et al.*. [KVT19] addressed this issue in their recent work in which they suggested finding the optimal number of clusters based on a minimization of established statistical score functions such as Akaike information criterion and Bayesian information criterion.

8.2. Summarization

Another approach for providing an overview of a scalar field ensemble is to summarize the set of scalar fields within a single abstract structure. For this task, topological structures have recently been shown to be very useful.

Lohfink *et al.* [LWL*20] described a technique for combining contour trees of multiple scalar fields of an ensemble in a single layout called fuzzy contour tree, which provides a summary of the ensemble. They use tree alignment, an idea similar to edit distance mappings, to identify similarities across multiple contour trees and to obtain a layout that can represent all trees simultaneously. A heuristic algorithm is used for computing the tree alignments for a given similarity measure. The approach is applied to the visualization of two scalar field ensembles: 2D flow around a cylinder and a heated cylinder ensemble. The utility of this approach is demonstrated well for these examples. However, the scalability of the approach for feature-rich data is not clear.

Yan *et al.* [YWM*20] proposed to use interleaving distances (Definition 8) to study a structural average of an ensemble of labeled merge trees. Such a structural average, referred to as a *1-center tree*, minimizes its maximum distance to any other tree in an input ensemble. They used global and local structural consistency measures between the input and the 1-center tree to encode uncertainty; see Fig. 19 for an example. Different heuristics are used to compute the labeling between pairs of merge trees to compute the 1-center tree. Their work demonstrated the application of a nice theoretical framework to compute structural averages of merge trees. In the supplementary material, the authors applied their framework to study neuron morphology. Their proposed consistency measure can be used to understand structural variations among an ensemble of neuron-cell-induced merge trees with respect to their 1-center, where such an ensemble arises from different reconstructions of the same neuron cell.

Finally, we would like to mention that the persistence atlas [FFST18] framework discussed earlier also provides a summary of an ensemble. The mandatory critical point regions identified for each cluster can be embedded in the spatial domain. See Fig. 17(e). However, this approach to summarization does not scale for large ensembles with complex features.

8.3. Interactive Exploration

Another significant aspect in the analysis of ensembles is to understand the relationship of the input parameters to the observed features in the scalar field instances, which is often explored fully

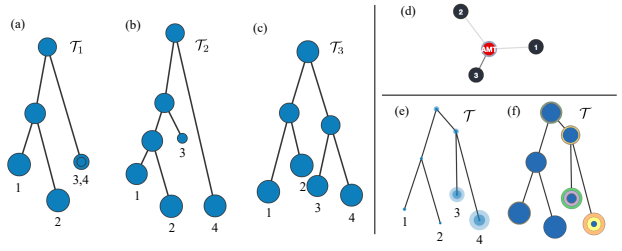

Figure 19: *An example of a structural average of three input merge trees and a visualization of its uncertainty. Vertex consistencies for each ensemble member are encoded using the radius of the glyph in (a), (b), and (c). Graduated circular glyphs encode variational (e) and statistical (f) vertex consistencies for 1-center trees by applying graduated glyphs to minimum, first quartile, median, third quartile, and maximum across all ensemble members. A summary plot (d) shows the interleaving distance between each input tree and the 1-center tree. Image generated using the visualization tool of [YWM*20].*

through an interactive visualization tool. The work by Yan *et al.* [YWM*20] provides some interactive exploration capabilities but it is restricted to exploring the space of merge trees.

A more appropriate example in this context of parameter space exploration using topological comparison methods is the work by Poco *et al.* [PDT*15] exploring differences in species distribution models. They computed a locality-aware correspondence between similar extrema of two scalar fields that represent two species distribution models. The extrema of the two scalar fields are represented as a bipartite graph, and the correspondence is computed as a maximum weight matching. The weight of each edge is a product of two terms, a functional similarity term and a spatial distance term. After computing the above maximum topology matching between the extrema, the topological similarity is computed as the minimum amount of simplification required to obtain a perfect bipartite matching. Additionally, a functional similarity is computed from the perfect bipartite matching as the maximum function difference between the matched extrema pairs. Species distribution models are high-dimensional scalar functions, where the dimensions of the domain correspond to predictor variables of the model algorithm. The topological and functional similarity measures are integrated into a visual exploration tool with typical statistical plots and other domain-specific charts. The tool facilitates detailed inspection of selected ensemble members with a focus on identifying the regions of similarity and differences between the high dimensional scalar functions.

8.4. Multi-field Data Comparison

Scalar field ensembles can also be considered as a single multi-field. Thus, they may be analyzed using topological techniques for multi-field analysis. Here, we discuss a few results in this direction. Additionally, we also mention initial work on pairwise comparison of two multi-fields, which in turn can be used for the analysis of time-varying multi-field ensembles.

Huettenberger *et al.* [HHC*13] extended the notion of extremal points to multi-fields based on Pareto optimality and Pareto dominance. The regions of consensus and disagreement among the ascending and descending manifolds of the extrema of the constituent scalar fields are identified. Pareto-extremal regions represent barriers across which all the constituent scalar fields cannot be jointly increased or decreased. These regions are joined by ascending and descending sets where the gradients of the scalar fields agree. Huettenberger *et al.*compared this approach against other topological methods for multi-field analysis like Jacobi sets and joint contour nets. The utility of the method is demonstrated on a simulated 2D flow with randomly perturbed input parameters, specifically to visualize the joint extremal structures of multiple realizations of the λ_2 field for vortex identification.

Liebmann and Scheuermann [LS16] introduced a topological description that represents a set of Gaussian-distributed scalar fields as a whole. A set of singular patches serves as a counterpart to critical points. These patches are assigned a probability as an attribute. The patches are defined by a classification of the local scalar value configuration in a point's neighborhood. Also, an adjacency relation between these patches is established, resulting in an abstract graph representation. In the proposed pipeline, the correlation matrix plays an important role in a first dimension reduction step (PCA). The analysis takes place in a high-dimensional space spanned by the largest eigenvectors of the correlation matrix. The graph is visualized using a force-directed layout to provide a first overview. Edge properties are encoded using lengths, thickness, and color. Furthermore, node properties, like patch probability and the topological type, are represented using glyphs. The spatial regions of the patches are shown with transparency, indicating the accumulated probability of all patches involved.

Both approaches discussed above are interesting ideas seeking to extend the well-established approaches of topological analysis for scalar fields to the largely unexplored domain of multi-field analysis. The wide popularity of contour trees and Reeb graphs in the visualization community resulted in a study of their generalization to multi-fields. The joint contour net (JCN) proposed by Carr and Duke [CD13] captures changes in specific subsets, called slabs, of the domain when varying multiple scalar fields simultaneously. They can be interpreted as a discrete approximation of the Reeb space [EHP08]. The main idea is to apply a quantization of the values in the range of the multi-field resulting in a structure that is amenable to computation. JCNs are also closely related to Mapper [SMC07].

Comparison of multi-fields. An approach proposed by Agarwal *et al.* [ACN21] for comparing multi-fields suggested using fiber component distributions. First, the JCN is computed for the given multi-field. Next, the frequency distribution is approximated by partitioning the range space into bins and counting the number of fiber components within each bin. A similarity score between two multi-fields is then simply computed as the L^p distance between their corresponding fiber component frequency distributions. The comparison measure finds application to computational chemistry, where it is used for identification of stable Pt-CO bonds using electron density multi-field data. The comparison measure is also used for identifying nuclear scission events.

In a follow-up work, Agarwal *et al.* [ARC20] extended the idea of comparison using multi-resolution Reeb graphs [HSKK01] and contour trees [ZBB04] to multi-fields as well. They proposed multi-resolution Reeb space of a multi-field and computed a similarity score as a weighted average of the scores comparing node attributes, such as volume, range, number of connected components, and degree. Refer to Sect. 7.2 for a more detailed discussion. This approach was also applied on the Pt-CO and the nuclear scission case studies with better results.

8.5. Uncertainty Visualization

In the context of ensemble data analysis, the study and visualization of uncertainty [BHJ*14] or variability in a set of scalar fields or their corresponding topological structures are crucial for drawing any meaningful conclusions from the data. We describe some selected approaches in this direction below.

Wu and Zhang [WZ13] proposed an exploration framework for uncertain scalar fields. Their main goal was to show the variance in contours and topology. The core of the visualization and analysis is the contour tree of the field average considered as a pivot tree. In the visualization, this tree is augmented with uncertainty-variability glyphs. For every contour of the *mean tree*, an average data-level uncertainty is attached to the tree branches in a ribbon-like fashion. For the contour comparison, they also introduced a branch correspondence concept, considering the branch with the largest overlap as most similar and computed an average difference of the contours. The abstract mean tree is overlaid with different variability measures and used to interact with the 3D spatial visualization. In 2D, the uncertainty glyphs are also visualized in the spatial representation. Two trees can be compared side-by-side or as an overlay. They showcase the framework via applications to uncertainty visualization for weather simulation and brain data. In the work of Yan *et al.* [YWM*20], a different notion of average merge tree is computed together with visual encodings of uncertainty information.

Mihai and Westermann [MW14a] described an approach for finding a feature distribution across ensemble instances. The features they considered were defined by level sets and critical points. They proposed methods for classification of critical points with respect to different notions of stability. Two indicator functions, the gradient and Hessian, are computed at all vertices of the data. The first indicates the likelihood of the existence of a critical point, the second reveals the tendency of its type. Both indicator functions are interpreted as independent multivariate random variables at each grid point. Confidence intervals of the gradient are computed at each vertex and if it contains zero, then a glyph is rendered at the vertex for visualization. They demonstrated their approach using climate simulation data.

Gunther *et al.* [GST14] introduced the notion of mandatory critical points that can be interpreted as the common topological denominator of uncertain scalar fields. These are regions and intervals where there is at least one critical point in any realization of the ensemble. Based on this concept, they also introduced the notion of mandatory merge trees. A simplification strategy enables multi-scale visualizations. This approach was effectively demonstrated on two flow datasets, Kármán vortex street and heated cylinder, and two more ensemble datasets from astronomy and climate science. Athawale *et al.* [AMY*20] utilized mediatory critical points of [GST14] within their pipeline to explore uncertainty visualization of an ensemble of Morse complexes. They introduced three types of statistical summary maps – the probabilistic map, the significance map, and the survival map – to characterize the uncertain behaviors of gradient flows of Morse complexes.

9. Desirable Properties of Comparative Measures

We now survey desirable properties of a comparative measure, denoted as $d(\mathcal{A}_1, \mathcal{A}_2)$, between a pair of topological descriptors $\mathcal{A}_1$ and $\mathcal{A}_2$. For each comparative measure d reviewed in Sect. 4, we study its properties surrounding *metricity*, *stability*, *discriminativity*, and *computational complexity*. We give a systematic classification of surveyed comparative measures based on these properties, see Table 2. There are a number of other properties that remain under-explored. For example, Bauer *et al.* [BLM20] considered a distance for Reeb graphs to be *universal* if it provides an upper bound to any other stable distance. Carrière and Oudot [CO17] considered a distance for Reeb graphs to be *intrinsic* if it can be realized by a geodesic. Gasparovic *et al.* [GMO*19] considered this property for distance between merge trees. It will be interesting to investigate these additional properties for a number of comparative measures in future works.

In this section, for discussions on computational complexity: n and m are the numbers of critical points in each topological descriptor (*w.l.o.g.*, assume $m \leq n$); N is the number of pixels in a persistence image; M denotes the number of vertices in the domain; l is density vectorization parameter; γ_{max} is the length of the maximal shortest path; I_ε is the search range for ε_{min}; B is the number of bins in a histogram; and t is the number of branches in a branch decomposition.

9.1. Comparative Measures for Persistence Diagrams

We describe properties associated with the bottleneck distance d_∞, the p-Wasserstein distance d_p, and the p-landscape distance Λ_p. We also discuss properties associated with a few other comparative measures for persistence diagrams.

Bottleneck distance. The space of persistence diagrams can be equipped with the bottleneck distance d_∞ [CSEH07]. Recall a persistence diagram is a multi-set of points in the extended plane $\overline{\mathbb{R}}^2$. In the most general case, d_∞ is an extended pseudometric [Oud17, Page 51]. d_∞ is a metric when the persistence diagrams are locally finite multi-sets in $\overline{\mathbb{R}}^2$; see [Oud17, Page 51] and [CDSGO16, Theorem 4.10]. d_∞ is proven to be stable [CSEH07] *w.r.t.* small perturbations of the input function.

Theorem 1 (Stability of d_∞ [CSEH07]) Let $\mathbb{X}$ be a triangulable space with continuous tame functions, $f, g : \mathbb{X} \to \mathbb{R}$. Then the persistence diagrams satisfy

$$d_\infty(\mathcal{D}_f, \mathcal{D}_g) \leq ||f - g||_\infty. \tag{21}$$

Measures	Citation	Notation	Metric	Stability	Discriminative	Complexity
Comparing persistence diagrams and their variants						
Bottleneck distance	[CSEH07]	d_∞	extended peudometric	Yes	baseline n/a	$O(n^{1.5}\log(n))$
p-Wasserstein distance	[CSEHM10]	d_p	extended peudometric	Yes	Yes	$O(n^3)$
p-landscape distance	[Bub15]	Λ_p	metric	Yes	No	$O(n^2)$
Persistence scale-space kernel	[RHBK15]	K_S	n/a	Yes	unknown	$O(n^2)$
Persistence weighted Gaussian kernel	[KFH17]	K_G	n/a	Yes	unknown	$O(n^2)$
sliced Wasserstein distance	[CCO17]	SW	extended peudometric	Yes	Yes	$O(n^2\log(n))$
Persistence Fisher kernel	[LY18]	K_F	n/a	Yes	unknown	$O(n^2)$
Lifted Wasserstein	[SPCT18]	—	conj. Metric	unknown	unknown	$O(n^3)$
WKPI distance	[ZW19]	—	pseudometric	Yes	unknown	O(N)
PHT distance	[TMB14]	—	metric	Yes	unknown	$O(n^3)$
L^p distance between persistence vectors	[LWA*17]	—	metric	unknown	unknown	$O(\max(n,l))$
Comparing Reeb graphs and their variants						
Functional distortion distance	[BGW14]	d_{FD}	extended pseudometric	Yes	Yes	NP-hard
Edit distance between merge trees	[SMKN20]	d_E	metric	We conj. No	conj. Yes	$O(n^2)$
Edit distance between labeled Reeb graphs	[BFL16]	d_{EG}	extended pseudometric	Yes	unknown	We conj. NP-hard
Interleaving distance between merge trees	[MBW13]	d_I	metric	Yes	Yes	NP-hard
Interleaving distance for labeled merge trees	[GMO*19]	d_{IL}	metric	Yes	We conj. Yes	$O(n^2)$
Interleaving distance between Reeb graphs	[DSMP16]	d_{IG}	extended pseudometric	Yes	unknown	NP-hard
Distance based on branch decompositions	[BYM*14]	d_{BR}	We conj. Yes	We conj. No	We conj. Yes	$O(n^5\log(I_\varepsilon))$
Distance based on histograms for merge trees	[SSW15]	—	metric	unknown	unknown	$O(n^2B)$
Distance based on tree alignment	[LWL*20]	—	unknown	unknown	unknown	$O(n^2)$
Distance based on subtrees of contour trees	[TN11]	—	unknown	unknown	unknown	$O(t^5\log t)$
Distance between extended Reeb graphs	[BMM*03]	—	metric	unknown	unknown	unknown
Kernel between extended Reeb graphs	[BB13]	—	unknown	unknown	unknown	$O(\gamma_{max}^4+2n^2\log(n))$
Comparing Morse complexes, Morse-Smale complexes and their variants						
Distance between extremum graphs	[NTN15]	d_ρ	metric	unknown	unknown	NP-hard
Similarity between Morse-Smale complexes	[FHJB13]	—	n/a	unknown	unknown	unknown
Similarity between Morse-Smale cells	[SHD*20]	—	n/a	unknown	unknown	NP-hard
Other comparative measures						
Distance between Morse shape descriptor	[AC07]	—	unknown	unknown	unknown	$O(n^2)$
Distance between persistence maps	[FFST18]	—	metric	unknown	unknown	$O(n^2M)$
Distance between fiber component distributions	[ACN21]	—	metric	unknown	unknown	unknown

Table 2: *Desirable properties for comparative measures surveyed in this paper.* ***Citation*** *indicates when a measure is first introduced.* ***Notation****: not all comparative measures are given a mathematical notation in this survey.* ***Complexity****: n is the number of critical points in a topological descriptor; N in the number of pixels in a persistence image; M denotes the number of vertices in the domain; l is density vectorization parameter; γ_{max} is the length of the maximal shortest path; I_ε is the search range for ε_{min}; B is the number of bins in a histogram; and t is the number of branches in a branch decomposition. Colors encode yes (green), no (pink), conjecture (purple, abbreviated as conj.), n/a (not applicable, white), and unknown (blue).*

A topological space is *triangulable* if there is a (finite) simplicial complex with homeomorphic underlying space [CSEH07]; and *tameness* is a technical condition that ensures f and g to be well behaved. Many existing works use d_∞ as a baseline comparative measure to determine the discriminativity of a newly introduced measure.

The main challenge of computing d_∞ between persistence diagrams is to find an optimal bijection η between them. Such a task can be viewed as a bipartite graph matching problem, which has been studied for decades [HK73, Ber79, Mun57]. Inspired by an auction algorithm [Ber79] and Hopcroft-Karp algorithm [HK73], Kerber *et al.* [KMN17] utilized the geometric structure of data into bijection matching and significantly improved the computation of both d_∞ and d_p in both runtime and memory consumption, where d_∞ can be computed in $O(n^{1.5}\log n)$ time.

p-Wasserstein distance. d_p is an extended pseudometric; it is a metric [EH10, Page 184] when the persistence diagrams are locally finite. Furthermore, the set of persistence diagrams equipped with d_p is shown to be a complete and separable metric space [CSEHM10]. d_p is also stable for a reasonably large class of functions [CSEHM10].

Theorem 2 ([CSEHM10]) Let $f,g:\mathbb{X}\to\mathbb{R}$ be tame Lipschitz functions on a metric space whose triangulations grow polynomially with constant exponent j. Then, there are constants C and $k>j$ no smaller than 1 such that, for every $p\geq k$,

$$d_p(\mathcal{D}_f,\mathcal{D}_g)\leq C\cdot||f-g||_\infty^{1-\frac{k}{p}}. \tag{22}$$

As p goes to ∞, d_p approaches d_∞ by defining the minimum of the maximum edge weight over all perfect matchings [KMN17]. Therefore, d_p is more discriminative than d_∞,

$$d_\infty(\mathcal{D}_f,\mathcal{D}_f)\leq d_p(\mathcal{D}_f,\mathcal{D}_g). \tag{23}$$

The exact computation of d_p needs minimum bipartite matching, which requires $O(n^3)$ effort. However, faster approximation algorithms are available for computing d_p using the geometric techniques of Kerber *et al.* [KMN17].

p-landscape distance. Λ_p is a metric [BD17]. It enjoys a form of stability under the same condition as for the stability of d_p in [CSEHM10]. Specifically, the persistence diagram is stable *w.r.t.* the p-landscape distance if $p > k$ and $\mathbb{X}$ has bounded degree-k total persistence.

Theorem 3 ([Bub15], Theorem 16) Let $\mathbb{X}$ be a triangulable, compact metric space that implies bounded degree-k total persistence for some real number $k \geq 1$, and let f and g be two tame Lipschitz functions. Then, for all $p \geq k$,

$$\Lambda_p(\mathcal{D}_f, \mathcal{D}_g)^p \leq C \cdot ||f-g||_\infty^{p-k}. \tag{24}$$

Here, the constant C is related to the Lipschitz constants of f and g. Λ_∞ is also shown to be stable *w.r.t.* to the L^∞ norm.

Theorem 4 ([Bub15], Theorem 12) Given real-valued functions $f, g : \mathbb{X} \to \mathbb{R}$ on a topological space,

$$\Lambda_\infty(\mathcal{D}_f, \mathcal{D}_g) \leq ||f-g||_\infty. \tag{25}$$

d_∞ is shown to be more discriminative than Λ_p for $p = \infty$ [Bub15].

Theorem 5 ([Bub15], Theorem 13)

$$\Lambda_\infty(\mathcal{D}_f, \mathcal{D}_g) \leq d_\infty(\mathcal{D}_f, \mathcal{D}_g) \tag{26}$$

Λ_p also contributes as a lower bound on the p-Wasserstein distance d_p [Bub15, Corollary15]. The above theorems suggest that Λ_p is not discriminative *w.r.t.* the baseline d_∞.

Bubenik and Dłotko [BD17] developed algorithms to compute persistence landscapes and the landscape distance Λ_p. The former takes $O(n^2)$, where n is the number of birth-death pairs, and the latter requires $O(P)$, where P is the maximum number of critical points of two input persistence landscapes. Since $P = O(n)$, we conclude the complexity is $O(n^2) + O(P) = O(n^2)$.

We now discuss properties associated with kernels on persistence diagrams.

Persistence scale-space kernel. K_S is 1-Wasserstein stable [RHBK15], that is, it is upper bounded by p-Wasserstein distance when $p = 1$ (denoted as $d_{p=1}$). Given two persistence diagrams (including points on the diagonal) $\mathcal{D}_1$ and $\mathcal{D}_2$, we have

Theorem 6 ([RHBK15], Theorem 2)

$$||\Phi_\sigma(\mathcal{D}_1) - \Phi_\sigma(\mathcal{D}_2)||_{L_2(\Omega)} \leq \frac{1}{2\sqrt{\pi\sigma}} d_{p=1}(\mathcal{D}_1, \mathcal{D}_2), \tag{27}$$

where the L_2-valued feature map $\Phi_\sigma : \mathcal{D} \to L_2(\Omega)$ at scale $\sigma > 0$ of a persistence diagram $\mathcal{D}$ is defined as $\Phi_\sigma(\mathcal{D})$, with $\Omega \subset \mathbb{R}^2$ denoting the closed half plane above the diagonal.

K_S can be computed in $O(m \cdot n)$ time, where m and n denote the cardinality of the multi-sets $\mathcal{D}_1$ and $\mathcal{D}_2$ (not counting the diagonal), respectively; *w.l.o.g.* assuming $m \leq n$, the complexity can be simplified as $O(n^2)$.

Persistence weighted Gaussian kernel. The distance defined by the reproducing kernel Hilbert spaces (RKHS) norm for the persistence weighted Gaussian kernel (PWGK) K_G satisfies a form of stability [KFH17] *w.r.t.* the bottleneck distance.

Theorem 7 ([KFH17], Theorem 9) Let $\mathbb{M}$ be a triangulable compact subspace in $\mathbb{R}^d$, $X, Y \subset \mathbb{M}$ be finite subsets, and $p > d+1$. A persistence diagram $\mathcal{D}$ can be vectorized via the map, $E_{k_G} : \mu^w_{\mathcal{D}} \mapsto w(x)k_G(\cdot, x)$, where w is a weight defined in Eq. 4 and $\mu_{\mathcal{D}} := \sum_{x \in \mathcal{D}} w(x)\delta_x$ for the Dirac delta measure at x. Now, given persistence diagrams $\mathcal{D}_X$ and $\mathcal{D}_Y$, and their vectorized representations $E_{k_G}(\mu^w_{\mathcal{D}_X})$ and $E_{k_G}(\mu^w_{\mathcal{D}_Y})$ of the RKHS,

$$||E_{k_G}(\mu^w_{\mathcal{D}_X}) - E_{k_G}(\mu^w_{\mathcal{D}_Y})||_{\mathcal{H}_{k_G}} \leq L(k,w) d_\infty(\mathcal{D}_X, \mathcal{D}_Y), \tag{28}$$

where $||\cdot||_{\mathcal{H}_{k_G}}$ represents the norm in RKHS, and $L(k,w)$ is a constant independent of X and Y.

Given two persistence diagrams, the computation of K_G involves $O(n^2)$ evaluations of $e^{-\frac{\|p-q\|^2}{2\sigma^2}}$, where n is the number of points in the larger persistence diagram [KFH17]. Kusano *et al.* also investigated faster approximation algorithms using the random Fourier features [RR07].

Sliced Wasserstein kernel and sliced Wasserstein distance. The sliced Wasserstein distance SW (Eq. 5), which is designed for the sliced Wasserstein kernel K_W, is stable and discriminative as it preserves the metric between persistence diagrams [CCO17]. In particular, SW is proved to be equivalent to $d_{p=1}$ (the p-Wasserstein distance for $p = 1$), which implies that SW is as discriminative as $d_{p=1}$.

Theorem 8 ([CCO17], Theorem 3.3) Let persistence diagrams $\mathcal{D}_1$, $\mathcal{D}_2$ have cardinalities bounded by N, then

$$\frac{d_{p=1}(\mathcal{D}_1, \mathcal{D}_2)}{2 \times (1 + 2N(2N-1))} \leq SW(\mathcal{D}_1, \mathcal{D}_2, M) \leq 2\sqrt{2} d_{p=1}(\mathcal{D}_1, \mathcal{D}_2). \tag{29}$$

The time required to compute K_W is $O(n^2 \log(n))$.

Persistence Fisher kernel. The persistence Fisher kernel K_F is proven to be stable [LY18], because its induced squared distance $d^2_{K_F}$ is bounded by the Hilbert norm of the difference between two corresponding mappings d_F.

Theorem 9 ([LY18], Lemma 4.1) Let $\mathcal{D}_1$, $\mathcal{D}_2$ be two bounded and finite persistent diagrams, then

$$d^2_{K_F}(\mathcal{D}_1, \mathcal{D}_2) \leq 2t d_F(\mathcal{D}_1, \mathcal{D}_2), \tag{30}$$

where t is a parameter of K_F and $d^2_{K_F}$ is defined as

$$d^2_{K_F}(\mathcal{D}_1, \mathcal{D}_2) = K_F(\mathcal{D}_1, \mathcal{D}_1) + K_F(\mathcal{D}_2, \mathcal{D}_2) - 2K_F(\mathcal{D}_1, \mathcal{D}_2) \tag{31}$$

The time required to compute K_F between a pair of persistence diagrams is $O(n^2)$. An accelerated version with fast Gauss transform can compute it in $O(n)$ time [LY18].

Other comparative measures using persistence diagrams. Soler *et al.* [SPCT18] proposed a lifted Wasserstein comparative measure for time-varying persistence diagrams. It is used to compare time-varying viscous finger datasets from ensemble simulation runs [SPD*19]. The lifted Wasserstein measure is computed by a

sparse persistence matching by augmenting points in the diagram with geometrical considerations, and the authors claimed (without proof) that the distance is a metric. They also claimed that it enhances geometrical stability of the feature tracking application in comparison to the Wasserstein distance. Its implementation uses an unbalanced Kuhn-Munkres algorithm for solving assignment problem, which takes $O(\min(m,n)^2 \max(m,n))$, where m and n are the number of points in the persistence diagrams (which is roughly the number of critical points from the corresponding scalar fields). Assuming m and n are roughly the same, we simplify this to be $O(n^3)$, which is the running time of the Hungarian algorithm.

The WKPI (weighted-kernel for persistence images) distance d_ω [ZW19] is a pseudometric induced by the inner product on a Hilbert space. It is proven to be stable *w.r.t.* small perturbation in persistence diagrams as measured by the p-th Wasserstein distance for $p = 1$ [ZW19, Theorem 3.4]. The runtime to compute the distance is $O(N)$, for N being the number of pixels in each persistence image.

The persistent homology transform (PHT) defines a distance metric between shapes or surfaces [TMB14]. Computing the PHT of an object needs the persistence diagrams of the height function from various directions. The PHT distance averages the distance between diagrams of two input objects. Turner *et al.* proved that the map from a space of well-behaved shapes into the space of PHTs is an injective map [TMB14]. This map comes with a sense of stability as PHT of a finite simplicial complex is proved to be continuous [TMB14, Lemma 2.1]. Turner *et al.* [TMB14] used the Hungarian algorithm to compute the distances between two persistence diagrams in each of the directions. It takes $O((n+m)^3)$, where $n+m$ is the number of off-diagonal points in the two diagrams combined; assuming $m \approx n$, we simplify this to be $O(n^3)$. The L^p distance between persistence vectors [LWA*17] is by definition a metric (on the space of persistence vectors). Its computation takes $O(\max(n,l))$, where n is the number of critical points and l is the parameter used to vectorize the density function.

Finally, there are a number of recent developments that aim to vectorize information from persistence diagrams to be interfaced with machine learning in a scalable way; most of which have not had direct applications in visualization. Noticeably, Hofer *et al.* [HKNU17] computed parametrized projections of persistence diagrams that can be learned during neural network training. Moor *et al.* [MHRB20] introduced a "topological loss term" for autoencoders that relate the topology of the data space with that of the latent space. The *PersLay* [CCI*20] framework learns vectorizations of persistence diagrams by developing a differentiable layer for neural networks that processes information encoded persistence diagrams.

9.2. Comparing Reeb graphs and their variants

We discuss properties of functional distortion distance d_{FD}, edit distance d_E, interleaving distance d_I, as well as other measures for graph-based topological descriptors.

Functional distortion distance. d_{FD} between a pair of Reeb graphs is an *extended pseudometric* [BGW14]. It is stable against perturbations of the input function [BGW14]. Let $\mathcal{G}_f$ and $\mathcal{G}_g$ denote the Reeb graphs for $(\mathbb{X}, f)$ and $(\mathbb{X}, g)$, respectively.

Theorem 10 ([BGW14], Theorem 4.1) Let $f, g : \mathbb{X} \to \mathbb{R}$ be tame functions whose Reeb quotient maps $\mu_f : \mathbb{X} \to \mathcal{G}_f$ and $\mu_g : \mathbb{X} \to \mathcal{G}_g$ have continuous sections. Then,

$$d_{FD}(\mathcal{G}_f, \mathcal{G}_g) \leq ||f - g||_\infty. \tag{32}$$

The *tameness* is a technical condition requiring the functions to be well behaved. d_{FD} between a pair of Reeb graphs is more discriminative than the bottleneck distance d_∞ between persistence diagrams of the Reeb graphs [BGW14].

Theorem 11 ([BGW14], Theorem 4.2)

$$d_\infty(\mathcal{D}_0(\mathcal{G}_f), \mathcal{D}_0(\mathcal{G}_g)) \leq d_{FD}(\mathcal{G}_f, \mathcal{G}_g). \tag{33}$$

$$d_\infty(\mathcal{D}_0(\mathcal{G}_{-f}), \mathcal{D}_0(\mathcal{G}_{-g})) \leq d_{FD}(\mathcal{G}_f, \mathcal{G}_g). \tag{34}$$

Here, $\mathcal{G}_{-f}$ and $\mathcal{G}_{-g}$ correspond to the Reeb graphs obtained by sweeping the range in the reverse direction.

The main cost of calculating d_{FD} is to calculate the Gromov-Hausdorff (GH) distance d_{GH} of input spaces. Schmiedl [Sch17] showed that d_{GH} cannot be approximated within any reasonable bound in polynomial time.

Edit distance. d_E was proven to be a metric by Sridharamurthy *et al.* [SMKN20]. The stability of d_E is unknown. We conjecture that d_E is not L^∞-stable, that is, small changes in function values may cause pairing switches in the merge tree, resulting in a large increase in distance. d_E is more discriminative than the bottleneck distance d_∞ and the Wasserstein distance d_p as conjectured by Sridharamurthy *et al.*. The computation of d_E takes $O(n^2)$ time for trees with bounded degree [SMKN20].

Edit distance between labeled Reeb graphs. d_{EG} is shown to be an extended pseudometric [BFL16, BLM20]. It is also proven to be stable.

Theorem 12 ([BFL16], Corollary 4.2.) Let $\mathbb{M}$ be a connected, closed, orientable, smooth manifold of dimension 1 or 2. For every simple Morse functions $f, g : \mathbb{M} \to \mathbb{R}$, we have

$$d_{EG}((\mathcal{G}_f, l_f), (\mathcal{G}_g, l_g)) \leq ||f - g||_\infty. \tag{35}$$

We conjecture that computing d_{EG} is NP-hard since it is at least graph-isomorphism hard [BLM20].

Interleaving distance between merge trees. d_I is proven to be a metric [MBW13, Lemma 1]. d_I is stable *w.r.t.* the largest difference between the two scalar functions [MBW13].

Theorem 13 ([MBW13], Theorem 2) Given two merge trees $\mathcal{T}_f, \mathcal{T}_g$ defined by two scalar functions $f, g : \mathbb{X} \to \mathbb{R}$, then

$$d_I(\mathcal{T}_f, \mathcal{T}_g) \leq ||f - g||_\infty. \tag{36}$$

d_I is more discriminative than distances between persistence diagrams, such as the bottleneck distance d_∞.

Theorem 14 ([MBW13], Theorem 3) Given two tame functions $f, g : \mathbb{X} \to \mathbb{R}$, then

$$d_\infty(\mathcal{D}_f, \mathcal{D}_g) \leq d_I(\mathcal{T}_f, \mathcal{T}_g). \tag{37}$$

Here, $\mathcal{D}_f$ and $\mathcal{D}_g$ are the 0-dimensional persistence diagram of f and g, respectively.

Computing d_I on pairs of merge trees or Reeb graphs is NP-hard [AFN*18]. Later Touli and Wang [TW19] gave an FPT (fixed-parameter tractable) algorithm for computing d_I for a pair of merge trees.

Interleaving distance between labeled merge trees. d_{IL} was shown to be a metric by Gasparovic *et al.* [GMO*19]. We conjecture that d_{IL} is not L^∞-stable *w.r.t.* perturbations of the scalar field, since small changes in function values may cause changes in node correspondences between a pair of merge tree. But it has its own notion of stability [GMO*19].

Theorem 15 [GMO*19, Lemma 3.2] Given any pair of valid matrices M_1, M_2, and their associated merge trees $\mathcal{T}_1$ and $\mathcal{T}_2$,

$$d_{IL}(\mathcal{T}_1, \mathcal{T}_2) \leq ||M_1 - M_2||_\infty. \tag{38}$$

We conjecture that d_{IL} is more discriminative that d_∞. Computing d_{IL} between a pair of (leaf) labeled merge trees is polynomial in the number of leaves [MS19]. Considering leaf labeling strategies in [YWM*20], the labeling step for d_{IL} with tree mapping and Euclidean mapping takes $O(n^3)$ due to the use of the Hungarian algorithm. The distance computation for d_{IL} takes $O(n^2)$.

Interleaving distance between Reeb graphs. d_{IG} is an extended pseudometric [DSMP16, Proposition 4.5]. It has L^∞-stability.

Theorem 16 [DSMP16, Theorem 4.6 with simplified notations] Let $(\mathbb{X}, f)$ and $(\mathbb{X}, g)$ be the space that gives rise to the Reeb graphs $\mathcal{G}_f$ and $\mathcal{G}_g$, then

$$d_{IG}(\mathcal{G}_f, \mathcal{G}_g) \leq ||f - g||_\infty, \tag{39}$$

where $\mathcal{G}_f$ represents the Reeb graph using a sheaf-theoretical language.

The calculation of d_{IG} is NP-hard [DSMP16].

Distance based on branch decompositions. For the distance d_{BR} between merge trees based on branch decompositions [BYM*14], whether d_{BR} is a metric is unknown. We conjecture that it is a metric. Its stability was not investigated in the original work [BYM*14]. On the other hand, binary decisions made during the branch decomposition create instabilities *w.r.t.* the resulting branches [SSW14]. The work in [BYM*14] is believed (by Saikia *et al.* [SSW14]) to alleviate such an issue by considering all possible branch decompositions. However, we still conjecture that d_{BR} is likely unstable. Although no theoretical proof is offered, Beketayev *et al.* [BYM*14] used d_∞ between persistence diagrams during their experiments as a baseline for the d_B between merge trees, and observed that d_{BR} is more discriminative than d_∞. The computational complexity of d_{BR} depends on two quantities: first, the time complexity of a function, which for a predefined ε, determines whether two branch decompositions match; and second, the number of iterations of binary search for ε_{min} – the smallest ε for two branch decompositions to be ε-similar. The former takes $O(n^2m^2(n+m))$, where n and m are the numbers of extrema in each merge tree respectively; assuming $n \approx m$, this is roughly $O(n^5)$. Beketayev *et al.* claimed the latter to be "moderate"; which we denote as $O(\log(I_\varepsilon))$, I_ε being the search range.

For comparative measures involving histograms derived from merge trees, both χ^2-distance [SW17] and L^2 distance [SSW15] between histograms give rise to a metric. The running time including the construction of a distance matrix is $O(n^2B)$, B being the number of bins. Saikia *et al.* [SSW15] claimed that the merging of histograms works well under small perturbations in the data. The computation of the distance using L^2 distance of the log-scaled bin values [SSW15] takes $O(n^2B)$. The χ^2-histogram distance [SW17] can also be computed in polynomial time.

The distance between a pair of extended Reeb graphs (ERGs) [BMM*03] is proven to be a metric. A kernel between extended Reeb graphs [BB13] has a worst case running time of $O(\gamma_{max}^4 + 2n^2\log(n))$, where γ_{max} is the length of the maximal shortest path.

Other graph-based or tree-based comparative measures. Some comparative measures are not well investigated regarding their mathematical properties, such as tree alignment distance [LWL*20], and similarity measure between subtrees of contour trees [TN11]. For two contour trees with bounded degree, the former can be calculated in $O(|T_1|\cdot|T_2|)$ (the cost of the alignment), which can be simplified to be $O(n^2)$ assuming the larger tree contains n critical points. For the latter, a polynomial time algorithm is available with a worst case running time $O(t^5\log t)$, where t is the number of branches. The eBDG [SSW14] can be calculated in $O(Nd\log d)$, where N is the number of nodes in eBDG and d is average branching factor. The comparison measure can be calculated in $O(N_1N_2\log_{l_1+1}(N_1+1)\log_{l_2+1}(N_2+1))$, where N_1, N_2 are number of nodes, and l_1, l_2 are average levels in the two trees.

Some comparative measures proposed for visualization have no analysis of mathematical properties or computational complexity, such as similarity measures derived from attributes [HSKK01, ZBB04, SB06, SWC*08, WZ13, SSGC*18, ARC20].

9.3. Comparing Morse and Morse-Smale Complexes

Distance between extremum graphs. d_ρ is proven to be a metric [NTN15]. Narayanan *et al.* mentioned studies on its stability and discriminativity as topics for future work. Computing d_ρ involves weighted maximum clique enumeration, which has exponential time complexity $O(3^{n/3})$ (n being the number of vertices, using the Bron-Kerbosch clique enumeration algorithm); hence, it is feasible for only small graphs [NTN15].

Feature correspondences with Morse-Smale cells. The feature correspondence framework using Morse-Smale complexes [FHJB13] contains no theoretical guarantee. Feng *et al.* gave empirical evidence of stability. However, the stability follows primarily because the Auto Diffusion Function (ADF) that they design is smooth and noise free. Schnorr *et al.* [SHD*20] introduced a feature tracking framework using dissipation elements (DEs), which are by definition, equivalent to 3D Morse-Smale cells [GBG*14]. They determined features correspondences by solving a maximum-weight independent set problem, which is NP-hard in general. However, with certain assumptions, the authors reduced the problem to computing weighted, bipartite graph matching in practice, making the computation tractable.

9.4. Other Comparative Measures

Finally, we discuss known properties associated with other topological descriptors that arise specifically in visualization tasks. As described in Sect. 6, given a Morse function defined on a manifold, the Morse shape descriptor (MSD) [AC07] uses relative homology groups to encode the topology among pairs of sublevel sets of the function. The ranks of relative homology groups (represented as Betti numbers) across multiple levels encode "complete topological information about the critical regions of the manifold as well as the extension of the regions where no topological change is observed" [AC07]. The distance between a pair of Morse shape descriptor (MSD) can be computed in $O(n^2)$ time (n is the number of critical points in a Morse function, which is used for the discretization of level sets).

For topological descriptors used in ensemble visualization, as described in Sect. 8, a L^2 distance is defined between two persistence maps [FFST18]; its computation takes $O(n^2M)$, where M denotes the number of vertices in the domain.

In the study of time-varying and multi-field comparison (see Sect. 7), the L^p distance between two fiber component distributions is proven to be a metric [ACN21]. The mathematical properties associated with the similarity measures for multi-resolution Reeb spaces remain unknown due to partially heuristic matchings between nodes and attributions, in a way similar to the situation in [HSKK01, ZBB04].

Poco *et al.* [PDT*15] introduced a topology-based measure that computes a locality-aware correspondence between similar extrema of two scalar fields, to help ecologists compare species distribution models. The authors found that this similarity measure is stable under the influence of noise in practice. The algorithm has a complexity of $O(n^3)$ because of the maximum weight bipartite graph matching step.

9.5. Open Source Implementations

Efficient and robust open source software for computing topological structures and comparative measures are key to their broad adoption within application domains. A number of tools are available for computing persistence diagrams and/or barcodes, together with their bottleneck and Wasserstein distances, including GUDHI [Gud21], PHAT [BKR*21], Dionysus [Mor21], R-TDA [FKL*21], HERA [KMN21], persim [Per21], Perseus [Nan21], and Ripser [Bau21]. Their utilities extend beyond visualization, into applications in machine learning tasks. Many of these tools are implemented in C/C++ or Python, while a few provide Python/R wrappers. Otter *et al.* [OPT*17] discussed software for persistence homology and also provided installation guides and use cases. Computation tools for merge trees, contour trees and Reeb graphs together with support for computing branch decomposition representations, symmetry detection, and feature tracking are available in software such as Recon [DN21], contour-tree [Dor21], mtlib [Sai21], AMT [Yan21], and SymmetryViewer [TN21]. These software are implemented in C/C++, Python, or Java. Tools for computing Morse-Smale complexes alike include mscomplex3d [SN21], MSCEER [Gyu21], CompExtGraph [Nar21]. Topology ToolKit (TTK) [TFL*21] is a popular toolkit designed to work together with the visualization software ParaView [AGL05], that supports the computation and visualization of persistence diagrams, merge trees, contour trees, Reeb graphs, and Morse-Smale complexes, together with persistence based simplifications of these descriptors. It also allows computation of bottleneck/Wasserstein distances between persistence diagrams and feature tracking via nested tracking graphs.

10. Future Research Opportunities

After analyzing the collection of work discussed in this survey, we found that there are several research gaps and hence opportunities in the study of scalar field comparison with topological descriptors. We now discuss this topic in detail.

Looking at Table 2, we immediately observe that many comparative measures are available with nice mathematical properties. Table 1, on the other hand, shows that topological descriptors and their associated comparative measures have been used for a wide variety of visualization tasks, for which they seem to be especially well suited. On a closer look, we observe that the measures appearing in both tables do not match very well. Many mathematically sound comparative measures have not found practical applications. On the other hand, some comparison-based visualization tasks have been developed using heuristics; the properties of the associated measures are not investigated comprehensively nor supported by theoretical guarantees, *which leads to the question about the reason for this gap between theory and practice.* We attempt to provide some answers below and point out respective research opportunities.

10.1. Computational Efficiency and Stability

The computation of metrics proposed for graph-based topological descriptors is NP-hard (*e.g.,* [Bil05, BB18, AFN*18]), which makes them feasible for only small graphs. Much more work is needed to develop concepts, algorithms, and implementations applicable to large real-world data.

Approximation algorithms. One approach to efficiency is to develop approximation algorithms for computing the comparative measures. This approach can be achieved by reducing the complexity of the input data using concepts proposed in mapper [SMC07] or cosheaf [DSMP16, BBMW21] *w.r.t.* the Reeb graph, which are gaining popularity in computational topology. Cavanna *et al.* [CJS15] studied sparse filtrations that selectively prune data points from the input and proved that such filtrations give good approximations to the barcodes. Touli *et al.* [TW19] suggested an FPT (fixed-parameter tractable) algorithm that approximates the interleaving distance between merge trees. However, for many of the distance measures introduced in Sect. 4, no efficient algorithms are available. Here, we see big opportunities for the development of approximation algorithms by applying controlled relaxation of hard mathematical constraints to achieve better performance.

Heuristic matching strategies. Many approaches tackle this challenge by providing heuristic graph matching strategies. This alternative to the use of classic algorithms for graph isomorphism significantly reduces the time for comparing graph-based topological

structures [ZBB04, WZ13, YWM*20, LWL*20]. Finding appropriate heuristics while maintaining a mathematically sound framework is a challenging task. Important tasks that need to be addressed in the future include a thorough evaluation of mapping strategies by conducting computational experiments and theoretical analysis.

New topological descriptors. Some works introduced new topological descriptors that are easier to compute (*e.g.,* [ZBB04]). However, a detailed investigation of their mathematical properties has not been undertaken. A general need for new topological descriptors remains, in particular, those that encode application-specific information for comparative analysis and visualization. Understanding the mathematical and computational properties associated with any new topological descriptor will help to increase its utilization in applications.

Scalable computation. Scalable computations of comparative measures rely partially on scalable computations of topological descriptors. Some efforts have been made to develop scalable computation of topological descriptors using careful engineering, parallel or distributed computation, for instance, for mapper constructions [HAR20, ZCR*20], merge trees [MW13], contour trees [MW14b], and Morse-Smale complexes [GBP19, SPN20], to name a few. A few recent efforts have adapted the comparative analysis to in situ environments [FAL*16, SPCT18], which is an important research direction in dealing with large real-world data.

Stable measures for gradient-based descriptors. Both Table 2 and Table 1 show that few comparative measures have been developed for gradient-based topological descriptors such as Morse and Morse-Smale complexes and their variants. These descriptors are more sensitive to perturbations in the scalar fields, making it difficult to design effective comparative measures. Furthermore, topological descriptors for multi-fields such as the Jacobi sets, Reeb spaces, and multivariate mappers are relatively less understood. Computing/comparing these descriptors remains a challenge.

10.2. Application-Specific Topological Feature Descriptors

Defining features of interest in an application context is a challenging problem in itself, involving a variety of considerations. While topological descriptors provide a good abstraction, they may not capture all characteristics of the data. A domain-specific interpretation of the comparison measure is a further challenge.

Augmentation of topological descriptors. Some attempts have been made to define topological descriptors that can be further augmented with more geometric- or attribute-related information, such as the comparison of dual contour trees that establishes correspondences based on node attributes [ZBB04]. An extension of this idea could be beneficial in many applications.

Heuristic node mapping strategies. Heuristic strategies for node mapping, as mentioned above, can lead to the possibility of integrating domain knowledge into the comparison process. The heuristic strategies often lead to a violation of the metric axioms and stability properties, and hence require a solid evaluation.

Topological descriptors for multi-fields. The design of comparative measures for multi-fields also has potential for application-specific feature comparison. However, this is also a challenging problem. The introduction of topological descriptors for multi-fields is still in its infancy, and the desirable properties are not well understood. Measures for their comparison also need further exploration.

Topological descriptors for ensembles. Ensemble simulations are ubiquitous and pose a specific challenge to all analysis methods, including topological ones. So far, most methods are based on pairwise comparison of ensemble members (Sect. 8.3), and often visualized as similarity matrices or topological summaries (Sect. 8.2). These methods usually assume a Gaussian distribution of the data, which is restrictive. Few papers go beyond pairwise comparison and support clustering and outlier detection (Sect. 8.3). Even in these cases, new comparative measures are required to complete the analysis. The work in [LPYW21] represents an interesting paradigm shift in the study of ensemble data, where matrix sketching (and, in general, techniques from randomized linear algebra) can be used to obtain ensemble representations and to detect outliers.

10.3. Integration in Visualization Pipelines

We see potential for the integration of topological methods into interactive tools and visualization pipelines for all visualization tasks discussed in this report. This integration will support the exploration and comparison of scalar fields, and analysis of time-varying data and ensembles.

The biggest hurdle to suc cess in applications is the availability of the methods within open-source software such as ParaView [AGL05] and the Topology ToolKit [TFL*18], which allows visualization experts to integrate topological methods even if they are not expert developers of TDA techniques.

11. Conclusion

This state-of-the-art report presents a taxonomy of existing approaches that develop or utilize topological descriptors for the comparative analysis and visualization of scalar fields. In addition, a major contribution of this report is the collection of mathematical and computational properties for the various comparative measures of topological descriptors in the literature, which spans applied topology, computational topology, topological data analysis, and visualization. Some of the comparative measures described in this report have been developed and used in fields outside visualization, such as machine learning, computer vision, and computer graphics. Although the focus of this report is on visualization applications, we have included a few references to these other connections when appropriate.

The development and deployment of visualization techniques and tools based on these comparative measures have impacted various application domains. We list below a set of application areas together with references to the description above that discusses specific visualization tasks.

- **Structural Biology**: Topological analysis of biomolecular structures imaged using various microscopy techniques has benefited

from the development of symmetry detection (highlighting repeating substructures in biomolecules, Sect. 6.1.1), shape matching (protein classification, Sect. 6.2.3), and contour-tree-based tracking methods (bond tracking, Sect. 7.1.2).

- **Climate Science**: Topological descriptors have played a key role in the development of cloud tracking methods (Sect. 7.1), identifying periodicity in surface temperatures (Sect. 7.2), and uncertainty visualization (Sect. 8.5).
- **Combustion Studies**: Pairwise comparison of physical quantities measured during a combustion simulation has resulted in an improved understanding of the different stages of combustion (Sect. 7.2 and Sect. 7.3).
- **Neuroscience**: Space-time structures built on fMRI data are helpful in the study of the dynamic organization of the brain. Comparing neuronal trees helps us understand how the brain functions (Sect. 7.3 and Sect. 6.2.1).
- **Computational Physics and Chemistry**: Comparative measures between multi-fields are helpful in the study of stable Pt-CO bonds and identification of nuclear scission points in simulation data (Sect. 8.4).
- **Ecology**: Comparative measures based on TDA have been successfully used for exploration and better understanding of the species distribution models (Sect. 8.3).

We believe that addressing the various research gaps outlined in Sect. 10 will enable further applications both in the above-mentioned and other areas of science and engineering.

Acknowledgement

This work is partially supported by the United States Department of Energy (DOE) grant DE-SC0021015 and National Science Foundation (NSF) grant IIS-1910733. This work is also partially supported by an Indo-Swedish joint network project: DST/INT/SWD/VR/P-02/2019 and Swedish Research Council (VR) grant 2018-07085, and the VR grant 2019-05487. RS and VN are partially supported by a scholarship from MHRD, Swarnajayanti Fellowship from the Department of Science and Technology, India (DST/SJF/ETA-02/2015-16), and a Mindtree Chair research grant.

References

[AC07] ALLILI M., CORRIVEAU D.: Topological analysis of shapes using morse theory. *Computer Vision and Image Understanding 105*, 3 (2007), 188–199. 12, 14, 23, 27

[ACN21] AGARWAL T., CHATTOPADHYAY A., NATARAJAN V.: Topological feature search in time-varying multifield data. In *Topological Methods in Visualization: Theory, Software and Applications*. Springer-Verlag, 2021. 21, 23, 27

[AEK*17] ADAMS H., EMERSON T., KIRBY M., NEVILLE R., PETERSON C., SHIPMAN P., CHEPUSHTANOVA S., HANSON E., MOTTA F., ZIEGELMEIER L.: Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research 18*, 1 (2017), 218–252. 4, 5, 8

[AFN*18] AGARWAL P. K., FOX K., NATH A., SIDIROPOULOS A., WANG Y.: Computing the Gromov-Hausdorff distance for metric trees. *ACM Transactions on Algorithms (TALG) 14*, 2 (2018), 1–20. 26, 27

[AGL05] AHRENS J., GEVECI B., LAW C.: Paraview: An end-user tool for large-data visualization. *The Visualization Handbook* (2005), 717. 27, 28

[AMY*20] ATHAWALE T., MALJOVEC D., YAN L., JOHNSON C. R., PASCUCCI V., WANG B.: Uncertainty visualization of 2D Morse complex ensembles using statistical summary maps. *IEEE Transactions on Visualization and Computer Graphics* (2020). 2, 12, 22

[ARC20] AGARWAL T., RAMARMURTHI Y., CHATTOPADHYAY A.: A topological similarity measure between multi-field data using multi-resolution Reeb spaces. *arXiv pre-print arXiv:2008.12567* (2020). 7, 12, 18, 22, 26

[Bau21] BAUER U.: Ripser. https://github.com/Ripser/ripser, 2021. 27

[BB13] BARRA V., BIASOTTI S.: 3D shape retrieval using kernels on extended Reeb graphs. *Pattern Recognition 46*, 11 (2013), 2985–2999. 10, 12, 14, 23, 26

[BB14] BARRA V., BIASOTTI S.: 3D shape retrieval and classification using multiple kernel learning on extended Reeb graphs. *The Visual Computer 30*, 11 (2014), 1247–1259. 6

[BB18] BJERKEVIK H., BOTNAN M.: Computational complexity of the interleaving distance. In *34th International Symposium on Computational Geometry (SoCG 2018)* (2018), Toth C., Speckmann B., (Eds.), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 1–15. 27

[BBK21] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Tosca shapes. http://tosca.cs.technion.ac.il/book/resources_data.html, 2021. 14

[BBMW21] BROWN A., BOBROWSKI O., MUNCH E., WANG B.: Probabilistic convergence and stability of random mapper graphs. *Journal of Applied and Computational Topology 5*, 1 (2021), 99–140. 6, 27

[BCBB14] BIASOTTI S., CERRI A., BRONSTEIN A., BRONSTEIN M.: Quantifying 3D shape similarity using maps: Recent trends, applications and perspectives. In *Eurographics 2014 - State of the Art Reports* (2014), Lefebvre S., Spagnuolo M., (Eds.), The Eurographics Association. 13

[BD17] BUBENIK P., DLOTKO P.: A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation 78* (2017), 91–114. 4, 24

[BEMP13] BENDICH P., EDELSBRUNNER H., MOROZOV D., PATEL A.: Homology and robustness of level and interlevel sets. *Homology, Homotopy and Applications 15* (2013), 51–72. 4

[Ber79] BERTSEKAS D. P.: A distributed algorithm for the assignment problem. *Lab. for Information and Decision Systems Working Paper, MIT* (1979). 23

[BFL16] BAUER U., FABIO B. D., LANDI C.: An edit distance for Reeb graphs. In *Eurographics Workshop on 3D Object Retrieval* (2016), Ferreira A., Giachetti A., Giorgi D., (Eds.), The Eurographics Association. 5, 9, 23, 25

[BGW14] BAUER U., GE X., WANG Y.: Measuring distance between Reeb graphs. *Proceedings of the 30th Annual Symposium on Computational Geometry* (2014), 464–474. 9, 23, 25

[BH12] BETHEL E. W., HOWISON M.: Multi-core and many-core shared-memory parallel raycasting volume rendering optimization and tuning. *The International Journal of High Performance Computing Applications 26*, 4 (2012), 399–412. 15

[BHJ*14] BONNEAU G.-P., HEGE H.-C., JOHNSON C. R., OLIVEIRA M. M., POTTER K., RHEINGANS P., SCHULTZ T.: Overview and state-of-the-art of uncertainty visualization. In *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, Chen M., Hagen H., Hansen C., Johnson C. R., Kauffman A., (Eds.). Springer, 2014, pp. 3–27. 22

[Bil05] BILLE P.: A survey on tree edit distance and related problems. *Theoretical computer science 337*, 1-3 (2005), 217–239. 27

[BKR*21] BAUER U., KERBER M., REININGHAUS J., WAGNER H., KELLER B.: Persistent Homology Algorithm Toolbox (PHAT). https://bitbucket.org/phat-code/phat/, 2021. 27

[BKW15] BECK F., KOCH S., WEISKOPF D.: Visual analysis and dissemination of scientific literature collections with Survis. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 180–189. 1, 11

[BLM20] BAUER U., LANDI C., MEMOLI F.: The Reeb graph edit distance is universal. *Foundations of Computational Mathematics* (2020). 9, 22, 25

[BMM*03] BIASOTTI S., MARINI S., MORTARA M., PATANE G., SPAGNUOLO M., FALCIDIENO B.: 3D shape matching through topological structures. *International conference on discrete geometry for computer imagery* (2003), 194–203. 10, 12, 14, 23, 26

[BMW15] BAUER U., MUNCH E., WANG Y.: Strong equivalence of the interleaving and functional distortion metrics for Reeb graphs. *Proceedings of 31st International Symposium on Computational Geometry 34* (2015), 461–475. 9

[Bub15] BUBENIK P.: Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research 16* (2015), 77–102. 4, 8, 9, 23, 24

[Bub20] BUBENIK P.: The persistence landscape and some of its properties. In *Topological Data Analysis. Abel Symposia*, Baas N. A., Carlsson G. E., Quick G., Szymik M., Thaule M., (Eds.), vol. 15. Springer, Cham, 2020, pp. 97–117. 4

[BWN*15] BHATIA H., WANG B., NORGARD G., PASCUCCI V., BREMER P.-T.: Local, smooth, and consistent Jacobi set simplification. *Computational Geometry: Theory and Applications 48*, 4 (2015), 311–332. 7

[BWP*10] BREMER P.-T., WEBER G. H., PASCUCCI V., DAY M. S., BELL J. B.: Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics 16*, 2 (2010), 248–260. 12, 15, 19

[BYM*14] BEKETAYEV K., YELIUSSIZOV D., MOROZOV D., WEBER G., HAMANN B.: Measuring the distance between merge trees. *Topological Methods in Data Analysis and Visualization III: Theory, Algorithms, and Applications, Mathematics and Visualization* (2014), 151–166. 1, 9, 10, 12, 15, 23, 26

[CB20] CARRIÉRE M., BLUMBERG A.: Multiparameter persistence image for topological machine learning. *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)* (2020). 7

[CCF*20] CATANZARO M. J., CURRY J. M., FASY B. T., LAZOVSKIS J., MALEN G., RIESS H., WANG B., ZABKA M.: Moduli spaces of Morse functions for persistence. *Journal of Applied and Computational Topology 4* (2020), 353–385. 5

[CCI*20] CARRIERE M., CHAZAL F., IKE Y., LACOMBE T., ROYER M., UMEDA Y.: PersLay: A neural network layer for persistence diagrams and new graph topological signatures. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, PMLR 108* (2020), 2786–2796. 25

[CCO17] CARRIÈRE M., CUTURI M., OUDOT S.: Sliced Wasserstein kernel for persistence diagrams. *Proceedings of the 34th International Conference on Machine Learning 70* (2017), 664–673. 8, 23, 24

[CCSG*09] CHAZAL F., COHEN-STEINER D., GLISSE M., GUIBAS L. J., OUDOT S. Y.: Proximity of persistence modules and their diagrams. In *Proceedings of the 25th Annual Symposium on Computational Geometry* (New York, NY, USA, 2009), ACM, pp. 237–246. 9

[CD13] CARR H., DUKE D.: Joint Contour Nets. *IEEE Transactions on Visualization and Computer Graphics 20*, 8 (2013), 1100–1113. 7, 21

[CdS10] CARLSSON G., DE SILVA V.: Zigzag persistence. *Foundations of Computational Mathematics 10*, 4 (2010), 367–405. 4

[CDSGO16] CHAZAL F., DE SILVA V., GLISSE M., OUDOT S.: *The structure and stability of persistence modules*. Springer, Cham, 2016. 4, 22

[CHS15] CHAZAL F., HUANG R., SUN J.: Gromov-Hausdorff approximation of filamentary structures using Reeb-type graphs. *Discrete & Computational Geometry 53*, 3 (2015), 621–649. 6

[CJS15] CAVANNA N. J., JAHANSEIR M., SHEEHY D. R.: A geometric perspective on sparse filtrations. In *Proceedings of the Canadian Conference on Computational Geometry* (Kingston, ON, Canada, 2015). 27

[CL20] CHUNG Y.-M., LAWSON A.: Persistence curves: A canonical framework forsummarizing persistence diagrams. arXiv preprint arXiv:1904.07768, 2020. 5

[CLB11] CORREA C., LINDSTROM P., BREMER P.-T.: Topological spines: A structure-preserving visual representation of scalar fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1842–1851. 7

[CMR*13] CARDONA G., MIR A., ROSSELLÓ F., ROTGER L., SÁNCHEZ D.: Cophenetic metrics for phylogenetic trees, after Sokal and Rohlf. *BMC Bioinformatics 14*, 1 (2013), 3. 9, 10

[CN20] CHOWDHURY S., NEEDHAM T.: Gromov-Wasserstein averaging in a riemannian framework. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2020), 842–843. 19

[CO17] CARRIÈRE M., OUDOT S.: Local equivalence and intrinsic metrics between Reeb graphs. *Proceedings of the 33rd International Symposium on Computational Geometry 77* (2017), 1–15. 22

[CO18] CARRIÉRE M., OUDOT S.: Structure and stability of the one-dimensional mapper. *Foundations of Computational Mathematics 18*, 6 (2018), 1333–1396. 7

[CSA03] CARR H., SNOEYINK J., AXEN U.: Computing contour trees in all dimensions. *Computational Geometry Theory and Applications 24*, 2 (2003), 75–94. 1, 5

[CSEH07] COHEN-STEINER D., EDELSBRUNNER H., HARER J.: Stability of persistence diagrams. *Discrete & Computational Geometry 37* (2007), 103–120. 8, 22, 23

[CSEHM10] COHEN-STEINER D., EDELSBRUNNER H., HARER J., MILEYKO Y.: Lipschitz functions have l_p-stable persistence. *Foundations of Computational Mathematics 10*, 2 (2010), 127–139. 8, 23, 24

[Cur18] CURRY J.: The fiber of the persistence map for functions on the interval. *Journal of Applied and Computational Topology 2*, 3-4 (2018), 301–321. 5

[CZ09] CARLSSON G., ZOMORODIAN A.: The theory of multidimensional persistence. *Discrete & Computational Geometry 42*, 1 (2009), 71–93. 7

[CZCG04] CARLSSON G., ZOMORODIAN A. J., COLLINS A., GUIBAS L. J.: Persistence barcodes for shapes. *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2004), 124–135. 1

[DMW16] DEY T. K., MÉMOLI F., WANG Y.: Mutiscale mapper: A framework for topological summarization of data and maps. *Proceedings of the 27th annual ACM-SIAM symposium on Discrete algorithms* (2016), 997–1013. 7

[DN21] DORAISWAMY H., NATARAJAN V.: Recon. https://bitbucket.org/vgl_iisc/reconjava, 2021. 27

[DNN13] DORAISWAMY H., NATARAJAN V., NANJUNDIAH R. S.: An exploration framework to identify and track movement of cloud systems. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2896–2905. 12, 15, 17

[Dor21] DORAISWAMY H.: Topological Data Analysis using Contour Trees (contour-tree). https://github.com/harishd10/contour-tree, 2021. 27

[DSMP16] DE SILVA V., MUNCH E., PATEL A.: Categorified Reeb graphs. *Discrete & Computational Geometry 55*, 4 (2016), 854–906. 9, 10, 23, 26, 27

[DSW15] DEY T., SHI D., WANG Y.: Comparing graphs via persistence distortion. *Proceedings of the 31st International Symposium on Computational Geometry* (2015), 491–506. 10, 12, 15

[ECdVGG12] ÉRIC COLIN DE VERDIÉRE, GINOT G., GOAOC X.: Multinerves and helly numbers of acyclic families. *Proceedings of the 28th annual symposium on Computational geometry* (2012), 209–218. 7

[EH04] EDELSBRUNNER H., HARER J.: Jacobi sets of multiple Morse functions. In *Foundations of Computational Mathematics, Minneapolis 2002*, Cucker F., DeVore R., Olver P., Sueli E., (Eds.). Cambridge Univ. Press, England, 2004, pp. 37–57. 7, 12, 19

[EH08] EDELSBRUNNER H., HARER J.: Persistent homology - a survey. *Contemporary Mathematics 453* (2008), 257–282. 8

[EH10] EDELSBRUNNER H., HARER J.: *Computational Topology: An Introduction*. American Mathematical Society, 2010. 3, 4, 5, 6, 7, 23

[EHMP04] EDELSBRUNNER H., HARER J., MASCARENHAS A., PASCUCCI V.: Time-varying Reeb graphs for continuous space-time data. *Proceedings of the annual symposium on Computational geometry* (2004), 366–372. 12, 16, 17

[EHNP03] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Morse-Smale complexes for piecewise linear 3-manifolds. *Proceedings of the 19th Annual symposium on Computational Geometry* (2003), 361–370. 1, 7

[EHNP04] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Local and global comparison of continuous functions. In *Proceedings of IEEE Conference on Visualization* (Austin, TX, USA, 2004), pp. 275–280. 7, 12, 18

[EHP08] EDELSBRUNNER H., HARER J., PATEL A. K.: Reeb spaces of piecewise linear mappings. In *Proceedings of the 24th Annual Symposium on Computational Geometry* (College Park, MD, USA, 2008), pp. 242–250. 7, 21

[EHZ01] EDELSBRUNNER H., HARER J., ZOMORODIAN A.: Hierarchical Morse complexes for piecewise linear 2-manifolds. In *Proceedings of the 17th Annual Symposium on Computational Geometry* (Medford, MA, USA, 2001), pp. 70–79. 1, 6, 7

[ELZ02] EDELSBRUNNER H., LETSCHER D., ZOMORODIAN A.: Topological persistence and simplification. *Discrete & Computational Geometry 28* (2002). 1

[EM90] EDELSBRUNNER H., MÜCKE E. P.: Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics 9* (1990), 66–104. 3

[EMB*20] ENGELKE W., MASOOD T. B., BERAN J., CABALLERO R., HOTZ I.: Topology-based feature design and tracking for multi-center cyclones. *arXiv preprint arXiv:2011.08676* (2020). 12, 15, 16

[FAL*16] FRIESEN B., ALMGREN A., LUKIC Z., WEBER G., MOROZOV D., BECKNER V., DAY M.: In situ and in-transit analysis of cosmological simulations. *Computational Astrophysics and Cosmology 3* (2016), 1–18. 28

[FFST18] FAVELIER G., FARAJ N., SUMMA B., TIERNY J.: Persistence atlas for critical point variability in ensembles. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2018), 1152–1162. 12, 19, 20, 23, 27

[FHJB13] FENG W., HUANG J., JU T., BAO H.: Feature correspondences using Morse Smale complex. *The Visual Computer 29*, 1 (2013), 53–67. 7, 10, 12, 13, 14, 23, 26

[FKL*21] FASY B. T., KIM J., LECCI F., MARIA C., MILLMAN D. L., ROUVREAU V.: R-TDA. https://rdrr.io/cran/TDA/, 2021. 27

[GBG*14] GYULASSY A., BREMER P.-T., GROUT R., KOLLA H., CHEN J., PASCUCCI V.: Stability of dissipation elements: A case study in combustion. *Computer Graphics Forum 33*, 3 (2014), 51–60. 26

[GBP19] GYULASSY A., BREMER P.-T., PASCUCCI V.: Shared-memory parallel computation of Morse-Smale complexes with improved accuracy. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 1183–1192. 28

[GBPW10] GERBER S., BREMER P.-T., PASCUCCI V., WHITAKER R.: Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics 16* (2010), 1271–1280. 7

[Ghr08] GHRIST R.: Barcodes: The persistent topology of data. *Bullentin of the American Mathematical Society 45* (2008), 61–75. 1

[GMK04] GAMEIRO M., MISCHAIKOW K., KALIES W.: Topological characterization of spatial-temporal chaos. *Physical Review E 70*, 035203 (2004). 5

[GMO*19] GASPAROVIC E., MUNCH E., OUDOT S., TURNER K., WANG B., WANG Y.: Intrinsic interleaving distance for merge trees. *arXiv eprint ArXiv:1908.00063* (2019). 9, 10, 22, 23, 26

[GP12] GERBER S., POTTER K.: Data analysis with the Morse-Smale complex: The msr package for R. *Journal of Statistical Software 50*, 2 (2012). 1, 7

[GRSW14] GÜNTHER D., REININGHAUS J., SEIDEL H.-P., WEINKAUF T.: Notes on the simplification of the Morse-Smale complex. In *Topological Methods in Data Analysis and Visualization III*, Bremer P.-T., Hotz I., Pascucci V., Peikert R., (Eds.). Springer International Publishing, 2014, pp. 135–150. 7

[GST14] GÜNTHER D., SALMON J., TIERNY J.: Mandatory critical points of 2D uncertain scalar fields. *Computer Graphics Forum 33*, 3 (2014), 31–40. 12, 20, 22

[Gud21] GUDHI DEVELOPERS: Gudhi 3.4.1. http://gudhi.gforge.inria.fr/, 2021. 27

[Gyu21] GYULASSY, ATTILA: MSCEER - Morse Smale Complex Extraction, Exploration, Reasoning. https://github.com/sci-visus/MSCEER, 2021. 27

[HAR20] HAJIJ M., ASSIRI B., ROSEN P.: Parallel mapper. *Proceedings of the Future Technologies Conference (FTC) 2* (2020), 717–731. 28

[HG15] HUETTENBERGER L., GARTH C.: A Comparison of Pareto Sets and Jacobi Sets. In *Topological and Statistical Methods for Complex Data*, Bennett J., Vivodtzev F., Pascucci V., (Eds.), Mathematics and Visualization. Springer, 2015, pp. 125–141. 7

[HHC*13] HUETTENBERGER L., HEINE C., CARR H., SCHEUERMANN G., GARTH C.: Towards multifield scalar topology based on pareto optimality. *Computer Graphics Forum 32*, 3 (2013), 341–350. 12, 21

[HK73] HOPCROFT J. E., KARP R. M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing 2*, 4 (1973), 225–231. 23

[HKNU17] HOFER C., KWITT R., NIETHAMMER M., UHL A.: Deep learning with topological signatures. *Advances in Neural Information Processing Systems* (2017), 1634–1644. 25

[HLH*16] HEINE C., LEITTE H., HLAWITSCHKA M., IURICICH F., FLORIANI L. D., SCHEUERMANN G., HAGEN H., GARTH C.: A survey of topology-based methods in visualization. *Computer Graphics Forum 35*, 3 (2016), 643–667. 1, 2

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), 203–212. 10, 12, 14, 18, 19, 22, 26, 27

[HWSR18] HAJIJ M., WANG B., SCHEIDEGGER C., ROSEN P.: Visual detection of structural changes in time-varying graphs using persistent homology. *IEEE Pacific Visualization Symposium (PacificVis)* (2018). 18

[HZLR20] HAJIJ M., ZHANG Y., LIU H., ROSEN P.: Persistent homology and the discrete Laplace operator for mesh similarity. *Computer Graphics and Visual Computing* (2020). 8, 12, 13

[KEF*17] KUHN A., ENGELKE W., FLATKEN M., HEGE H.-C., HOTZ I.: Topology-based analysis for multimodal atmospheric data of volcano eruptions. *In Topological Methods in Data Analysis and Visualization V: Theory, Algorithms, and Applications (Proceedings of TopoInVis 2017)* (2017). 12, 15, 16, 17

[KFH17] KUSANO G., FUKUMIZU K., HIRAOKA Y.: Kernel method for persistence diagrams via kernel embedding and weight factor. *Journal of Machine Learning Research 18*, 1 (2017), 6947–6987. 8, 23, 24

[KGH20] KANARI L., GARIN A., HESS K.: From trees to barcodes and back again: theoretical and statistical perspectives. *Algorithms 13*, 12 (2020), 335. 5

[KHNH12] KASTEN J., HOTZ I., NOACK B., HEGE H.-C.: Vortex merge graphs in two-dimensional unsteady flow fields. *Proceedings of EG/VGTC Conference on Visualization Short Papers* (2012). 12, 16

[KMN17] KERBER M., MOROZOV D., NIGMETOV A.: Geometry helps to compare persistence diagrams. *Journal of Experimental Algorithmics 22*, 1.4 (2017). 23, 24

[KMN21] KERBER M., MOROZOV D., NIGMETOV A.: HERA. https://bitbucket.org/grey_narn/hera/, 2021. 27

[KRHH11] KASTEN J., REININGHAUS J., HOTZ I., HEGE H.-C.: Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2080–2087. 12, 15, 16

[KTSK00] KLEIN P., TIRTHAPURA S., SHARVIT D., KIMIA B.: A tree-edit-distance algorithm for comparing simple, closed shapes. In *Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms* (San Francisco, CA, USA, 2000), pp. 696–704. 9

[KVT19] KONTAK M., VIDAL J., TIERNY J.: Statistical parameter selection for clustering persistence diagrams. *Proceedings of Super Computing workshop on Urgent HPC* (2019). 12, 20

[KZHH12] KASTEN J., ZOUFAHL A., HEGE H.-C., HOTZ I.: Analysis of vortex merge graphs. In *Proceedings of 17th International Workshop on Vision, Modeling and Visualization (VMV)* (Magdeburg, Germany, 2012). 12, 15, 16

[LAS*17] LUKASCZYK J., ALDRICH G., STEPTOE M., FAVELIER G., GUEUNET C., TIERNY J., MACIEJEWSKI R., HAMANN B., LEITTE H.: Viscous fingering: A topological visual analytic approach. *Applied Mechanics and Materials 869* (2017), 9–19. 17

[Les12] LESNICK M.: *Multidimensional Interleavings and Applications to Topological Inference*. PhD thesis, Stanford University, 2012. 7

[LPYW21] LI M., PALANDE S., YAN L., WANG B.: Sketching merge trees for scientific data visualization. *arXiv preprint arXiv:2101.03196* (2021). 12, 18, 28

[LS16] LIEBMANN T., SCHEUERMANN G.: Critical points of Gaussian-distributed scalar fields on simplicial grids. *Computer Graphics Forum 35*, 3 (2016). 21

[LWA*17] LI Y., WANG D., ASCOLI G. A., MITRA P., WANG Y.: Metrics for comparing neuronal tree shapes based on persistent homology. *PLOS ONE 12*, 8 (2017), e0182184. 5, 9, 12, 13, 14, 23, 25

[LWL*20] LOHFINK A.-P., WETZELS F., LUKASCZYK J., WEBER G. H., GARTH C.: Fuzzy contour trees: Alignment and joint layout of multiple contour trees. *Computer Graphics Forum 39*, 3 (2020), 343–355. 9, 12, 20, 23, 26, 28

[LWM*17] LUKASCZYK J., WEBER G., MACIEJEWSKI R., GARTH C., LEITTE H.: Nested tracking graphs. *Computer Graphics Forum 36*, 3 (2017), 12–22. 2, 12, 16, 17

[LY18] LE T., YAMADA M.: Persistence Fisher kernel: A Riemannian manifold kernel for persistence diagrams. *Advances in Neural Information Processing Systems 31* (2018), 10028–10039. 8, 23, 24

[Mat02] MATSUMOTO Y.: *An introduction to Morse theory*, vol. 208. American Mathematical Society, 2002. 3

[MBW13] MOROZOV D., BEKETAYEV K., WEBER G.: Interleaving distance between merge trees. *Proceedings of Topology-Based Methods in Visualization (TopoInVis)* (2013). 9, 23, 25

[MHRB20] MOOR M., HORN M., RIECK B., BORGWARDT K.: Topological autoencoders. *Proceedings of the 37th International Conference on Machine Learning, PMLR 119* (2020), 7045–7054. 25

[Mil63] MILNOR J.: *Morse Theory*. Princeton University Press, New Jersey, NY, USA, 1963. 3

[Mor21] MOROZOV D.: Dionysus. https://mrzv.org/software/dionysus2/, 2021. 27

[MS19] MUNCH E., STEFANOU A.: The ℓ^∞-cophenetic metric for phylogenetic trees as an interleaving distance. In *Research in Data Science*, Association for Women in Mathematics Series. Springer International Publishing, 2019, pp. 109–127. 9, 10, 26

[MTN13] MASOOD T. B., THOMAS D. M., NATARAJAN V.: Scalar field visualization via extraction of symmetric structures. *The Visual Computer 29*, 6-8 (2013), 761–771. 11

[Mun57] MUNKRES J.: Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics 5*, 1 (1957), 32–38. 16, 23

[MW13] MOROZOV D., WEBER G.: Distributed merge trees. *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (2013), 93–102. 28

[MW14a] MIHAI M., WESTERMANN R.: Visualizing the stability of critical points in uncertain scalar fields. *Computers & Graphics 41* (2014), 13–25. 12, 22

[MW14b] MOROZOV D., WEBER G.: Distributed contour trees. *Topological Methods in Data Analysis and Visualization III: Theory, Algorithms, and Applications, Mathematics and Visualization* (2014), 89–102. 28

[MW16] MUNCH E., WANG B.: Convergence between categorical representations of Reeb space and mapper. In *32th International Symposium on Computational Geometry (SoCG 2016)* (Boston, MA, USA, 2016), vol. 51, pp. 53:1–53:16. 7

[Nan21] NANDA V.: Perseus, the persistent homology software. http://www.sas.upenn.edu/~vnanda/perseus, 2021. 27

[Nar21] NARAYANAN V.: Complete Extremum Graph. https://bitbucket.org/vgl_iisc/compextgraph/, 2021. 27

[NEFH20] NILSSON E., ENGELKE W., FRIEDERICI A., HOTZ I.: Tracking and visualizing multi-center cyclones. In *Levia* (2020). 12, 16

[NN11] NAGARAJ S., NATARAJAN V.: Simplification of Jacobi sets. In *Topological Data Analysis and Visualization: Theory, Algorithms and Applications* (2011), Pascucci V., Tricoche X., Hagen H., Tierny J., (Eds.), Springer-Verlag, pp. 91–102. 7

[NTN15] NARAYANAN V., THOMAS D. M., NATARAJAN V.: Distance between extremum graphs. *IEEE Pacific Visualization Symposium* (2015), 263–270. 2, 11, 12, 18, 23, 26

[OPT*17] OTTER N., PORTER M. A., TILLMANN U., GRINDROD P., HARRINGTON H. A.: A roadmap for the computation of persistent homology. *EPJ Data Science 6* (2017), 1–38. 27

[Oud17] OUDOT S. Y.: *Persistence Theory: From Quiver Representations to Data Analysis (Mathematical Surveys and Monographs)*. American Mathematical Society, 2017. 22

[PCMS04] PASCUCCI V., COLE-MCLAUGHLIN K., SCORZELLI G.: Multi-resolution computation and presentation of contour trees. *Proceedings of the IASTED Conference on Visualization, Imaging, and Image Processing* (2004), 452–290. 6

[PDT*15] POCO J., DORAISWAMY H., TALBERT M., MORISETTE J., SILVA C. T.: Using maximum topology matching to explore differences in species distribution models. *IEEE Scientific Visualization Conference* (2015), 9–16. 12, 21, 27

[Per21] PERSIM DEVELOPERS: Distances and representations of persistence diagrams (Persim 0.3.0). https://github.com/scikit-tda/persim, 2021. 27

[Pos03] POST F. H.: The state of the art in flow visualization: Feature extraction and tracking. *Computer Graphics Forum 22*, 4 (2003), 775–792. 15

[Ree46] REEB G.: Sur les points singuliers d'une forme de pfaff completement intergrable ou d'une fonction numerique (on the singular points of a complete integral pfaff form or of a numerical function). *Comptes Rendus Acad. Science Paris 222* (1946), 847–849. 1

[RFLL17] RIECK B., FUGACCI U., LUKASCZYK J., LEITTE H.: Clique community persistence: A topological visual analysis approach for complex networks. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 822–831. 15

[RHBK15] REININGHAUS J., HUBER S., BAUER U., KWITT R.: A stable multi-scale kernel for topological machine learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), 4741–4748. 8, 23, 24

[RKG*11] REININGHAUS J., KOTAVA N., GÜNTHER D., KASTEN J., HAGEN H., HOTZ I.: A scale space based persistence measure for critical points in 2D scalar fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2045–2052s. 12, 16

[RKWH12] REININGHAUS J., KASTEN J., WEINKAUF T., HOTZ I.: Efficient computation of combinatorial feature flow fields. *IEEE Transactions on Visualization and Computer Graphics 18*, 9 (2012), 1563–1573. 12, 16

[RL16a] RIECK B., LEITTE H.: Exploring and comparing clusterings of multivariate data sets using persistent homology. *Computer Graphics Forum 35*, 3 (2016), 81–90. 12, 15

[RL16b] RIECK B., LEITTE H.: 'Shall I compare thee to a network?': Visualizing the topological structure of Shakespeare's plays. In *Proceedings of the 1st Workshop on Visualization for the Digital Humanities* (Baltimore, MD, USA, 2016). 15

[Rob02] ROBINS V.: Computational topology for point data: Betti numbers of α-shapes. In *Morphology of Condensed Matter. Lecture Notes in Physics*, Mecke K., Stoyan D., (Eds.), vol. 600. Springer, Berlin, Heidelberg, 2002, pp. 261–274. 5

[RR92] RAMESH R., RAMAKRISHNAN I.: Nonlinear pattern matching in trees. *Journal of the ACM 39*, 2 (1992), 295–316. 9

[RR07] RAHIMI A., RECHT B.: Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007* (2007), Platt J. C., Koller D., Singer Y., Roweis S. T., (Eds.), Curran Associates, Inc., pp. 1177–1184. 24

[RSL17] RIECK B., SADLO F., LEITTE H.: Hierarchies and ranks for persistence pairs. In *Topological Methods in Data Analysis and Visualization V (TopoInVis 2017)*. Springer, 2017, pp. 3–17. 5, 9, 12, 18

[RSL20a] RIECK B., SADLO F., LEITTE H.: Persistence concepts for 2D skeleton evolution analysis. In *Topological Methods in Data Analysis and Visualization V: Theory, Algorithms, and Applications*. Springer, 2020. 12, 15, 17

[RSL20b] RIECK B., SADLO F., LEITTE H.: Topological machine learning with persistence indicator functions. In *Topological Methods in Data Analysis and Visualization V: Theory, Algorithms, and Applications*, Carr H., Sadlo F., Fujishiro I., Takahashi S., (Eds.). Springer International Publishing, 2020, pp. 87–101. 5, 9

[RY98] RISTAD E. S., YIANILOS P. N.: Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 5 (1998), 522–532. 9

[Sai21] SAIKIA, HIMANGSHU: A merge tree library (mtlib). https://github.com/hsaikia/mtlib, 2021. 27

[SB06] SOHN B.-S., BAJAJ C.: Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics 12*, 1 (2006), 14–25. 12, 15, 16, 26

[SBS02] SHAMIR A., BAJAJ C., SOHN B.-S.: *Progressive Tracking of Isosurfaces in Time-Varying Scalar Fields*. Tech. rep., CS & TICAM Technical Report TR-02-4, University of Taxes Austin, 2002. 12, 16

[Sch17] SCHMIEDL F.: Computational aspects of the Gromov-Hausdorff distance and its application in non-rigid shape matching. *Discrete & Computational Geometry 57*, 4 (2017), 854–880. 25

[SHC*19] SCHNORR A., HELMRICH D. N., CHILDS H., KUHLEN T. W., HENTSCHEL B.: Feature tracking utilizing a maximum-weight independent set problem. In *IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV)* (Vancouver, BC, Canada, 2019), pp. 6–15. 12, 17

[SHD*20] SCHNORR A., HELMRICH D. N., DENKER D., KUHLEN T. W., HENTSCHEL B.: Feature tracking by two-step optimization. *IEEE Transactions on Visualization and Computer Graphics 26*, 6 (2020). 12, 16, 17, 23, 26

[Sik01] SIKORA T.: The MPEG-7 visual standard for content description-an overview. *IEEE Transactions on circuits and systems for video technology 11*, 6 (2001), 696–702. 14

[SMC07] SINGH G., MÉMOLI F., CARLSSON G.: Topological methods for the analysis of high dimensional data sets and 3D object recognition. In *Eurographics Symposium on Point-Based Graphics* (2007), pp. 91–100. 21, 27

[SMKN20] SRIDHARAMURTHY R., MASOOD T. B., KAMAKSHIDASAN A., NATARAJAN V.: Edit distance between merge trees. *IEEE Transactions on Visualization and Computer Graphics 26*, 3 (2020), 1518–1531. 2, 9, 12, 14, 18, 23, 25

[SN21] SHIVASHANKAR N., NATARAJAN V.: mscomplex3d. https://bitbucket.org/vgl_iisc/mscomplex-3d/, 2021. 27

[SNN11] SUTHAMBHARA N., NATARAJAN V., NANJUNDIAH R. S.: A gradient-based comparison measure for visual analysis of multifield data. *Computer Graphics Forum 30*, 3 (2011), 1101–1110. 12, 15, 18

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG) 23*, 3 (2004), 399–405. 13

[SP21] SUMNER R. W., POPOVIĆ J.: Mesh data from deformation transfer for triangle meshes. https://people.csail.mit.edu/sumner/research/deftransfer/data.html, 2021. [Online; accessed 18-January-2021]. 13

[SPCT18] SOLER M., PLAINCHAULT M., CONCHE B., TIERNY J.: Lifted Wasserstein matcher for fast and robust topology tracking. In *IEEE 8th Symposium on Large Data Analysis and Visualization (LDAV)* (Berlin, Germany, 2018), pp. 23–33. 2, 12, 16, 23, 24, 28

[SPD*19] SOLER M., PETITFRERE M., DARCHE G., PLAINCHAULT M., CONCHE B., TIERNY J.: Ranking viscous finger simulations to an acquired ground truth with topology-aware matchings. In *IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV)* (Vancouver, BC, Canada, 2019), pp. 62–72. 12, 16, 17, 18, 24

[SPN20] SUBHASH V., PANDEY K., NATARAJAN V.: GPU parallel computation of morse-smale complexes. In *IEEE Visualization Conference* (2020), pp. 36–40. 28

[SSGC*18] SAGGAR M., SPORNS O., GONZALEZ-CASTILLO J., BANDETTINI P. A., CARLSSON G., GLOVER G., REISS A. L.: Towards a new approach to reveal dynamical organization of the brain using topological data analysis. *Nature Communications 9*, 1399 (2018). 10, 12, 15, 19, 26

[SSW14] SAIKIA H., SEIDEL H. P., WEINKAUF T.: Extended branch decomposition graphs: Structural comparison of scalar data. *Computer Graphics Forum 33*, 3 (2014), 41–50. 2, 6, 9, 10, 11, 12, 18, 26

[SSW15] SAIKIA H., SEIDEL H.-P., WEINKAUF T.: Fast similarity search in scalar fields using merging histograms. *Topological Methods in Data Analysis and Visualization IV (Proceedings of Topology-Based Methods in Visualization)* (2015), 121–134. 10, 12, 23, 26

[SW17] SAIKIA H., WEINKAUF T.: Global feature tracking and similarity estimation in time-dependent scalar fields. *Computer Graphics Forum 36*, 3 (2017), 1–11. 2, 10, 12, 17, 26

[SWC*08] SCHNEIDER D., WIEBEL A., CARR H., HLAWITSCHKA M., SCHEUERMANN G.: Interactive comparison of scalar fields based on largest contours with applications to flow visualization. *IEEE Transaction on Visualization and Computer Graphics 14*, 6 (2008), 1475–1482. 12, 13, 26

[TFL*18] TIERNY J., FAVELIER G., LEVINE J. A., GUEUNET C., MICHAUX M.: The Topology ToolKit. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018). 28

[TFL*21] TIERNY J., FAVELIER G., LEVINE J. A., GUEUNET C., MICHAUX M.: The Topology ToolKit. https://topology-tool-kit.github.io/, 2021. 27

[The20] THE RIVET DEVELOPERS: RIVET 1.1.0. https://github.com/rivetTDA/rivet/, 2020. 7

[Tie17] TIERNY J.: *Topological Data Analysis for Scientific Visualization*. Springer, 2017. 3

[TITP19] TANTARDINI M., IEVA F., TAJOLI L., PICCARDI C.: Comparing methods for comparing networks. *Scientific Reports 9*, 17557 (2019). 10

[TMB14] TURNER K., MUKHERJEE S., BOYER D. M.: Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA 3*, 4 (2014), 310–344. 5, 9, 12, 14, 23, 25

[TMMH14] TURNER K., MILEYKO Y., MUKHERJEE S., HARER J.: Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry 52*, 1 (2014), 44–70. 8

[TN11] THOMAS D. M., NATARAJAN V.: Symmetry in scalar field topology. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2035–2044. 2, 10, 12, 13, 23, 26

[TN13] THOMAS D. M., NATARAJAN V.: Detecting symmetry in scalar fields using augmented extremum graphs. *IEEE Transactions of Visualization and Computer Graphics 19*, 12 (2013), 2663–2672. 2, 7, 11, 12, 13

[TN14] THOMAS D. M., NATARAJAN V.: Multiscale symmetry detection in scalar fields by clustering contours. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2427–2436. 11, 12, 13

[TN21] THOMAS D., NATARAJAN V.: SymmetryViewer. https://vgl.csa.iisc.ac.in/symmetryViewer/, 2021. 27

[TW19] TOULI E. F., WANG Y.: FPT-algorithms for computing Gromov-Hausdorff and interleaving distances between trees. *Proceedings of the 27th Annual European Symposium on Algorithms* (2019), 83:1–83:14. 9, 26, 27

[VBT20] VIDAL J., BUDIN J., TIERNY J.: Progressive Wasserstein barycenters of persistence diagrams. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 151–161. 12, 20

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research 9* (2008). 19

[Vil03] VILLANI C.: *Topics in Optimal Transportation*, vol. 58. American Mathematical Society, 2003. 7

[Vip20] VIPOND O.: Multiparameter persistence landscapes. *Journal of Machine Learning Research 21* (2020), 1–38. 7

[VMN*19] VALSANGKAR A. A., MONTEIRO J. M., NARAYANAN V., HOTZ I., NATARAJAN V.: An exploratory framework for cyclone identification and tracking. *IEEE Transaction on Visualization and Computer Graphics 25*, 3 (2019), 1460–1473. 12, 16

[WBD*11] WEBER G., BREMER P.-T., DAY M. S., BELL J. B., PASCUCCI V.: Feature tracking using Reeb graphs. In *Topological Methods in Data Analysis and Visualization. Mathematics and Visualization.*, Pascucci V., Tricoche X., Hagen H., Tierny J., (Eds.). Springer, Berlin, Heidelberg, Germany, 2011. 12, 19

[WM20] WILLS P., MEYER F. G.: Metrics for graph comparison: A practitioner's guide. *PLOS ONE 15*, 2 (2020), e0228728. 10

[WTGP10] WEINKAUF T., THEISEL H., GELDER A. V., PANG A.: Stable feature flow fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 6 (2010), 770–780. 12, 16

[WZ13] WU K., ZHANG S.: A contour tree based visualization for exploring data with uncertainty. *International Journal for Uncertainty Quantification 3*, 3 (2013). 10, 12, 22, 26, 28

[Yan21] YAN, LIN: Average Merge Tree (AMT). https://github.com/tdavislab/amt, 2021. 27

[YWM*20] YAN L., WANG Y., MUNCH E., GASPAROVIC E., WANG B.: A structural average of labeled merge trees for uncertainty visualization. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 832–842. 2, 9, 12, 20, 21, 22, 26, 28

[ZBB04] ZHANG X., BAJAJ C. L., BAKER N.: *Fast matching of volumetric functions using multi-resolution dual contour trees*. Tech. rep., Texas Institute for Computational and Applied Mathematics, Austin, Texas, 2004. 6, 12, 14, 18, 20, 22, 26, 27, 28

[ZCR*20] ZHOU Y., CHALAPATHI N., RATHORE A., ZHAO Y., WANG B.: Mapper Interactive: A scalable, extendable, and interactive toolbox for the visual exploration of high-dimensional data. *IEEE Pacific Visualization Symposium* (2020). 28

[Zha96] ZHANG K.: A constrained edit distance between unordered labeled trees. *Algorithmica 15* (1996), 205–222. 9

[Zom05] ZOMORODIAN A.: *Topology for Computing*, vol. 16 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2005. 3

[ZS89] ZHANG K., SHASHA D.: Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing 18*, 6 (1989), 1245–1262. 9

[ZSS92] ZHANG K., STATMAN R., SHASHA D.: On the editing distance between unordered labeled trees. *Information processing letters 42*, 3 (1992), 133–139. 9

[ZW19] ZHAO Q., WANG Y.: Learning metrics for persistence-based summaries and applications for graph classification. In *Advances in Neural Information Processing Systems* (2019), pp. 9859–9870. 8, 9, 12, 14, 23, 25

Short Biographies

Lin Yan is a PhD student in the Scientific Computing and Imaging (SCI) Institute, University of Utah. Her research interests are topological data analysis and visualization. Her recent work includes statistical analysis of merge trees, uncertainty visualization of Morse–Smale complexes, and topological signatures for vector fields. Email: lin.yan@utah.edu.

Talha Bin Masood is a Postdoctoral Fellow at Linköping University in Sweden. He received his Ph.D. in Computer Science from Indian Institute of Science, Bangalore. His research interests include scientific visualization, computational geometry, computational topology, and their applications to various scientific domains. Email: talha.bin.masood@liu.se.

Raghavendra Sridharamurthy is a PhD student in the Department of Computer Science and Automation at Indian Institute of Science, Bangalore. His interests include scientific visualization, computational topology and its applications. Email: raghavendrag@iisc.ac.in.

Farhan Rasheed is PhD student at Linköping University in Sweden. He graduated from Heidelberg University with a degree in Scientific Computing. His research interests includes scientific visualization, topological data analysis, machine learning, and medical image computing. Email: farhan.rasheed@liu.se.

Vijay Natarajan is the Mindtree Chair Professor in the Department of Computer Science and Automation at Indian Institute of Science, Bangalore. He received his Ph.D. in Computer Science from Duke University. His research interests include scientific visualization, computational topology, and geometry processing. In current work, he is developing topological methods for time-varying and multi-field data visualization, and studying applications in biology, material science, and climate science. Email: vijayn@iisc.ac.in.

Ingrid Hotz is currently a Professor in Scientific Visualization at the Linköping University in Sweden. She received her Ph.D. degree from the Computer Science Department at the University of Kaiserslautern, Germany. She worked as a postdoctoral researcher at IDAV at the University of California, Davis. Previous positions include leading a junior research group in visualization at the Zuse Institute Berlin and leading the scientific visualization group at the German Aerospace Center (DLR). Her research interests lie in data analysis and scientific visualization, ranging from basic research questions to effective solutions to visualization problems in applications. This includes developing and applying concepts originating from different areas of computer sciences and mathematics, such as computer graphics, computer vision, dynamical systems, computational geometry, and combinatorial topology. Email: ingrid.hotz@liu.se.

Bei Wang is an Assistant Professor at the School of Computing and a faculty member at the Scientific Computing and Imaging (SCI) Institute, University of Utah. She received her Ph.D. in Computer Science from Duke University. She is interested in the analysis and visualization of large and complex data. Her research interests include topological data analysis, data visualization, computational topology, computational geometry, machine learning, and data mining. Her recent work includes statistical analysis and uncertainty visualization of topological descriptors. Email: beiwang@sci.utah.edu.