



HAL
open science

Visual Comparison of Topological Features

Adhitya Kamakshidasan, Vijay Natarajan

► **To cite this version:**

Adhitya Kamakshidasan, Vijay Natarajan. Visual Comparison of Topological Features. [Research Report] University of Glasgow, Glasgow, Scotland, UK; Indian Institute of Science, Bangalore, India. 2020. hal-02569089v1

HAL Id: hal-02569089

<https://inria.hal.science/hal-02569089v1>

Submitted on 11 May 2020 (v1), last revised 11 Aug 2020 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Visual Comparison of Topological Features

Adhitya Kamakshidasan*
University of Glasgow

Vijay Natarajan†
Indian Institute of Science, Bangalore

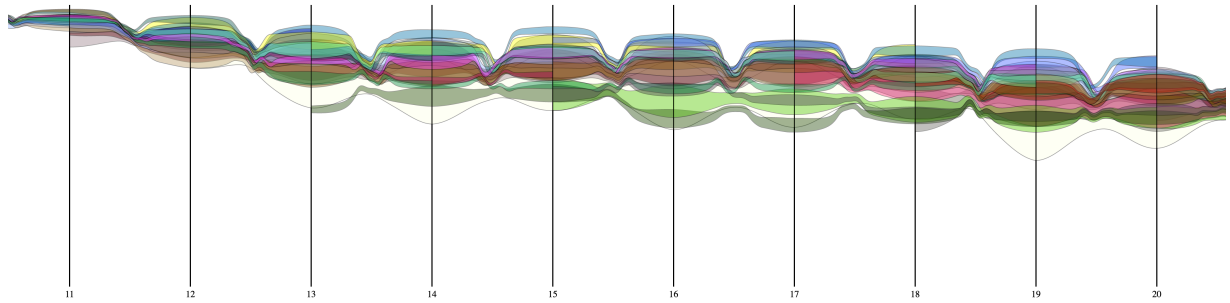


Figure 1: *persistenceBundles* for one ensemble member of the viscous finger dataset. In contrast to current methods that track the nesting hierarchy of a few predetermined levels, our approach intuitively captures the evolution of all persistent features.

ABSTRACT

Understanding the temporal evolution of topological features by means of tracking graphs is a common task, often performed using scalar field analysis. However, these graphs are either hard to interpret or capture information at a very high level that is not intuitive to the user. To bridge this gap, we propose *persistenceBundles*, a hierarchical edge-bundling approach for visualizing tracked features using persistence hierarchies. We demonstrate the effectiveness of our approach using the viscous finger dataset.

Keywords: persistence hierarchy, edge bundling, tracking graphs

1 INTRODUCTION

Topological data analysis have proven to be useful in multiple time-varying contexts ranging from exploring urban relationships to tracking cyclones. Usually in such analysis, topological features derived from one temporal snapshot are compared with another, thereby helping to understand the structure, relationships, and prominent features of the dataset. While sophisticated methods exist for such comparison, an intuitive visual representation is often challenging, owing to the large number of features. To overcome this, existing methods compare a user-specified collection of levels [2, 5–7, 9] on a global scale, are statically rendered and support less interaction capabilities. However, we are interested in supporting the comparison of the entire hierarchy of features and local comparisons, along with interactive queries. Therefore, we provide a design called *persistenceBundles*, that focuses on visual scalability and supports both coarse-grained events of interest, such as aggregated similarities and differences, as well as capabilities to filter fine-grained events, based on persistence or other geometric features. Our design builds upon a topological abstraction called the persistence hierarchy.

2 PERSISTENCE HIERARCHY

Consider a scalar function $f: D \rightarrow \mathbb{R}$ defined on a manifold domain D . A value c in the range of f is called an *isovalue*. Given an isovalue, an *isocontour* or *level set* is defined as the collection of all points $x \in D$ such that $f(x) = c$. A merge tree captures the

*e-mail: adhitya.kamakshidasan@glasgow.ac.uk

†e-mail: vijayn@iisc.ac.in

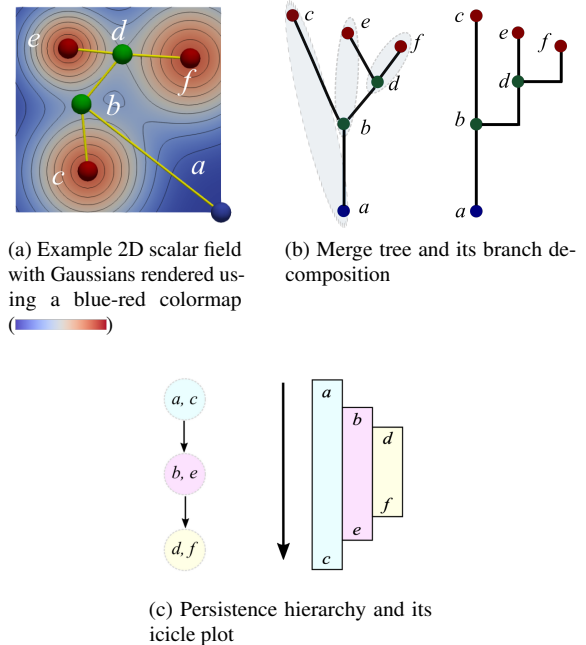


Figure 2: Constructing the persistence hierarchy from a scalar field

connectivity of sub-level sets $f^{-1}(-\infty, c]$ (*join tree*) or super-level sets $f^{-1}[c, \infty)$ (*split tree*) of f . The split tree consists of maxima $M = \{M_i\}$, split saddles $S = \{s_j\}$, and the global minimum. The join tree consists of minima $m = \{m_i\}$, join saddles $S = \{s_k\}$, and the global maximum. For the sake of convenience, we use only the split tree for explanation. Here, all maxima can be paired with saddles based on the notion of topological persistence, except for the global maximum which is paired with the global minimum. Each such pair represents a topological feature and its *persistence* can be defined as the absolute difference between the two scalar function values. The persistence pair nodes are referred to as creator/birth (upper) and destroyer/death (lower). A merge tree can be decomposed into a set of branches, such that each branch contains a persistence pair. This generates a nested hierarchy of branches, wherein each parent branch

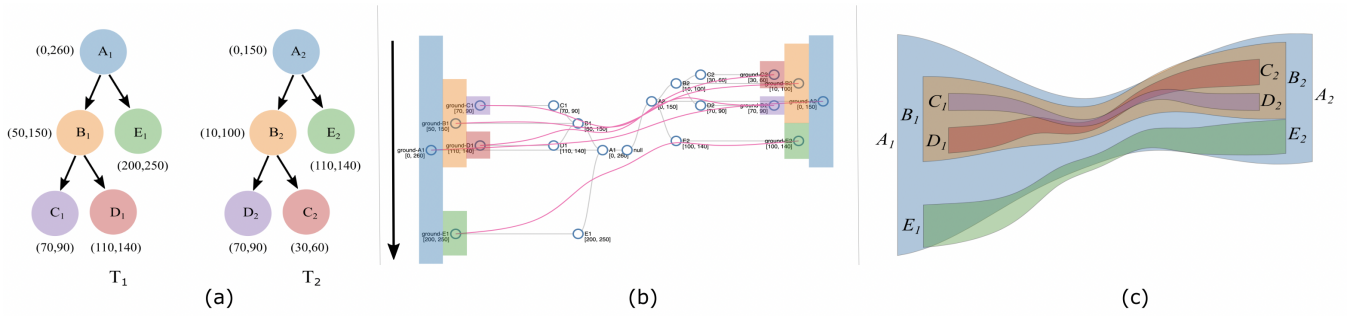


Figure 3: Using persistenceBundles for a simple comparison. (a) Two persistence hierarchies with the same structure and but with different valued (birth-death) pairs. (b) Icicle plots, overlaid with the control-tree, and hierarchically bundled edges. Nodes and edges of the control-tree are shown in blue and grey respectively, while bundled edges between the matched icicles are in pink. (c) Modified 2D hierarchical edge bundling.

has a persistence greater than that of its children. This hierarchy of branches is called the *branch decomposition* representations of the merge tree. While the branch decomposition is often displayed as a collection of L-shaped branches, it can also be represented as a rooted tree, whose nodes represents individual branches. Such a representation is called a persistence hierarchy [4].

3 METHOD

The algorithm for constructing persistenceBundles follows from [8], a tool for comparing hierarchical stack traces. We adapt the algorithm for time-varying persistence hierarchies.

Our goal is to visually compare the persistence hierarchies of two consecutive timesteps of a time-varying dataset, and then extend it to look at all timesteps. Our intuition for the design stems from the fact that each child node of the persistence hierarchy has both its birth and death values within that of its parent. This property makes the persistence hierarchy a good candidate for an icicle plot visualization, wherein each persistence pair denotes an icicle, rendered vertically with its upper-left coordinate equivalent to the pair’s birth value and height equivalent to the pair’s persistence (see Figure 3). The two icicle plots are then placed facing each other for comparison. A one-to-one correspondence between the icicles is by a method that computes the Wasserstein or Bottleneck distance between the persistence diagrams.

Next, we use edge bundles to connect matched pairs in the two hierarchies. We start by the hierarchical edge bundling (HEB) design of [3], and the match-edges are routed along a control-tree structure computed using the hierarchy represented by the icicle plot nodes, mirrored along the y-axis. However, HEB represents edges as 1D curves and renders all edges identically, making it difficult to spot more important edges. Therefore, we draw the edges as curved 2D tubes instead of 1D curves.

Consider two persistent pairs p and q that are connected by a match r . Let x_p, y_p, h_p be the x and y coordinates of the center and height of the icicle plot rectangle for p , and x_q, y_q, h_q be the similar quantities for q . Let γ be the HEB curve that connects the centers. Let $t : [0,1]$ be an arc-length parameterization for γ . We construct two curves γ^{top} and γ^{bottom} which represent the top and bottom curved borders of our tube shape. If $\gamma^{top} = (\gamma_x^{top}(t), \gamma_y^{top}(t))$ and $\gamma^{bottom} = (\gamma_x^{bottom}(t), \gamma_y^{bottom}(t))$,

$$\begin{aligned} \gamma_x^{top}(t) &= \gamma_x^{bottom}(t) = \gamma_x(t) \\ \gamma_y^{bottom/top}(t) &= \gamma_y(t) \pm \phi(t) \left((1-t) \frac{h_p}{2} + t \frac{h_q}{2} \right) \end{aligned}$$

where $\phi(t) : [0,1] \rightarrow [0,1]$ is a function that models the gradual shrinking or thinning of the tube from its ends towards its center. Profiles that generate bundle-like tubes are given by

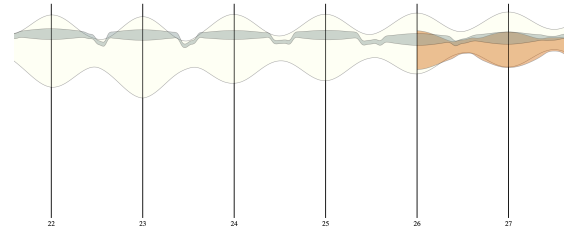


Figure 4: Tracks of persistent features thresholded based on length.

$$\phi(t) = \lambda + \frac{1-\lambda}{2} (1 + \cos 2\pi t)$$

where λ is the bundling strength factor. This above method generates a visual metaphor for comparing two hierarchies, that we call as *persistenceBundles*. We can then look at the entire evolution and tracks, by generating bundles across consecutive timesteps, and then cascading them horizontally. We believe that this bundling approach can significantly reduce clutter and allow for visual scalability.

4 EXAMPLES

We discuss how persistenceBundles can be applied for comparing persistence hierarchies, using an illustrative example and the viscous finger dataset. Consider our example persistence hierarchies T_1 and T_2 with the birth-death pairs and correspondences as shown in Figure 3. By constructing our visual metaphor for these hierarchies, we can clearly see the changes in persistence as well as the nesting structure of the persistent pairs.

The viscous finger dataset [1] describes instability phenomenon that occurs at the interface between two mixing fluids of different viscosities. Scientists are typically interested in tracking how the ‘fingers’ evolve across time. After subsampling the initial dataset to a 64^3 grid, we track features using a Wasserstein matching between the timesteps. Due to the large number of levels, we render all icicle plot nodes with the same x-coordinate, while maintaining the hierarchy. Figure 1 provides a coarse-grained view to look at the gradual increase of all stable viscous fingers. It can be seen that the hierarchical bundling provides an overall aggregated understanding without much clutter. Since each of the tracks are individually rendered, we can interactively filter and provide fine-grained views to look at individual viscous finger tracks based on geometrical features like persistence or hypervolume or track-length (see Figure 4).

ACKNOWLEDGMENTS

This work is partially supported by the Department of Science and Technology, India (DST/SJF/ETA-02/2015-16), a Mindtree Chair research grant, and the Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bangalore. Adhitya sincerely thanks the gurudwaras of Vancouver and Glasgow for providing free meals.

REFERENCES

- [1] IEEE VIS Scientific Visualization Contest, 2016. <http://www.uni-kl.de/sciviscontest/>.
- [2] P. Bremer, G. H. Weber, J. Tierny, V. Pascucci, M. S. Day, and J. B. Bell. A topological framework for the interactive exploration of large scale turbulent combustion. In *IEEE International Conference on e-Science*, 2009.
- [3] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE TVCG*, 2006.
- [4] A. Kamakshidasan and V. Natarajan. Mergemaps: Treemaps for scientific data. *TopoInVis*, 2019.
- [5] W. Köpp and T. Weinkauff. Temporal treemaps: Static visualization of evolving trees. *IEEE TVCG*, 2019.
- [6] J. Lukaszcyk, C. Garth, G. H. Weber, T. Biedert, R. Maciejewski, and H. Leitte. Dynamic nested tracking graphs. *IEEE TVCG*, 2020.
- [7] J. Lukaszcyk, G. Weber, R. Maciejewski, C. Garth, and H. Leitte. Nested tracking graphs. *Computer Graphics Forum*, 2017.
- [8] J. Trümper, J. Döllner, and A. Telea. Multiscale visual comparison of execution traces. In *ICPC*, 2013.
- [9] W. Widanagamaachchi, A. Jacques, Bei Wang, E. Crosman, P. Bremer, V. Pascucci, and J. Horel. Exploring the evolution of pressure-perturbations to understand atmospheric phenomena. In *PacificVis*, 2017.