An Overview of the Topology ToolKit

Talha Bin Masood, Joseph Budin, Martin Falk, Guillaume Favelier, Christoph Garth, Charles Gueunet, Pierre Guillou, Lutz Hofmann, Petar Hristov, Adhitya Kamakshidasan, et al.

An Overview of the Topology ToolKit

Talha Bin Masood, Joseph Budin, Martin Falk, Guillaume Favelier, Christoph Garth, Charles Gueunet, Pierre Guillou, Lutz Hofmann, Petar Hristov, Adhitya Kamakshidasan, Christopher Kappe, Pavol Klacansky, Patrick Laurin, Joshua A. Levine, Jonas Lukasczyk, Daisuke Sakurai, Maxime Soler, Peter Steneteg, Julien Tierny, Will Usher, Jules Vidal, Michal Wozniak

Abstract This software paper gives an overview of the features supported by the Topology ToolKit (TTK), which is an open-source library for Topological data analysis (TDA). TTK implements, in a generic and efficient way, a substantial collection of reference algorithms in TDA. Since its initial public release in 2017, both its user and developer bases have grown, resulting in a significant increase in the number of supported features. In contrast to the original paper introducing TTK [36] (which detailed the core algorithms and data structures of TTK), the purpose of this software paper is to describe the list of features currently supported by TTK, ranging from image segmentation tools to advanced topological analysis of high-dimensional data, with concrete usage examples available on the TTK website [38].

Talha Bin Masood is with Linköping University. e-mail: talha.bin.masood@liu.se · Joseph Budin is with Sorbonne Université. e-mail: joseph.budin@sorbonne-universite.fr · Martin Falk is with Linköping University. e-mail: martin.falk@liu.se · Guillaume Favelier is with INRIA. e-mail: guillaume.favelier@inria.fr · Christoph Garth is with TU Kaiserslautern. e-mail: garth@cs.uni-kl.de · Charles Gueunet is with Kitware. e-mail: charles.gueunet@kitware.com · Pierre Guillou is with Sorbonne Université. e-mail: pierre.guillou@sorbonne-universite.fr · Lutz Hofmann is with Heidelberg University. e-mail: lutz.hofmann@iwr.uni-heidelberg.de · Petar Hristov is with the University of Leeds. e-mail: p.hristov@leeds.ac.uk · Adhitya Kamakshidasan is with INRIA. e-mail: adhitya.kamakshidasan@inria.fr Christopher Kappe is with TU Kaiserslautern. e-mail: kappe@cs.uni-kl.de · Pavol Klacansky is with SCI Institute, University of Utah. e-mail: klacansky@sci.utah.edu · Patrick Laurin is with ShapeShift3D. e-mail: patrick.laurin@shapeshift3d.com · Joshua A. Levine is with University of Arizona. e-mail: josh@email.arizona.edu · Jonas Lukasczyk is with Arizona State University. e-mail: jl@jluk.de · Daisuke Sakurai is with Kyushu University. e-mail: d.sakurai@computer.org · Maxime Soler is with Total / Sorbonne Université. e-mail: soler.maxime@total.com · Peter Steneteg is with Linköping University. e-mail: peter.steneteg@liu.se · Julien Tierny is with CNRS / Sorbonne Université. e-mail: julien.tierny@sorbonne-universite.fr · Will Usher is with SCI Institute, University of Utah. e-mail: will@sci.utah.edu · Jules Vidal is with Sorbonne Université. e-mail: jules.vidal@sorbonne-universite.fr · Michal Wozniak is with ShapeShift3D. e-mail: michal.wozniak@shapeshift3d.com

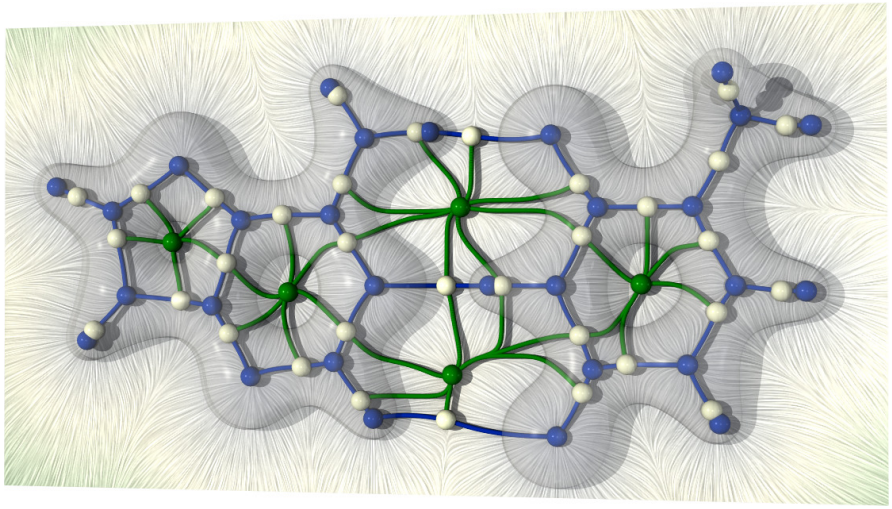

Fig. 1 Extraction of the covalent and non-covalent interactions in a molecular system with TTK. Covalent and hydrogen bonds are captured by the blue separatrices of the Morse-Smale complex, and steric effects (repulsive forces induced by the carbon cycles) are captured by saddle-saddle connectors (green).

1 Introduction

Topological data analysis (TDA) [9, 30, 34] is a vibrant field of study at the cross roads between mathematics and computer science, which considers the *structure* of complex data. In particular thanks to advanced concepts such as persistent homology [9], TDA provides theories and algorithms for the multi-scale representation and analysis of the structural features of interest present in the data. It has been shown to be useful in a variety of fields, ranging from machine learning [7] to geometry processing [45]. In scientific applications, TDA is particularly effective for the analysis of large-scale data sets [14]. The *Topology ToolKit* (TTK) [36] is an open-source library for TDA that has been released in 2017 under the permissive BSD license. It features a substantial collection of generic and efficient implementations of reference TDA algorithms. TTK is mostly written in C++ (~110k lines of code) to offer the best possible performance. To date, 12 institutions have contributed code to TTK, including 9 academic organizations (CNRS, Heidelberg University, INRIA, Linkoping University, Sorbonne Universite, TU Kaiserslautern, University of Arizona, University of Utah, Zuse Institute Berlin) and 3 companies (Kitware, Total, ShapeShift3D). Since its initial release, TTK's website has collected more than 135k page-views, from more than 17k unique visitors, and its video tutorials have collected more than 10k Youtube views. TTK is accessible to developers through several APIs: C++, VTK/C++ or Python. For end users, TTK is directly accessible in the form of a plugin for ParaView [1] and an anaconda package [41]. Data can be provided to TTK in multiple forms: it can be sampled along 1D, 2D, or 3D regular

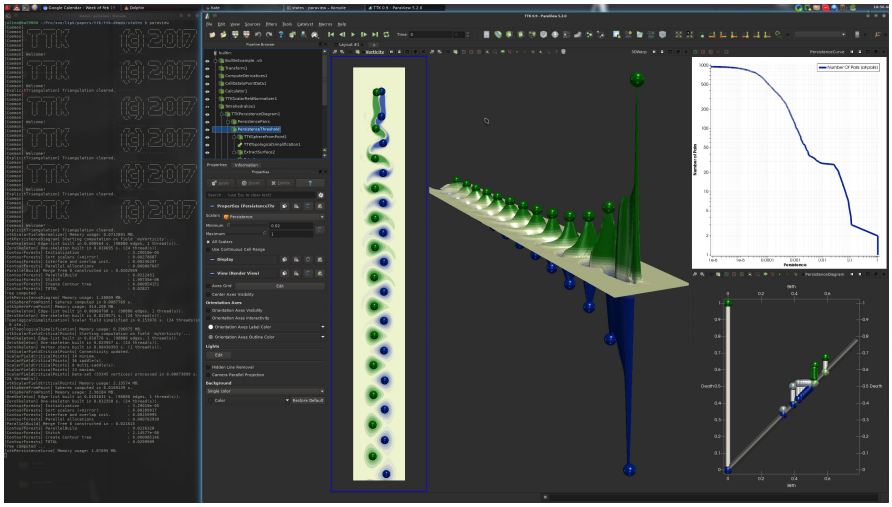

Fig. 2 Persistence-driven analysis pipeline applied to vortex tracking in computational fluid dynamics. Once TTK and its data package are installed, from the `ttk-data/` directory, run the following command to reproduce this example: "`paraview states/BuiltInExample1.pvsm`" (see [38] for further details).

grids (periodic grids are now also supported), or 1D, 2D, or 3D meshes (simplicial complexes). It can also be provided as point clouds of arbitrary dimension.

The internal data structures and algorithms of TTK have already been presented in its companion paper [36], its end-user features have not been formally presented, other than in oral tutorials [13, 11]. This software paper fills this gap by describing the high-level features of TTK through a list of concrete examples. Note that although the following examples will be discussed based on a usage of TTK with ParaView, the entire discussion holds for all TTK's APIs (C++, VTK/C++, Python) as each TTK *filter* in the presented ParaView pipelines (green box in the *pipeline browser*, top left of each screenshot) represents an individual TTK object. We also note that ParaView state files can be automatically exported to Python scripts. All the material necessary to reproduce the examples presented in this paper (data and ParaView state files) is available on the TTK website (section *Tutorials* [38]). Readers are invited to run these examples (see the caption of each figure) to further inspect interactively the content of each illustration.

2 Scalar data

TTK suppports the computation of a large number of topological abstractions for scalar data. For the application, each topological abstraction supported by TTK serves a specific purpose. Critical points [5] can be used to extract points of interest. Merge/contour trees and Reeb graphs [16, 17, 18, 19] can be used to estimate

skeletons and to segment data along level sets. Persistence diagrams [9] can be used to visually represent the population of points of interest (critical points) as well as their salience (topological persistence). The Morse-Smale complex can be used to extract filament structures in data.

Typically, as illustrated in Figure 2, to explore the data at multiple scales, the persistence diagram [9] is first computed to identify the main topological features present in the data and to discard the irrelevant features that correspond to noise. In particular, long vertical bars in the diagram (Figure 2, bottom right) denote topologically salient features, while small bars near the diagonal correspond to spurious features. The persistence curve (Figure 2, top right) provides a visual tool for inspecting relevant persistence thresholds, under which features should be interpreted as noise. In particular, these curves often exhibit in practice flat plateaus which either separate features from noise (at persistence 0.07 in Figure 2), or which separate group of features of different scales. Once a proper persistence threshold has been identified by the user, to reflect the corresponding noise removal in the original data, TTK's support for *topological simplification* [37] can be used. Then, any topological object mentioned above (critical point, merge/contour tree, Reeb graph, Morse-Smale complex) computed after this data simplification step will subsequently be simplified, allowing multi-scale feature exploration. In Figure 2, selecting the persistence threshold corresponding to the flat plateau located at the center of the persitence curve (persistence threshold of 0.07) enables the simplification of all spurious features, and robustly extracts the centers of the vortices of this fluid mechanic example (persistent extrema of the flow orthogonal curl component).

Figures 3, 4, and 5 show further typical usage examples illustrating classical topological data analysis pipelines, where data is pre-simplified by preserving only the most persistent features (highlighted in the corresponding persistence diagrams).

In Figure 3, the simplification is combined with the Morse complex (bottom) to extract cells in confocal microscopy (top left: input data). In this example, the segmentation is illustrated by representing the manifold of each local maximum with a distinct color. In particular, each maximum denotes the nucleus of a cell, its manifold the geometry of the cell, and the network of filament structures extracted from the separatrices of the Morse complex indicate the boundaries separating the cells. In this application, the removal of the small bars from the persistence diagram (top right) removes spurious maxima from the data and consequently resolves the possible over-segmentation provided by the Morse complex alone.

In Figure 4, data pre-simplification is combined with the merge tree to extract bones in medical imaging. In particular, in this example, the user segmented the regions corresponding to each arc of the split tree containing a local maximum (which corresponds to locally dense regions). Here the level of persistence has been tuned to maintain only the five most persistent features (corresponding to the toes of the foot). Maintaining more features (in this example, for a persistence threshold of 150) would precisely segment the bones along each joint, which further illustrates the potential for multi-scale data exploration.

Note that TTK also offers functionality to design harmonic scalar fields by solving the Laplace equations subject to Dirichlet constraints [46] provided by the user at key

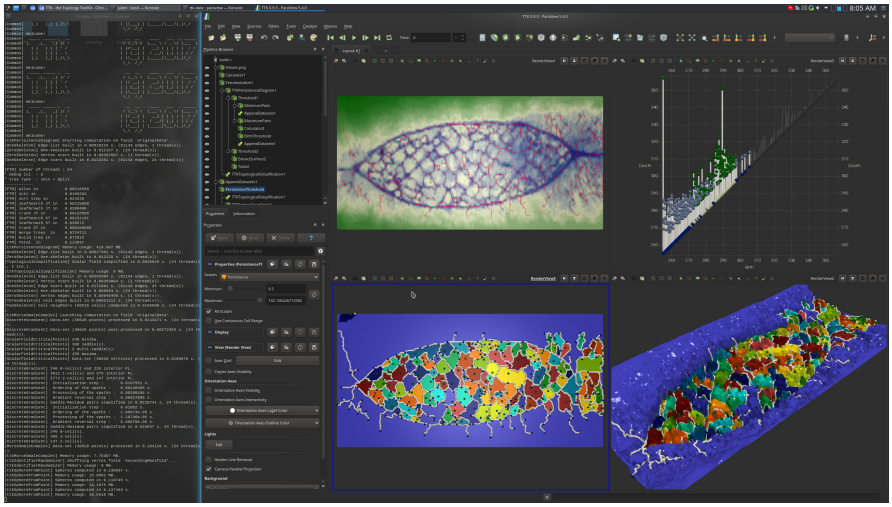

Fig. 3 Persistence-driven analysis pipeline, combined with Morse complex computation for cell enumeration in confocal microscopy (example reproduced from [9], page 217). Once TTK and its data package are installed, from the `ttk-data/` directory, run the following command to reproduce this example: "`paraview states/tribute.pvsm`" (see [38] for further details).

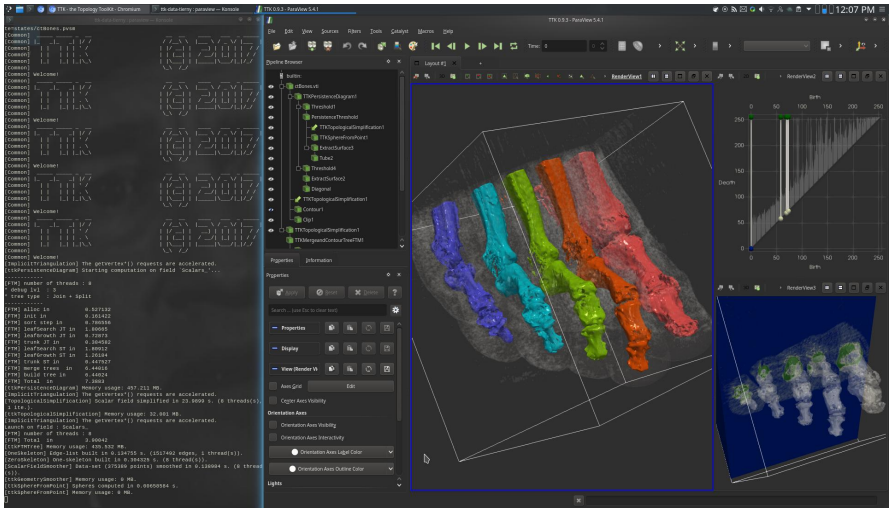

Fig. 4 Persistence-driven merge-tree based segmentation applied to bone extraction in medical imaging. Once TTK and its data package are installed, from the `ttk-data/` directory, run the following command to reproduce this example: "`paraview states/ctBones.pvsm`" (see [38] for further details).

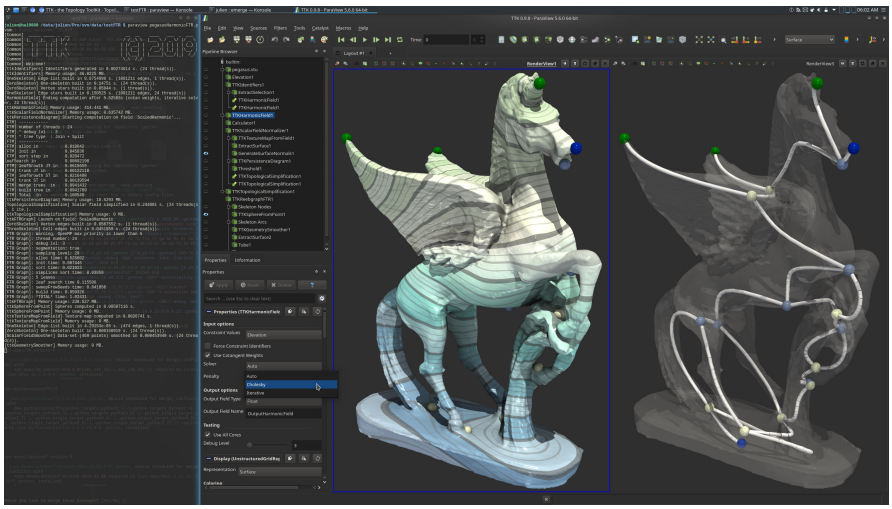

Fig. 5 Skeleton estimation from the Reeb graph [19] of a user designed harmonic field [46]. Once TTK and its data package are installed, from the `ttk-data/` directory, run the following command to reproduce this example: "`paraview states/harmonicSkeleton.pvsm`" (see [38] for further details).

locations (typically at extremities of prominent shape features). This is illustrated in Figure 5 (left), which also illustrates skeleton extraction with the Reeb graph (right).

TTK also implements efficient algorithms [29, 22, 32] for the estimation of distances between persistence diagrams (such as the bottleneck and Wasserstein distances [9]). Recently, efficient and progressive algorithms [44, 24] have been integrated for the computation of barycenters of persistence diagrams [42, 25], which visually summarize the topological features of an ensemble data set, and which enable an efficient clustering of the ensemble members based on their persistence diagram.

3 Bivariate scalar data

TTK supports the computation of several topological abstractions for bivariate data (where the data is characterized by two values defined at each vertex of the geometrical domain). TTK provides a fast implementation of continuous scatterplots [4], which can be interpreted as continuous histograms of bivariate data defined on volumes. They are particularly useful for understanding where and how volumetric data projects to the data range. Fiber surfaces [6, 23] extend the notion of isosurfaces to bivariate data and enable users to explore the regions in the volume corresponding to features of interest segmented manually in the continuous scatterplot. The Jacobi sets [8] are also implemented in TTK. They are the bivariate analog of critical points (points where both gradients are colinear), and they enable the extraction of filament structures in bivariate data. They correspond to *folds* of the volume when

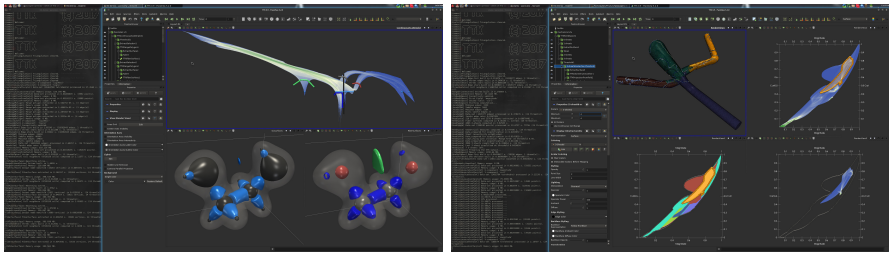

Fig. 6 Gallery of bivariate scalar data analysis. *Left:* Continuous scatterplot (top) of the electron density and reduced gradient of the ethane-diol molecule, some isosurfaces (bottom left) and fiber surfaces [6, 23] (bottom right) corresponding to the curves of matching color in the scatterplot. *Right:* Interactive continuous scatterplot peeling on fluid mechanics example (flow and curl magnitudes): a sheet of the simplified Reeb space [35] is selected by the user (orange), and its projection is independently isolated in the scatterplot for further individual inspection. Once TTK and its data package are installed (see [38] for further details), from the `ttk-data/` directory, run the following commands to reproduce these examples: "`paraview states/BuiltInExample2.pvsm`" (left) and "`paraview states/mechanical.pvsm`" (right).

projecting it to the plane according to the bivariate data. TTK also supports the fast computation of Reeb spaces of bivariate data [35], which allows the *peeling* of the continuous scatterplot in regions that do not self-overlap during the projection of the volume induced by the bivariate data. These capabilities are illustrated in Figure 6. In the left image, the user provides a few strokes on the main visual features of the continuous scatterplot (colored curves, top), and the corresponding structures in 3D are extracted as *fiber surfaces* (surfaces of matching colors, bottom). This feature definition captures subtle structures that are difficult to extract with the isosurfaces of either of the two fields of the bivariate data (bottom left). In the right image, the Reeb space segments the volume into regions that do not self-overlap when projected onto the plane given the bivariate data. Such regions can be isolated from the continutous scatterplot for further inspection. Furthermore, TTK also provides heuristics for persistence-like simplification mechanisms on bivariate Reeb spaces to enable multi-scale interactive exploration.

4 Uncertain scalar data

TTK supports the analysis of uncertain data, where the data is given as two scalar fields, representing the bounds of the interval of possible data values for each vertex of the domain. From this representation, mandatory critical points [15] can be extracted (Figure 7). These objects correspond to regions where the appearance of at least one critical point is guaranteed for any realization of the uncertain data (i.e., for any scalar field randomly generated from the input intervals). This topological analysis enables the estimation of the structures that always occur despite the uncertainty as well as their geometrical variability. This construction can be used for instance to

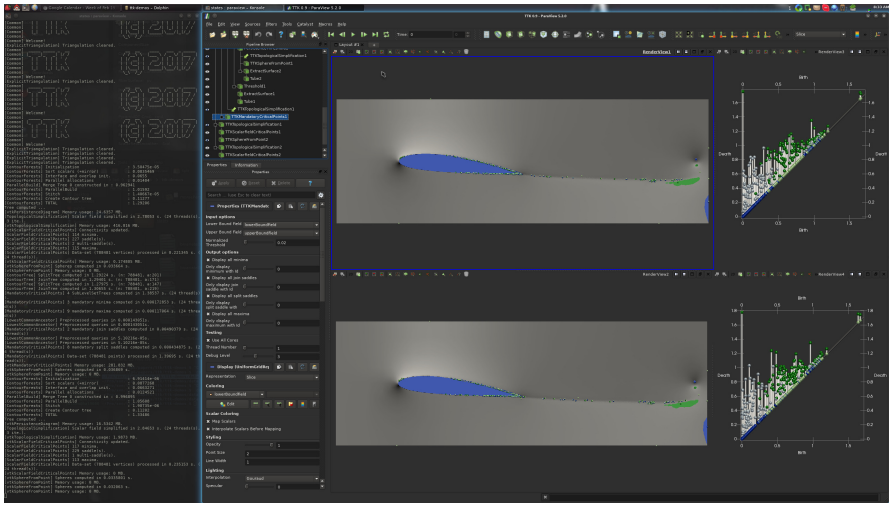

Fig. 7 Mandatory critical points [15] (colored regions) on the starting vortex example. Two members of the ensemble are shown, along with their persistence diagrams and their critical points in the domain. These critical points correspond to the vortices forming behind the wing. The most salient critical points land in the colored regions predicted by the algorithm. In this example, mandatory critical points (colored regions) help estimate visually the geometrical variability that can be expected in the locations of these vortices, given the uncertainty of the data. Once TTK and its data package are installed, from the `ttk-data/` directory, run the following command to reproduce this example: "`paraview states/uncertainStartingVortex.pvsm`" (see [38] for further details).

analyze ensemble data sets, in conjunction with clustering techniques, as illustrated by Favelier et al. [12].

5 Time-varying scalar data

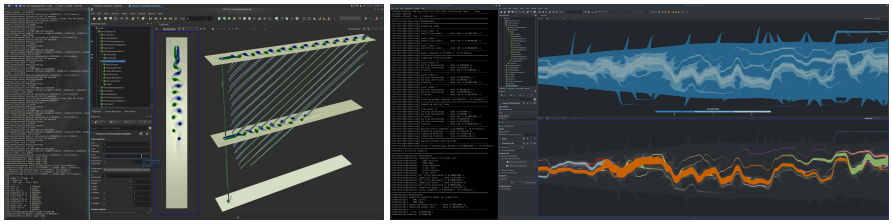

Fig. 8 Gallery of feature tracking in time-varying data. *Left:* critical point trajectory tracking with the Wasserstein matcher [32] (the height denote the temporal component). *Right:* Nested tracking graph [27] (viscous fingering data). Once TTK and its data package are installed (see [38] for further details), from the `ttk-data/` directory, run the following commands to reproduce these examples: "`paraview states/timeTracking.pvsm`" (left) and "`paraview states/nestedTrackingGraph.pvsm`".

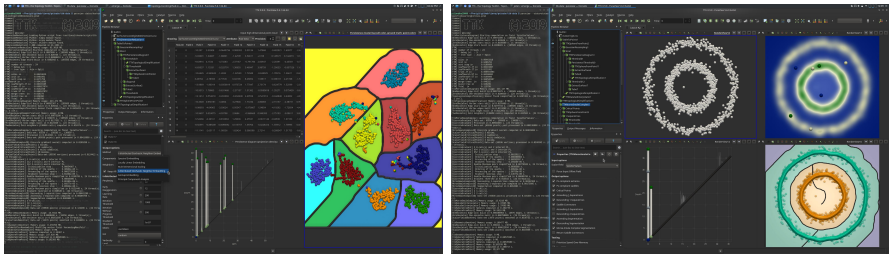

Fig. 9 Examples of topological analysis of high-dimensional point cloud data. *Left:* Persistence-driven clustering [7] of the *"mfeat"* data set (64 dimensions). The data is first projected to 2D with the t-SNE method (available from TTK's integration of scikit-learn [31]). Point colors indicate the ground-truth classification, whereas the clustering computed by TTK is reported by the background color (cells of the Morse complex). *Right:* Persistence-driven clustering [7] and beyond, on a toy point cloud example. In addition to the extraction of the correct clusters, TTK can also extract generators of the first homology group (1-dimensional cycles) with looping separatrices connecting saddles to maxima of density estimation (Gaussian kernel). Once TTK and its data package are installed (see [38] for further details), from the `ttk-data/` directory, run the following commands to reproduce these examples: "`paraview states/karhunenLoveDigits64Dimensions.pvsm`" (left) and "`paraview states/1manifoldLearningCircles.pvsm`" (right).

TTK also provides several features for the analysis and visualization of time-varying data. The trajectory of critical points through time can be tracked with the Wasserstein matcher method introduced by Soler et al. [32]. This technique enables for instance to represent the path taken by vortices in computational fluid dynamics (Figure 8, left). In addition, TTK supports the visualization and analysis of the topological evolution through time of features of interest, with the notion of nested tracking graphs [27] (Figure 8, right). These graphs encode the temporal evolution of the connected components of sub-level sets, in the form of a nested hierarchy, where each hierarchy level (each shade of blue in Figure 8, top right) correspond to a distinct isovalue (and hence a specific sub-level set).

6 High-dimensional point cloud data

TTK recently integrated the popular package *scikit-learn* [31], leveraging in particular its dimension reduction capabilities: Principal Component Analysis, Spectral Embedding, Locally Linear Embedding, Isomap, Multi-Dimensional Scaling, t-distributed Stochastic Neighbor Embedding. Then, high-dimensional point cloud data (typically in the form of a CSV file) can be processed by TTK. Typically, the data is first projected to 2D or 3D with one of the above dimension reduction methods (Figure 9). Next, a density estimation (e.g., Gaussian kernel) is performed on a regular grid to describe the projection of the input point cloud (Figure 9, top right). From this point, any tool of the TTK arsenal can be employed to further analyze, visualize, and explore the data. For instance, persistence-driven clustering [7] can easily be deployed with TTK. The k most persistent features can be selected from

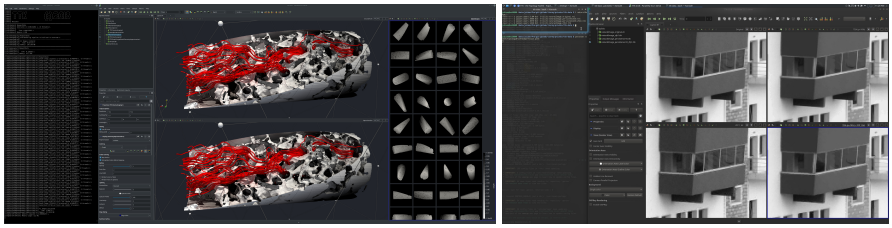

Fig. 10 Examples of in situ data reduction with TTK. *Left:* View-based surface approximation [26] (top: ground-truth, bottom: approximation). *Right:* Topology-controlled lossy compression [33]. Once TTK and its data package are installed (see [38] for further details), from the `ttk-data/` directory, run the following commands to reproduce these examples: "`paraview states/geometryApproximation.pvsm`" (left) and "`paraview states/persistenceDrivenCompression.pvsm`" (right).

the persistence diagram (Figure 9) to drive a pre-simplification of the data, in order to control the number of clusters (where k is the number of desired clusters). Note that, in practice, a relevant value of k can often be visually inferred from the flat plateaus of the persistence curve (see Figure 2, top right), similarly to the notion of eigen gap [28] in spectral clustering. Next, the Morse complex can be extracted to isolate each basin of attraction of each of the k remaining maxima (Figure 9, bottom right, where two clusters are extracted, corresponding to the two *rings* present in the data). The final clustering can be projected from the cells of the Morse complex to the input point cloud with TTK's generic interpolator. Note that TTK enables topological explorations that go beyond simple clustering, such as the extraction of generators of homology groups, as illustrated in Figure 9 (bottom, right), where looping separatrices linking saddles to maxima are used to extract such generators, hence visually conveying to the user additional information about the internal structure of each cluster. In particular, such generators, when mapped back onto the data, provide visual hints that enable users to identify to which cycle a given data point belongs to. Moreover, it also helps users appreciate the importance of a given group of cycles, given the size of its generator in the data. The left example of Figure 9 further illustrates the clustering capabilities of TTK on the *mfeat* data set (64 dimensions, 2000 points). The ground-truth classification is given by the colors on the points, whereas the non-supervised classification obtained from the topological clustering is given by the background color (one color per cell of the Morse complex). This example nicely illustrates how TTK can effectively help visualize the intrinsic structure of high-dimensional data.

7 In situ topological analysis

TTK can be efficiently run in situ (i.e., directly from a simulation source code without storing data to disk) using the Catalyst API [3]. TTK's website reports a complete tutorial [40] with the open-source fluid mechanic simulation code *Code_Saturne*

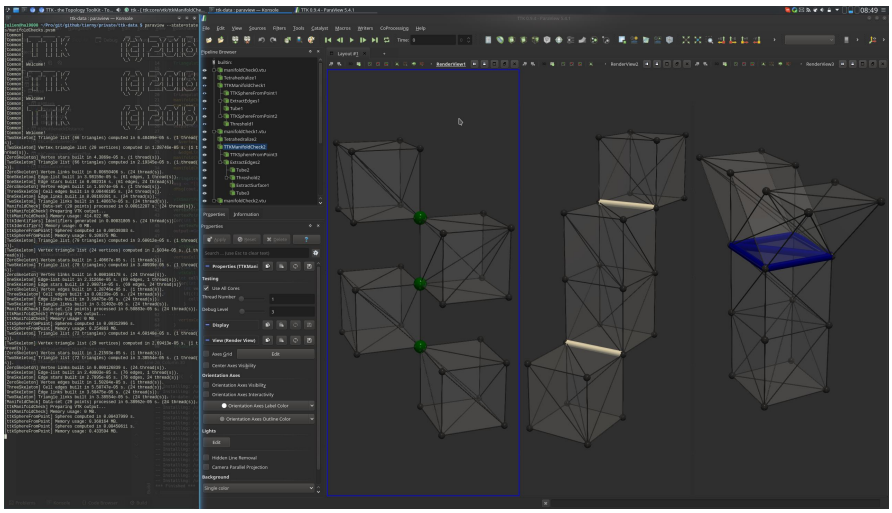

Fig. 11 Example of convenience TTK module: check for manifold-ness on several simplicial complexes. Non-manifold vertices, edges and triangles are reported in green, white, and blue respectively. Once TTK and its data package are installed, from the `ttk-data/` directory, run the following command to reproduce this example: "`paraview states/manifoldChecks.pvsm`" (see [38] for further details).

[10], where TDA capabilities are run on the file, without data storage, after each computation of a simulation time step.

In addition, TTK offers lossy compression and data reduction tools, to allow the in situ storage of reduced information. In particular, regular grid data can be saved in the TTK file format (*.*ttk*), which implements the topologically controlled compression framework by Soler et al. [33]. This framework enables to compress data in a lossy way while guaranteeing the exact preservation of the persistence diagrams of the most salient features. This methodology guarantees, in practice, that any topological analysis run on the compressed data is faithful to the original data. TTK also implements the award-winning image-based geometry approximation method by Lukasczyk et al. [26]. Additionally, TTK implements the latest specification of Cinema databases [2], which enables users to interactively explore large ensembles of data sets stored as Cinema databases and to apply specific analysis pipelines to selections of members, expressed with SQL queries on the meta-data of the members.

8 Convenience

Finally, TTK provides a number of features that make its deployment more convenient for users, including generic data interpolators (interpolating data from any type of object onto any type of object), convertors, mesh processing, and analysis (subdivision, point merging, manifold checks, Figure 11, etc.).

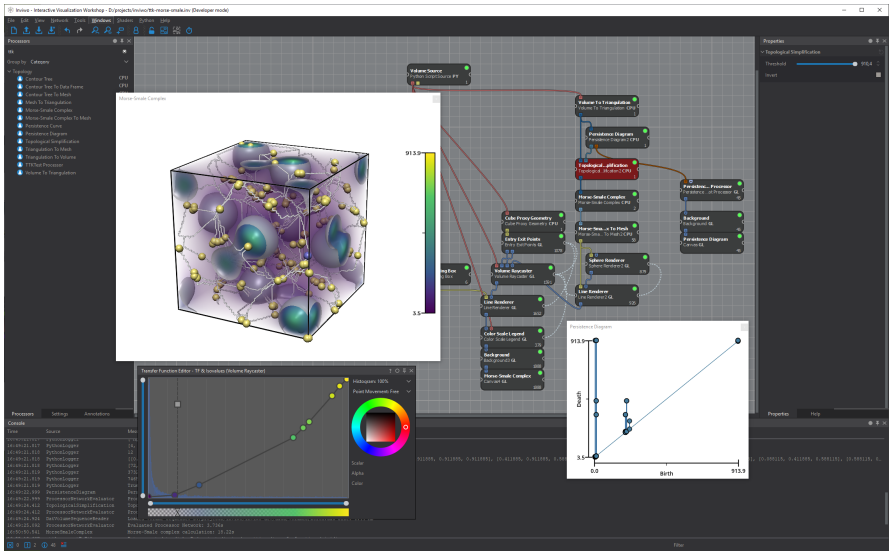

Fig. 12 Integration of TTK in a software ecosystem other than VTK/ParaView: Inviwo [21], a software framework for the rapid prototyping of visualizations, written in C++ and exploiting modern graphics hardware. This example shows the topological analysis with the Morse-Smale complex (with persistence-driven data pre-simplification) of charge densities in iron oxide [20].

9 Conclusion and perspectives

This paper presented a brief overview of the main end-user features available in the Topology ToolKit (TTK) along with example application scenarios. The material that is necessary to reproduce these examples is available on the TTK website [38]. The data analysis pipelines presented in this paper can be easily reproduced with ParaView, with Python scripts (ParaView supports the automatic export of analysis pipelines to Python scripts), with VTK or direct C++ code. The examples illustrated in this paper ranged from basic image segmentation capabilities to the advanced topological analysis of high-dimensional point cloud data. We refer the reader to TTK's online user forum for further discussions and usage examples [39].

In the future, we are looking forward to further extending TTK's developer and user communities. We see TTK as an opportunity to grow as a community by federating our software engineering efforts, to make our research more accessible, reproducible and visible to others. In that regard, we warmly welcome contributors with experience in vector and tensor data analysis. We will also work toward the improved integration of TTK in third-party data analysis and visualization tools, as done, for example, in collaboration with the Inviwo [21] development team (Figure 12). Future directions of development of TTK include an improved support for statistical tasks based on topological data representations as well as an improved integration of TTK on supercomputers. Such improvements will be partially conducted in the context of the *VESTEC* project [43], which focuses on novel supercomputing

methodologies for urgent decision making, and for which TTK is one of the core software technologies.

Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful remarks and suggestions. This work is partially supported by the European Commission grant H2020-FETHPC-2017 “VESTEC” (ref. 800904).

References

1. Ahrens, J., Geveci, B., Law, C.: Paraview: An end-user tool for large-data visualization. The Visualization Handbook (2005)
2. Ahrens, J., Jourdain, S., O’Leary, P., Patchett, J., Rogers, D.H., Petersen, M.: An image-based approach to extreme scale in situ visualization and analysis. In: IEEE SuperComputing (2014)
3. Ayachit, U., Bauer, A.C., Geveci, B., O’Leary, P., Moreland, K., Fabian, N., Mauldin, J.: Paraview catalyst: Enabling in situ data analysis and visualization. In: In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, ISAV 2015 (2015)
4. Bachthaler, S., Weiskopf, D.: Continuous scatterplots. IEEE TVCG (2008)
5. Banchoff, T.F.: Critical points and curvature for embedded polyhedral surfaces. Amer. Math. Monthly (1970)
6. Carr, H., Geng, Z., Tierny, J., Chattopadhyay, A., Knoll, A.: Fiber surfaces: Generalizing isosurfaces to bivariate data. CGF (2015)
7. Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Persistence-based clustering in riemannian manifolds. J. ACM (2013)
8. Edelsbrunner, H., Harer., J.: Jacobi Sets of Multiple Morse Functions. Cambridge Books Online (2004)
9. Edelsbrunner, H., Harer, J.: Computational Topology: An Introduction. AMS (2009)
10. EDF: Code_saturne. `https://www.code-saturne.org/cms/`
11. Falk, M., Garth, C., Gueunet, C., Levine, J.A., Lukasczyk, J., Tierny, J., Vidal, J.: Topological Data Analysis Made Easy with the Topology ToolKit, A Sequel. In: Proc. of IEEE VIS Tutorials (2019). `https://topology-tool-kit.github.io/ieeeVis2019Tutorial.html`
12. Favelier, G., Faraj, N., Summa, B., Tierny, J.: Persistence Atlas for Critical Point Variability in Ensembles. IEEE TVCG (2018)
13. Favelier, G., Gueunet, C., Gyulassy, A., Jomier, J., Levine, J., Lukasczyk, J., Sakurai, D., Soler, M., Tierny, J., Usher, W., Wu, Q.: Topological data analysis made easy with the Topology ToolKit. In: Proc. of IEEE VIS Tutorials (2018). `https://topology-tool-kit.github.io/ieeeVis2018Tutorial.html`
14. Favelier, G., Gueunet, C., Tierny, J.: Visualizing ensembles of viscous fingers. In: IEEE SciVis Contest (2016)
15. Guenther, D., Salmon, J., Tierny, J.: Mandatory critical points of 2D uncertain scalar fields. CGF (2014)
16. Gueunet, C., Fortin, P., Jomier, J., Tierny, J.: Contour Forests: Fast Multi-threaded Augmented Contour Trees. In: Proc. of IEEE LDAV (2016)
17. Gueunet, C., Fortin, P., Jomier, J., Tierny, J.: Task-based Augmented Merge Trees with Fibonacci heaps. In: LDAV (2017)
18. Gueunet, C., Fortin, P., Jomier, J., Tierny, J.: Task-based Augmented Contour Trees with Fibonacci heaps. IEEE TPDS (2019)

19. Gueunet, C., Fortin, P., Jomier, J., Tierny, J.: Task-based Augmented Reeb Graphs with Dynamic ST-Trees. In: Eurographics Symposium on Parallel Graphics and Visualization (2019)
20. Jakobsson, E., Bin-Masood, T., Hotz, I., Abrikosov, I., Steneteg, P.: Topology-guided analysis and visualization of charge density fields : A case study (2019). Submitted manuscript.
21. Jönsson, D., Steneteg, P., Sundén, E., Englund, R., Kottravel, S., Falk, M., Ynnerman, A., Hotz, I., Ropinski, T.: Inviwo – a visualization system with usage abstraction levels. IEEE TVCG (2019). DOI 10.1109/TVCG.2019.2920639. https://inviwo.org/
22. Kerber, M., Morozov, D., Nigmetov, A.: Geometry helps to compare persistence diagrams. ACM Journal of Experimental Algorithmics (2016)
23. Klacansky, P., Tierny, J., Carr, H.A., Geng, Z.: Fast and exact fiber surfaces for tetrahedral meshes. IEEE TVCG (2017)
24. Kontak, M., Vidal, J., Tierny, J.: Statistical Parameter Selection for Clustering Persistence Diagrams. In: Proc. of SuperComputing workshop on Urgent HPC (2019)
25. Lacombe, T., Cuturi, M., Oudot, S.: Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport. In: NIPS (2018)
26. Lukasczyk, J., Kinner, E., Ahrens, J., Leitte, H., Garth, C.: Voidga: A view-approximation oriented image database generation approach. In: Proc. of IEEE LDAV (2018)
27. Lukasczyk, J., Weber, G.H., Maciejewski, R., Garth, C., Leitte, H.: Nested tracking graphs. CGF (2017)
28. von Luxburg, U.: A tutorial on spectral clustering. In: Statistics and Computing (2007)
29. Morozov, D.: Dionysus. http://www.mrzv.org/software/dionysus (2010)
30. Pascucci, V., Tricoche, X., Hagen, H., Tierny, J.: Topological Data Analysis and Visualization: Theory, Algorithms and Applications. Springer (2010)
31. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. JMLR (2011)
32. Soler, M., Plainchault, M., Conche, B., Tierny, J.: Lifted wasserstein matcher for fast and robust topology tracking. In: Proc. of IEEE LDAV (2018)
33. Soler, M., Plainchault, M., Conche, B., Tierny, J.: Topologically controlled lossy compression. In: Proc. of PV (2018)
34. Tierny, J.: Topological Data Analysis for Scientific Visualization. Springer (2018)
35. Tierny, J., Carr, H.: Jacobi fiber surfaces for bivariate reeb space computation. IEEE TVCG (2016)
36. Tierny, J., Favelier, G., Levine, J.A., Gueunet, C., Michaux, M.: The Topology ToolKit. IEEE TVCG (2017). https://topology-tool-kit.github.io/
37. Tierny, J., Pascucci, V.: Generalized topological simplification of scalar fields on surfaces. IEEE TVCG (2012)
38. TTK-Contributors: TTK Online Tutorials. https://topology-tool-kit.github.io/tutorials.html
39. TTK-Contributors: TTK User Forum. https://groups.google.com/forum/#!forum/ttk-users
40. TTK-Contributors: Tutorial on In-situ Topological Data Analysis with TTK and Catalyst. https://topology-tool-kit.github.io/catalyst.html
41. TTK-Contributors: TTK Anaconda package (2019). https://anaconda.org/conda-forge/topologytoolkit
42. Turner, K., Mileyko, Y., Mukherjee, S., Harer, J.: Fréchet Means for Distributions of Persistence Diagrams. Disc. Compu. Geom. (2014)
43. VECSTEC-Consortium: Visual Exploration and Sampling ToolKit for Extreme Computing. https://vestec-project.eu/
44. Vidal, J., Budin, J., Tierny, J.: Progressive Wasserstein Barycenters of Persistence Diagrams. IEEE TVCG (2019)
45. Vintescu, A., Dupont, F., Lavoué, G., Memari, P., Tierny, J.: Conformal factor persistence for fast hierarchical cone extraction. In: Eurographics (short papers) (2017)
46. Xu, K., Zhang, H., Cohen-Or, D., Xiong, Y.: Dynamic harmonic fields for surface processing. Computers & Graphics (2009)