An Overview of the Topology ToolKit

Martin Falk, Guillaume Favelier, Charles Gueunet, Pierre Guillou, Adhitya Kamakshidasan, Pavol Klacansky, Joshua Levine, Jonas Lukasczyk, Daisuke Sakurai, Maxime Soler, Julien Tierny, Will Usher, Jules Vidal, Michal Wozniak

Martin Falk
Linköping University. e-mail: martin.falk@liu.se

Guillaume Favelier
INRIA. e-mail: guillaume.favelier@inria.fr

Charles Gueunet
Kitware. e-mail: charles.gueunet@kitware.com

Pierre Guillou
Sorbonne Université. e-mail: pierre.guillou@sorbonne-universite.fr

Adhitya Kamakshidasan
INRIA. e-mail: adhitya.kamakshidasan@inria.fr

Pavol Klacansky
SCI Institute, University of Utah. e-mail: klacansky@sci.utah.edu

Joshua Levine
University of Arizona. e-mail: josh@email.arizona.edu

Jonas Lukasczyk
TU Kaiserslautern. e-mail: jl@jluk.de

Daisuke Sakurai
Kyushu University. e-mail: d.sakurai@computer.org

Maxime Soler
Total / Sorbonne Université. e-mail: soler.maxime@total.com

Julien Tierny
CNRS / Sorbonne Université. e-mail: julien.tierny@sorbonne-universite.fr

Will Usher
SCI Institute, University of Utah. e-mail: will@sci.utah.edu

Jules Vidal
Sorbonne Université. e-mail: jules.vidal@sorbonne-universite.fr

Michal Wozniak
Caboma Inc. e-mail: michal.wozniak@caboma.com

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 [32] (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 [34].

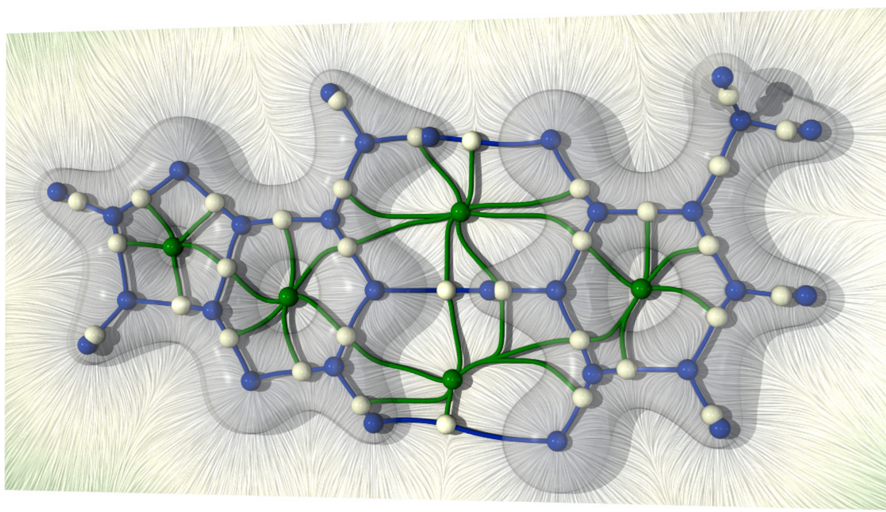

Fig. 1 Extraction of the covalent and non-covalent interactions in a molecular system with TTK. Covalent and hydrogen bonds are captured by the blue separatrices of the Morse-Smale complex, and steric repulsion is captured by saddle connectors (green).

1 Introduction

Topological data analysis (TDA) [9, 26, 30] is a vibrant field of study at the cross roads between mathematics and computer science, that suggests to look at complex data under the perspective of its *structure*. 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 particularly useful in a variety of fields, ranging from machine learning [7] to geometry processing [38]. In scientific applications, TDA is particularly effective for the analysis of large-scale data sets [13]. The *Topology*

ToolKit (TTK) [32] is an open-source library for TDA that has been released in 2017 under the permissive BSD license. It features a generic, efficient, and substantial collection of implementations of reference TDA algorithms. TTK is mostly written in C++ (~100k lines of code) to offer the best possible performances. To date, 11 institutions have contributed code to TTK, including 8 academic organizations (CNRS, INRIA, Linkoping University, Sorbonne Universite, TU Kaiserslautern, University of Arizona, University of Utah, Zuse Institute Berlin) and 3 companies (Kitware, Total, Caboma). Since its initial release, TTK's website has collected more than 135k page-views, from more than 14k unique visitors, and its video tutorials have collected more than 8.5k 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]. Data can be provided to TTK in multiple forms: it can be sampled along 1D, 2D, or 3D regular grids, 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 [32], its end-user features have not been formally presented, other than in oral tutorials [12]. 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 item 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, ParaView state files, etc.) is available on the TTK website (section *Tutorials* [34]).

2 Scalar data

TTK suppports the computation of a large number of topological abstractions for scalar data. Critical points [5] can be used to extract points of interest. Merge/contour trees and Reeb graphs [15, 16, 17, 18] 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, 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. To reflect this noise removal on the original data, *Topological Simplification* [33] is implemented in TTK. Then, any topological object mentioned above (critical point, merge/contour tree, Reeb graph, Morse-Smale complex) computed after this data simplification step will therefore be simplified, allowing multi-scale feature exploration.

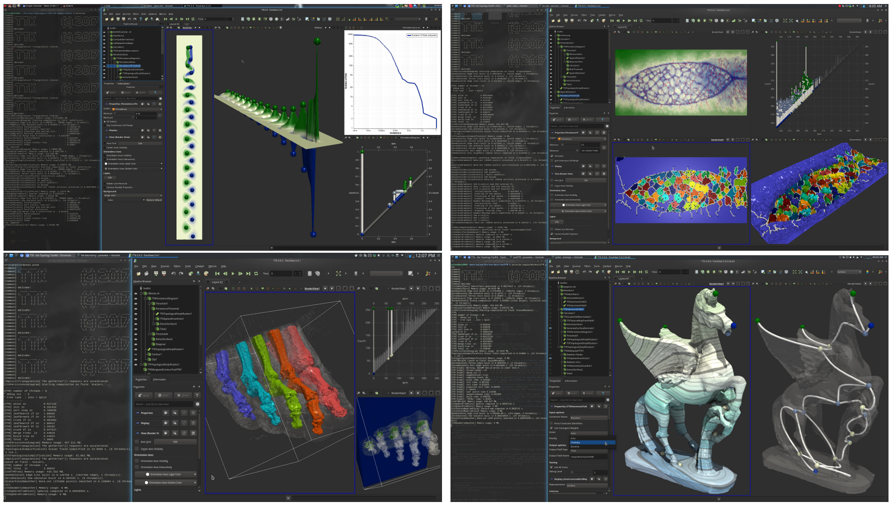

Fig. 2 Gallery of scalar data analysis. *Top left:* Typical persistence-driven analysis pipeline applied to vortex tracking in computational fluid dynamics. *Top right:* Typical persistence-driven analysis pipeline, combined with Morse complex computation for cell enumeration in confocal microscopy (example reproduced from [9], page 217). *Bottom left:* Typical persistence-driven merge-tree based segmentation applied to bone extraction in medical imaging. *Bottom right:* Skeleton estimation from the Reeb graph [18] of a user designed harmonic field [39].

Note that TTK also offers functionalities to design harmonic scalar fields by solving the Laplace equations subject to Dirichlet constraints [39] provided by the user at key locations (typically at extremities of prominent shape features). TTK also implements efficient algorithms [25, 28] for the estimation of distances between Persistence diagrams (such as the Bottleneck and Wasserstein distances [9]).

Figure 2 provides 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). This simplification is combined with critical point extraction (top left) to extract the center of each vortex in a computational fluid dynamic example. The simplification is combined with the Morse complex (top right) to extract cells in confocal microscopy. In this example, the segmentation is obtained by representing the manifold of each local maximum with a distinct color. Data pre-simplification is combined with the merge tree (bottom left) 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. Here the level of persistence has been tuned to maintain only the five most persistent features (corresponding to the bones of the foot). Maintaining more features (in this example, for a persistence threshold of 150) would precisely segment the bones along each join, which further illustrates the potential for multi-scale data exploration. The last example (bottom right) illustrates the scalar field design capabilities of TTK (with harmonic fields) for skeleton extraction (with the Reeb graph).

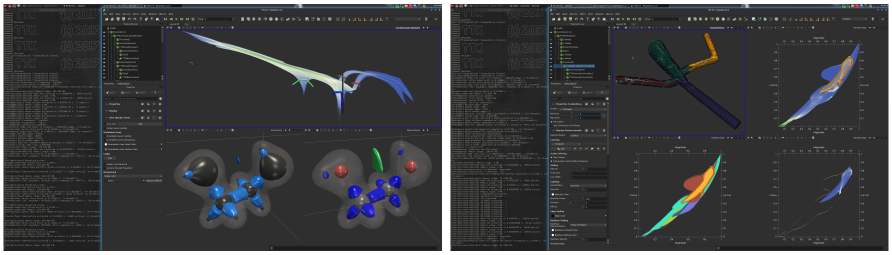

Fig. 3 Gallery of bivariate scalar data analysis. *Left:* Continuous scatterplot (top) of the electron density and reduced gradient of the ethane-diol molecule, some isosurfaces (bottom left) and fiber surfaces [6, 21] (bottom right) corresponding to the curves of matching color in the scatterplot. *Right:* Interactive continuous scatterplot peeling on fluid mechanics example (flow and curl magnitudes): a sheet of the simplified Reeb space [31] is selected by the user (orange), and its projection is independently isolated in the scatterplot for further individual inspection.

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 to understand where and how volumetric data projects to the data range. Fiber surfaces [6, 21] 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 projecting it to the plane according to the bivariate data. TTK also supports the fast computation of Reeb spaces of bivariate data [31], 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 3. 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 allows to capture 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.

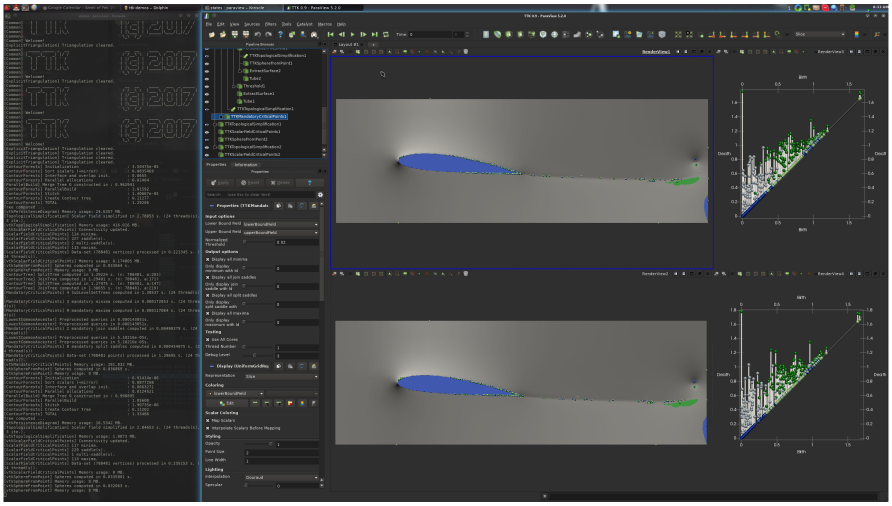

Fig. 4 Mandatory critical points [14] (colored regions) on the starting vortex example. Two members of the ensemble are shown, along with their persistence diagrams and their critical points in the domain. These critical points correspond to the vortices forming behind the wing. The most salient critical 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.

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 [14] can be extracted (Figure 4). 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, in practice, 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 analyze ensemble data sets, in conjunction with clustering techniques, as illustrated by Favelier et al. [11].

5 Time-varying scalar data

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. [28]. This technique enables for instance to represent the path taken by vortices in computational fluid dynamics (Figure 5,

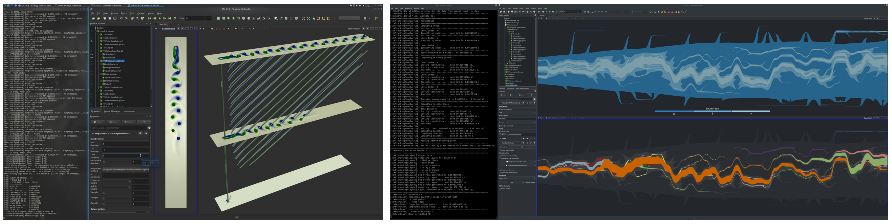

Fig. 5 Gallery of feature tracking in time-varying data. *Left:* critical point trajectory tracking with the Wasserstein matcher [28] (the height denote the temporal component). *Right:* Nested tracking graph [23] (viscous fingering data).

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 [23] (Figure 5, right), which enables, in particular, the representation of the temporal evolution of nested structures (e.g., sub-level sets for distinct isovalues).

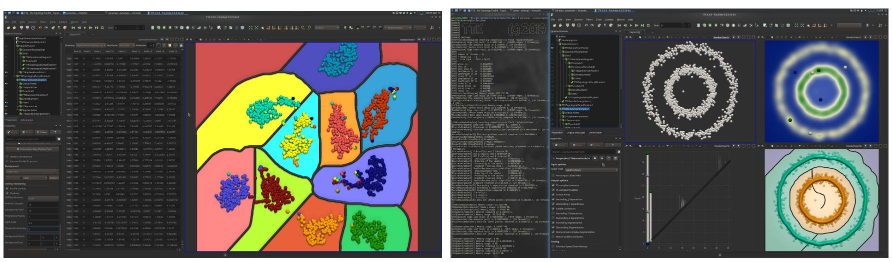

Fig. 6 Examples of topological analysis of high-dimensional point cloud data. *Left:* Persistence-driven clustering [7] of the *"mfeat"* data set (64 dimensions). The data is first projected to 2D with the t-SNE method (available from TTK's integration of scikit-learn [27]). Point colors indicate the 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).

6 High-dimensional point cloud data

TTK recently integrated the popular package *scikit-learn* [27], 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 6). Next, a density estimation (e.g., Gaussian kernel) is performed on a regular grid to describe the projection of the input point cloud (Figure 6, 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 the persistence diagram (Figure 6) 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 left), similarly to the notion of eigen gap [24] in spectral clustering. Next, the Morse complex can be extracted to isolate each basin of attraction of each of the k remaining maxima (Figure 6, 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 6 (bottom, right), where looping separatrices linking saddles to maxima are used to extract such generators, hence conveying to the user additional information about the internal structure of each cluster. The left example of Figure 6 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

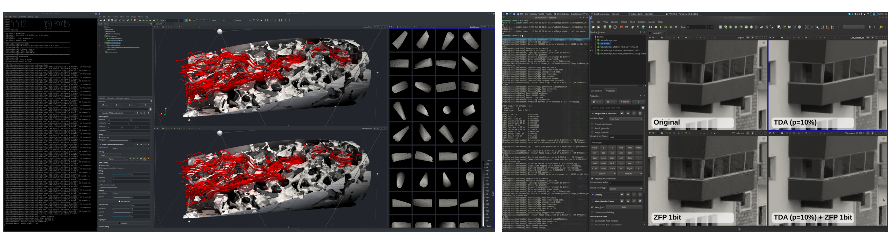

Fig. 7 Examples of in situ data reduction with TTK. *Left:* View-based surface approximation [22] (top: ground-truth, bottom: approximation). *Right:* Topology-controlled lossy compression [29].

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 [36] with the open-source fluid mechanic simulation code *Code_Saturne*

[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. [29]. 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. [22]. 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

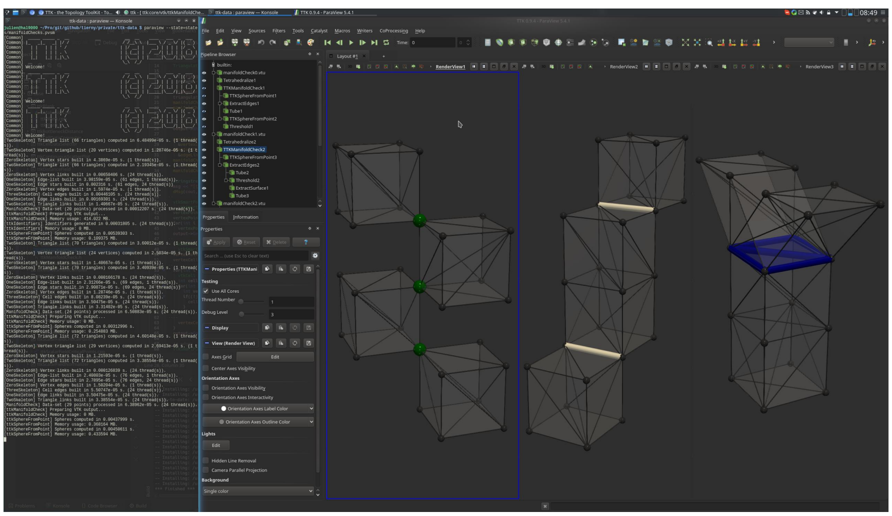

Fig. 8 Example of convenience TTK module: check for manifold-ness on several simplicial complexes. Non-manifold vertices, edges and triangles are reported in green, white, and blue respectively.

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 8, etc.).

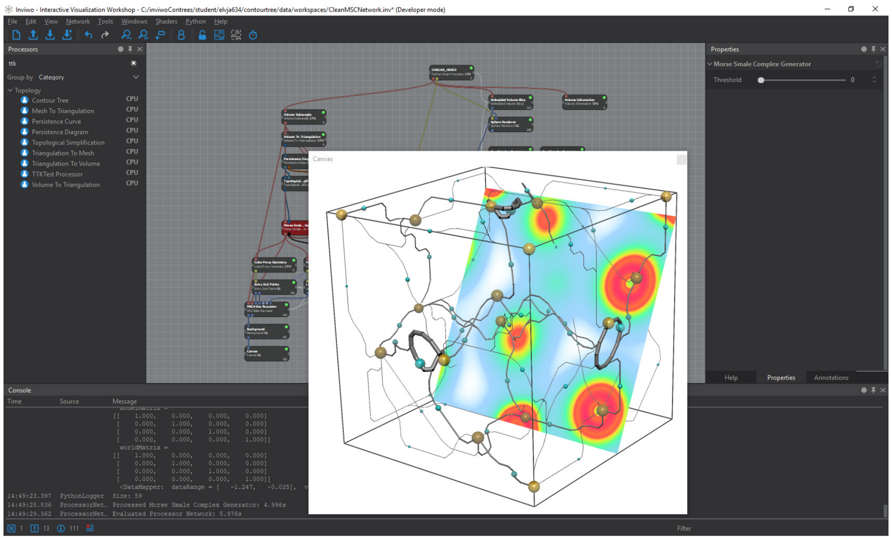

Fig. 9 Integration of TTK in a software ecosystem other than VTK/ParaView: Inviwo [20], a software framework for the rapid prototyping of visualizations, written in C++ and exploiting modern graphics hardware. This example shows the topological analysis with the Morse-Smale complex (with persistence-driven data pre-simplification) of charge densities in iron oxide [19].

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 [34]. 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 [35].

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 [20] development team (Figure 9). 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 conducted, in particular, in the context of the *VESTEC* project [37], which focuses on novel supercomputing methodologies for urgent decision making, and for which TTK is one of the core software technologies.

Acknowledgments

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: Proc. of 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. Favelier, G., Faraj, N., Summa, B., Tierny, J.: Persistence Atlas for Critical Point Variability in Ensembles. IEEE TVCG (2018)
12. 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/ieeeVisTutorial.html`
13. Favelier, G., Gueunet, C., Tierny, J.: Visualizing ensembles of viscous fingers. In: IEEE SciVis Contest (2016)
14. Guenther, D., Salmon, J., Tierny, J.: Mandatory critical points of 2D uncertain scalar fields. CGF (2014)
15. Gueunet, C., Fortin, P., Jomier, J., Tierny, J.: Contour Forests: Fast Multi-threaded Augmented Contour Trees. In: Proc. of IEEE LDAV (2016)
16. Gueunet, C., Fortin, P., Jomier, J., Tierny, J.: Task-based Augmented Merge Trees with Fibonacci heaps. In: LDAV (2017)
17. Gueunet, C., Fortin, P., Jomier, J., Tierny, J.: Task-based Augmented Contour Trees with Fibonacci heaps. IEEE TPDS (2019)
18. 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)

19. 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.
20. 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/`
21. Klacansky, P., Tierny, J., Carr, H.A., Geng, Z.: Fast and exact fiber surfaces for tetrahedral meshes. IEEE TVCG (2017)
22. 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)
23. Lukasczyk, J., Weber, G.H., Maciejewski, R., Garth, C., Leitte, H.: Nested tracking graphs. CGF (2017)
24. von Luxburg, U.: A tutorial on spectral clustering. In: Statistics and Computing (2007)
25. Morozov, D.: Dionysus. `http://www.mrzv.org/software/dionysus` (2010)
26. Pascucci, V., Tricoche, X., Hagen, H., Tierny, J.: Topological Data Analysis and Visualization: Theory, Algorithms and Applications. Springer (2010)
27. 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)
28. Soler, M., Plainchault, M., Conche, B., Tierny, J.: Lifted wasserstein matcher for fast and robust topology tracking. In: Proc. of IEEE LDAV (2018)
29. Soler, M., Plainchault, M., Conche, B., Tierny, J.: Topologically controlled lossy compression. In: Proc. of PV (2018)
30. Tierny, J.: Topological Data Analysis for Scientific Visualization. Springer (2018)
31. Tierny, J., Carr, H.: Jacobi fiber surfaces for bivariate reeb space computation. IEEE TVCG (2016)
32. Tierny, J., Favelier, G., Levine, J.A., Gueunet, C., Michaux, M.: The Topology ToolKit. IEEE TVCG (2017). `https://topology-tool-kit.github.io/`
33. Tierny, J., Pascucci, V.: Generalized topological simplification of scalar fields on surfaces. IEEE TVCG (2012)
34. TTK-Contributors: TTK Online Tutorials. `https://topology-tool-kit.github.io/tutorials.html`
35. TTK-Contributors: TTK User Forum. `https://groups.google.com/forum/#!forum/ttk-users`
36. TTK-Contributors: Tutorial on In-situ Topological Data Analysis with TTK and Catalyst. `https://topology-tool-kit.github.io/catalyst.html`
37. VECSTEC-Consortium: Visual Exploration and Sampling ToolKit for Extreme Computing. `https://vestec-project.eu/`
38. Vintescu, A., Dupont, F., Lavoué, G., Memari, P., Tierny, J.: Conformal factor persistence for fast hierarchical cone extraction. In: Eurographics (short papers) (2017)
39. Xu, K., Zhang, H., Cohen-Or, D., Xiong, Y.: Dynamic harmonic fields for surface processing. Computers & Graphics (2009)