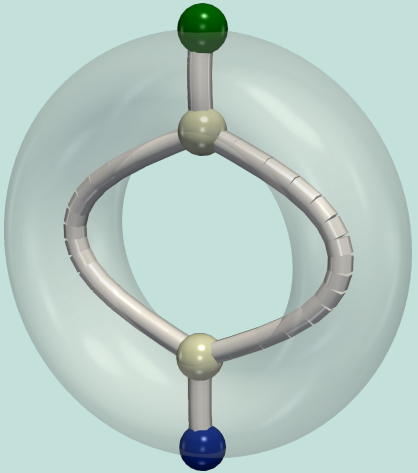
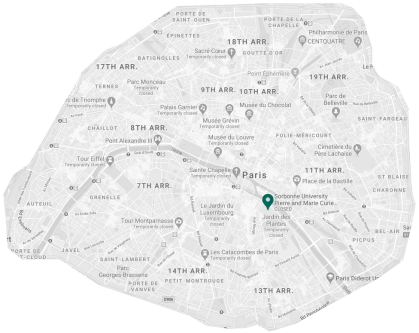
An Introduction to the Topology ToolKit



Julien Tierny

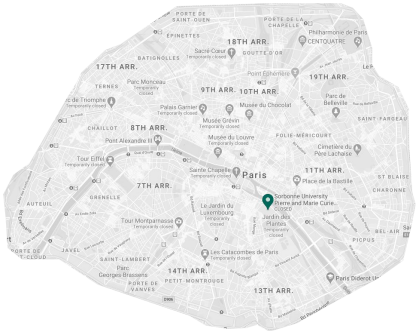
About our research

- **TDA @ Sorbonne**
  - Topological data analysis and visualization
    - Interactive data analysis
    - Data from engineering & science
  - Computational aspects
    - Practical algorithms
    - Emerging data types



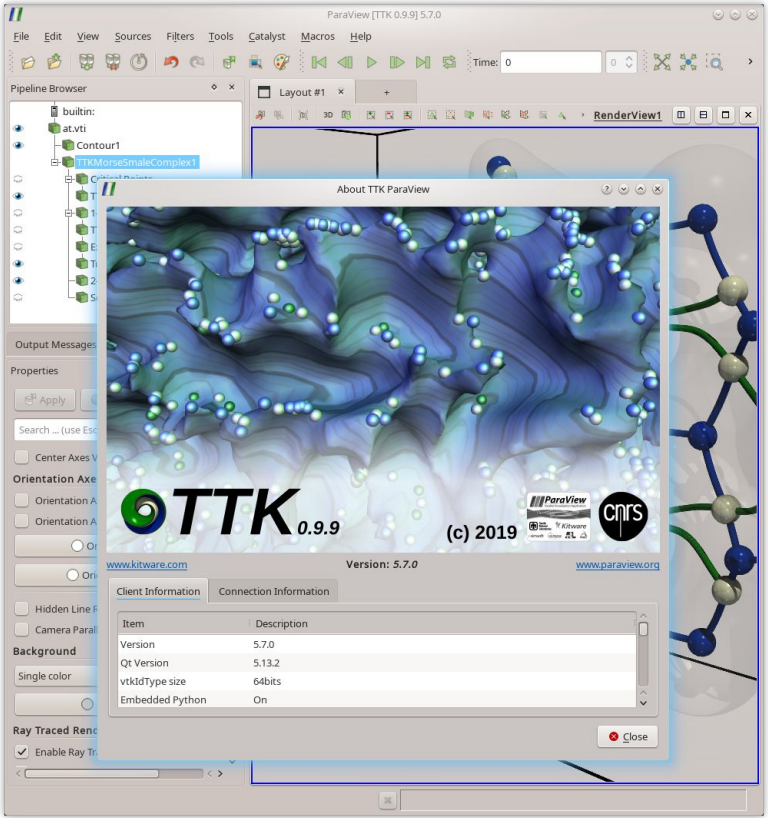
About our research

- **TDA @ Sorbonne**
  - Topological data analysis and visualization
    - Interactive data analysis
    - Data from engineering & science
  - Computational aspects
    - Practical algorithms
    - Emerging data types
  - Main venues
    - IEEE VIS, EuroVis, IEEE LDAV, TopoInVis



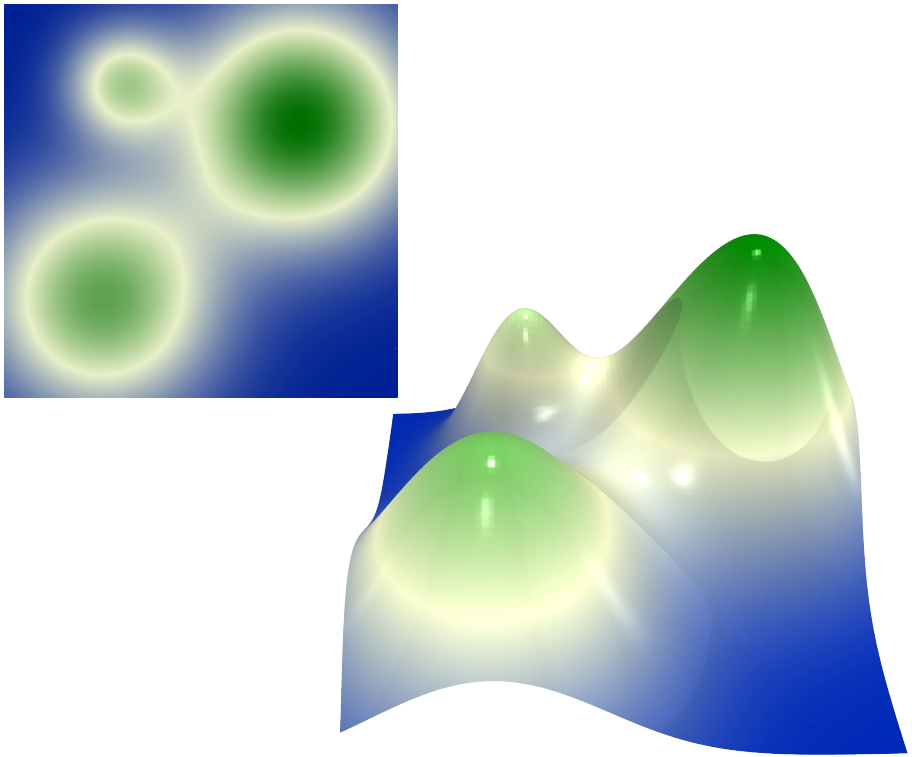
About TTK

- **Initially**
  - Internal software platform (since 2015)
- **Public release (2017)**
  - BSD license, 115k lines of C++
- **Contributing institutions**
  - Academia (12):
    - Arizona State University, CNRS, Heidelberg University, INRIA, Linkoping University, Los Alamos National Laboratory, Sorbonne Universite, TU Kaiserslautern, University of Arizona, University of Leeds, University of Utah, Zuse Institute Berlin
  - Companies (3):
    - Kitware, ShapeShift3D, Total



Why using TTK?

- **What is it good for?**
  - Low dimensional data
  - Continuous scalar fields
  - Science & engineering
    - Astrophysics, biological imaging, quantum chemistry, fluid dynamics, material sciences
- **What is it not good for? (yet)**
  - Vector / tensor data
  - High dimensional data

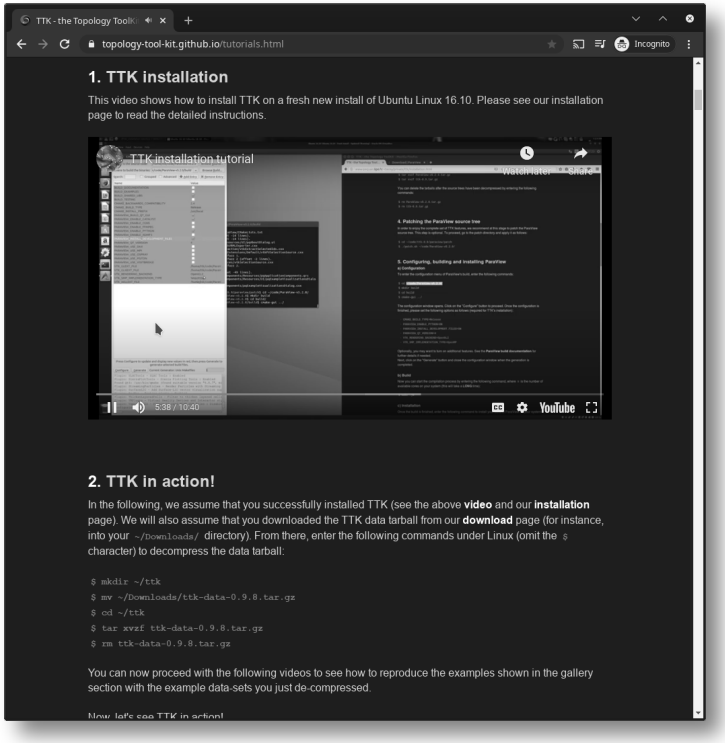


Online resources

- **https://topology-tool-kit.github.io/**
  - Installation instructions
  - Gallery
  - Data and examples
  - Video tutorials
  - Documentation
  - Exercises
  - Mailing lists
    - **ttk-users@googlegroups.com**

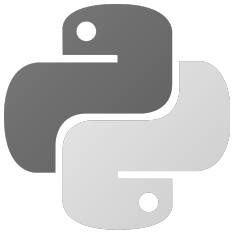
Installation

- **Recommended**
  - From source
  - Step-by-step instructions
    - Linux, MacOS, Windows
- **To simply give it a try**
  - Virtualbox & docker images
- **For python usage only**
  - Anaconda package
    - `conda install -c conda-forge topologytoolkit`
- **For the latest features**
  - Github repository
    - **https://github.com/topology-tool-kit/ttk**



How should I interact with TTK?

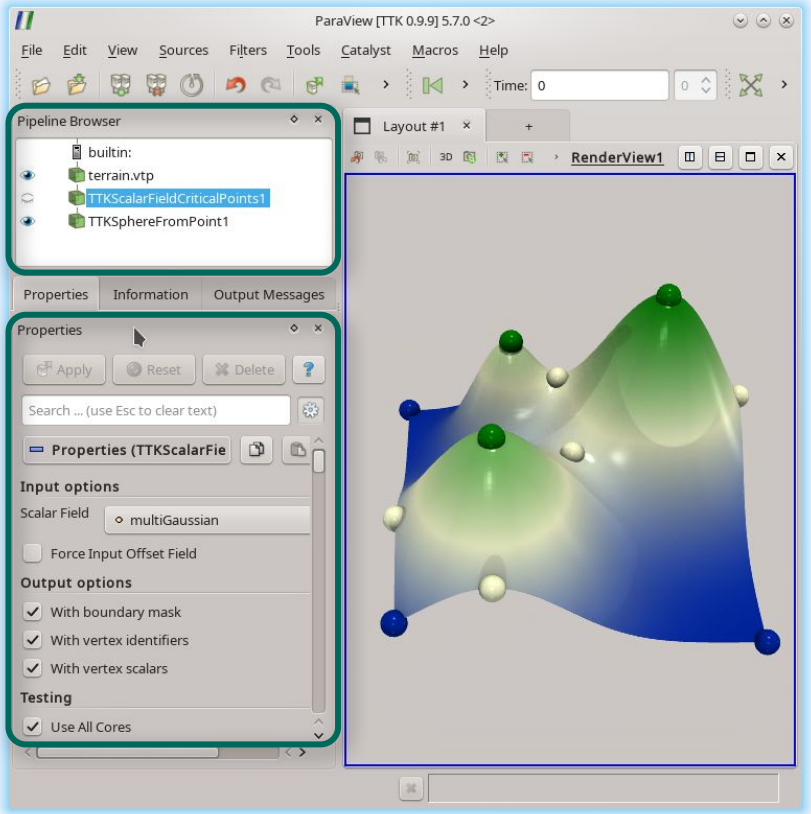
- **End users**
  - Plugins for ParaView
    - De-facto standard in scientific computing
  - Light Python API
    - Fast scripting
  - Inviwo environment
    - **https://inviwo.org/**



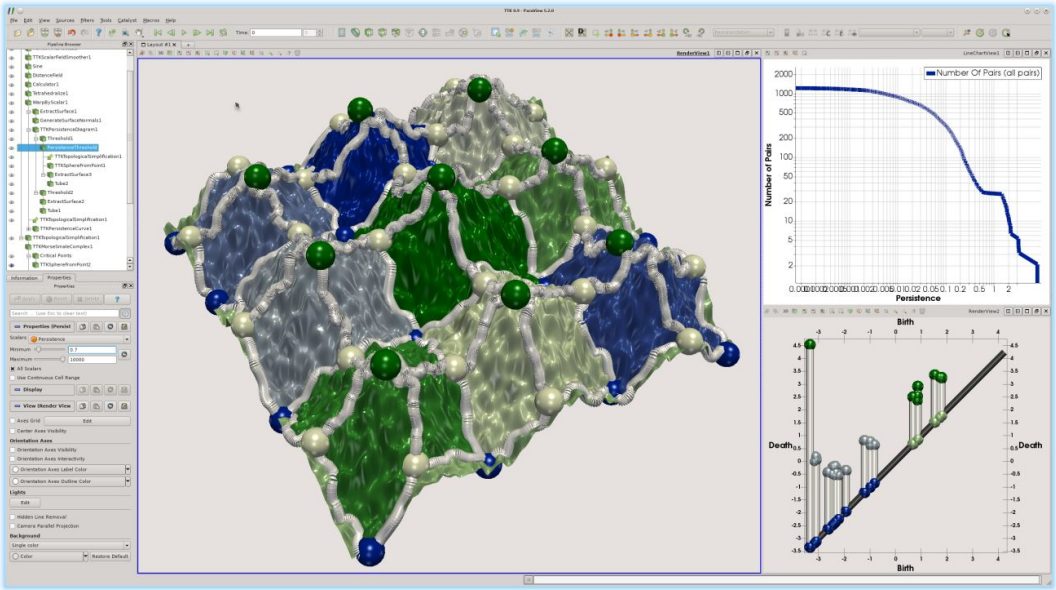
- **Developers**
  - Recommended
    - VTK-based APIs: C++ & Python
  - Advanced
    - Dependency-free plain C++ API

About ParaView

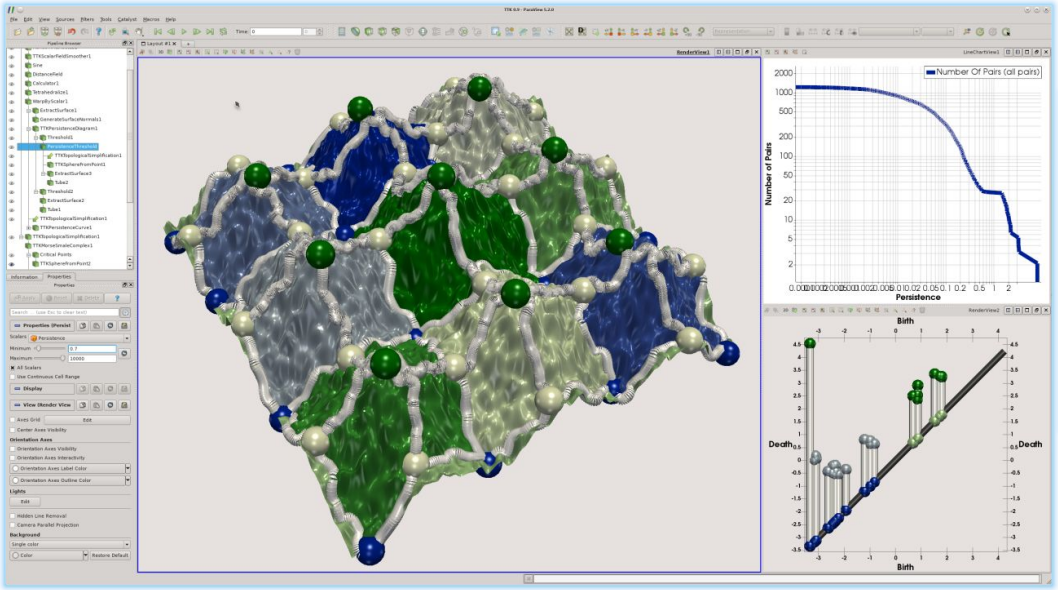
- **Interesting features**
  - Rich IO support
  - Modern rendering
    - OpenGL, OSPRay, Optix
  - Advanced user interface
  - “Visual” programming
    - Pipeline philosophy
  - Python scripting

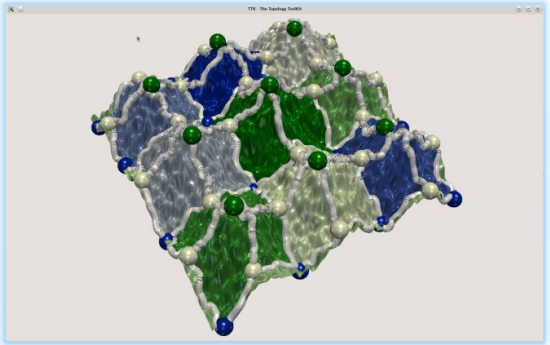


User experience

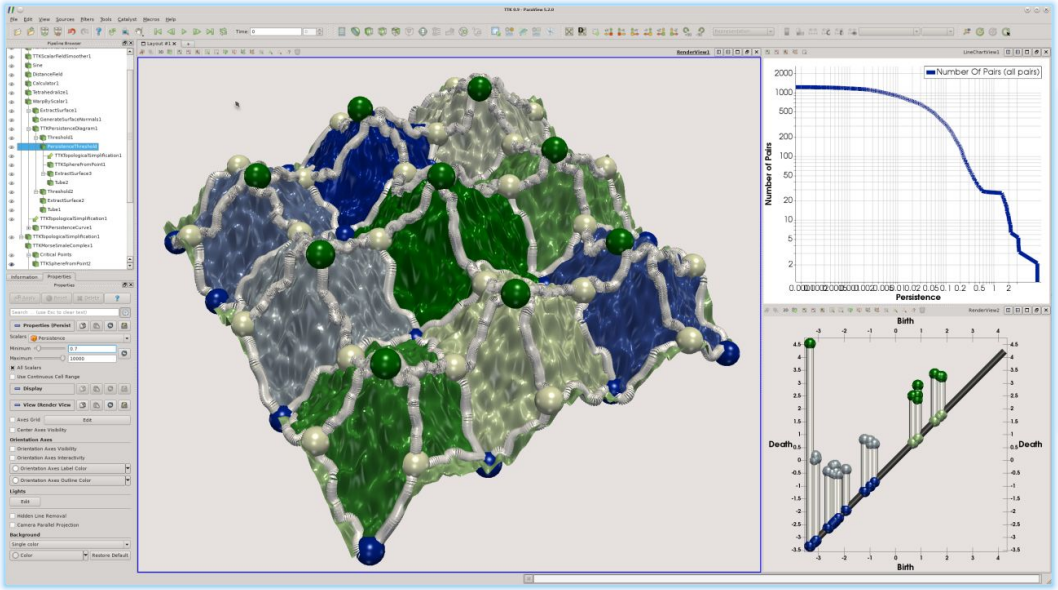


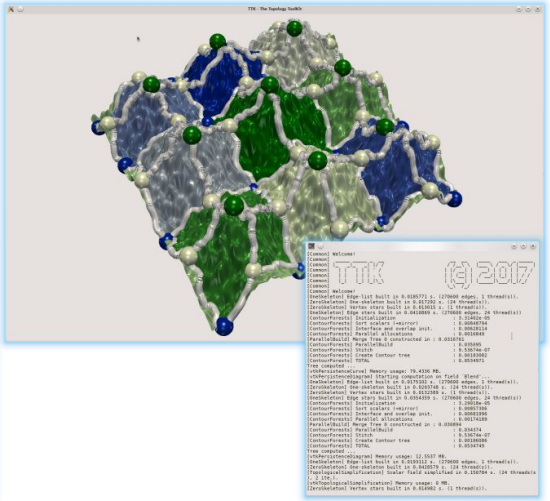
User experience



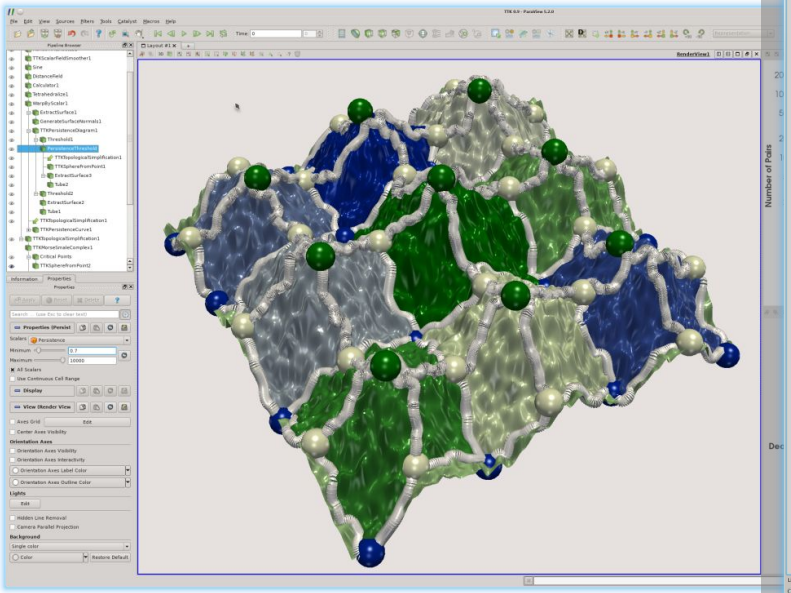


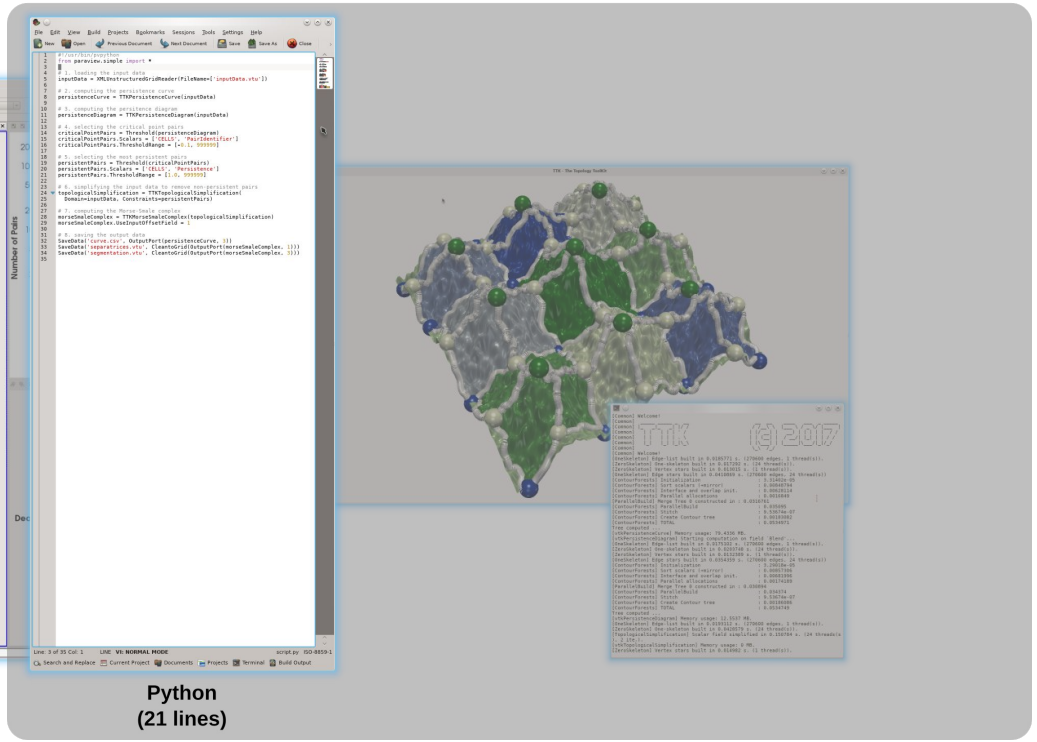
User experience





User experience

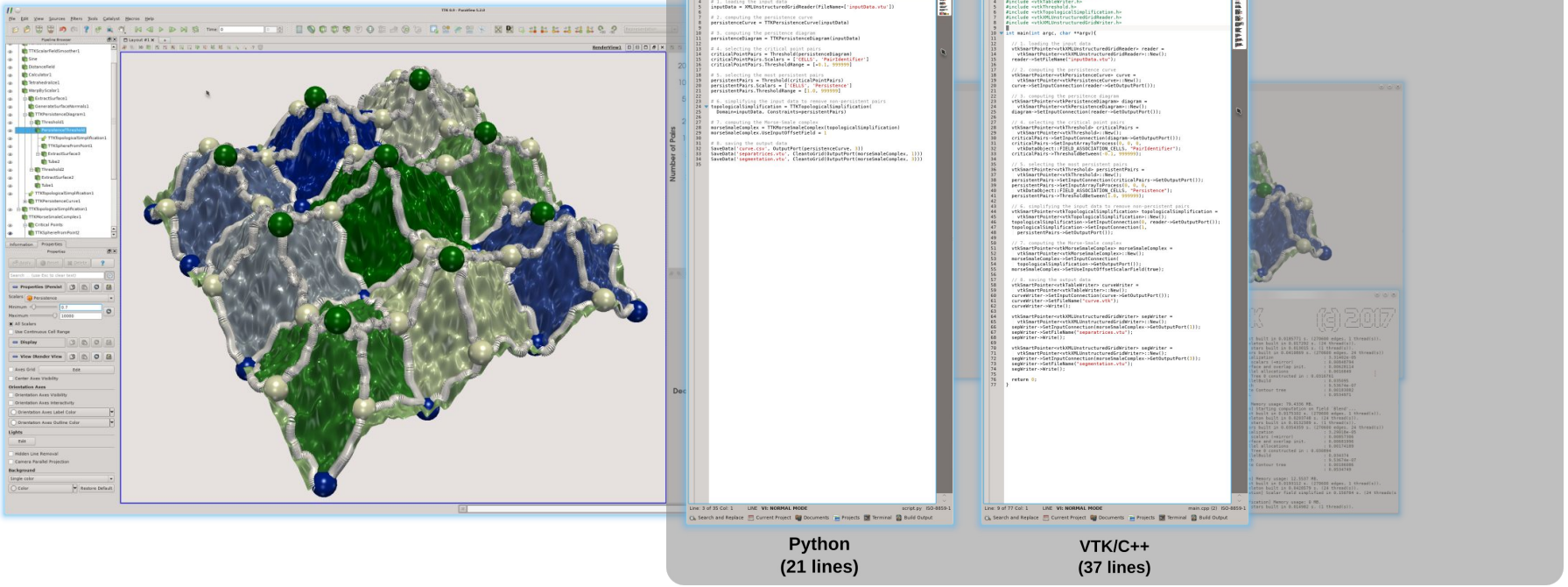




```
# Trigger Pipeline Execution
from paraview.simple import *

# 1: loading the input data
inputData = XMLUnstructuredGridReader(FileName=['inputData.vtu'])

# 2: computing the persistence curve
persistenceCurve = TTKPersistenceCurve(inputData)

# 3: computing the persistence diagram
persistenceDiagram = TTKPersistenceDiagram(inputData)

# 4: selecting the critical point pairs
criticalPointPairs = Threshold(persistenceDiagram)
criticalPointPairs.Scalars = ['CELLS', 'PairIdentifier']
criticalPointPairs.ThresholdRange = [-1, 999999]

# 5: selecting the most persistent pairs
persistentPairs = Threshold(criticalPointPairs)
persistentPairs.Scalars = ['CELLS', 'Persistence']
persistentPairs.ThresholdRange = [1.0, 999999]

# 6: simplifying the input data to remove non-persistent pairs
topologicalSimplification = TTKTopologicalSimplification(
  Domain=inputData, Constraints=persistentPairs)

# 7: computing the Morse Smale complex
morseSmaleComplex = TTKMorseSmaleComplex(topologicalSimplification)
morseSmaleComplex.UseInputOffsetField = 1

# 8: saving the output data
SaveData('curve.csv', OutputPort(persistenceCurve, 1))
SaveData('separatrices.vtu', CleanToGrid(OutputPort(morseSmaleComplex, 1)))
SaveData('segmentation.vtu', CleanToGrid(OutputPort(morseSmaleComplex, 3)))
```

Python
(21 lines)

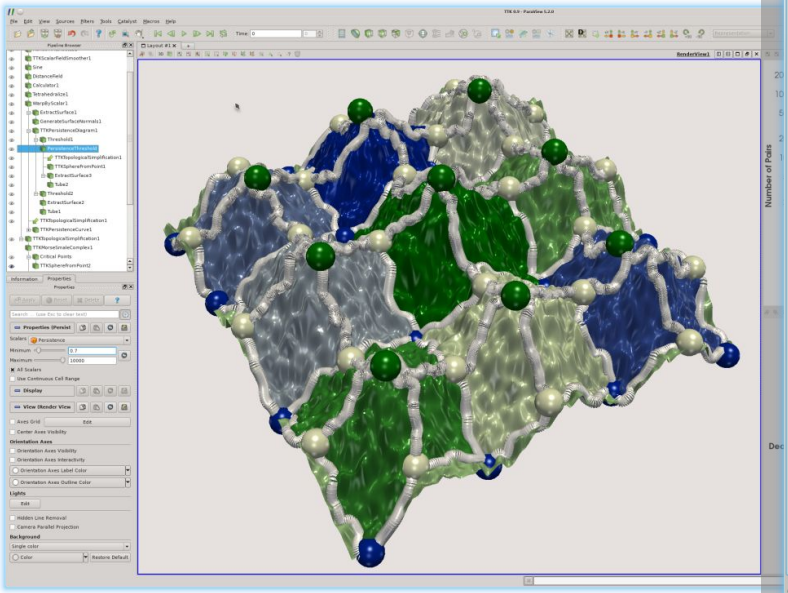
User experience

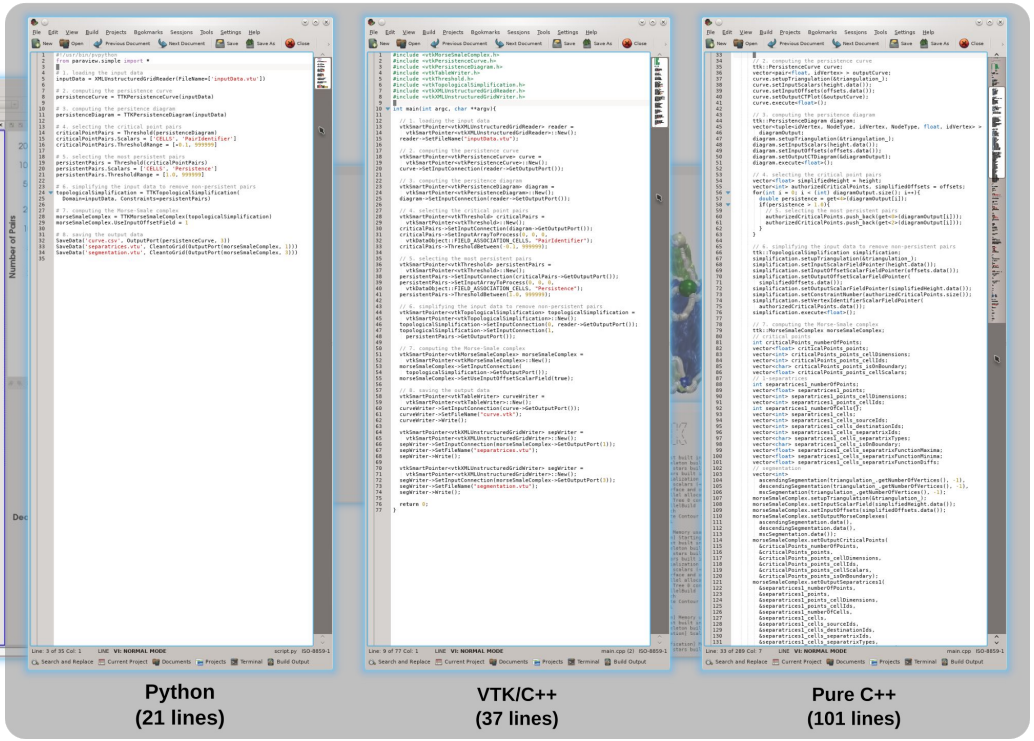


Python
(21 lines)

VTK/C++
(37 lines)

User experience



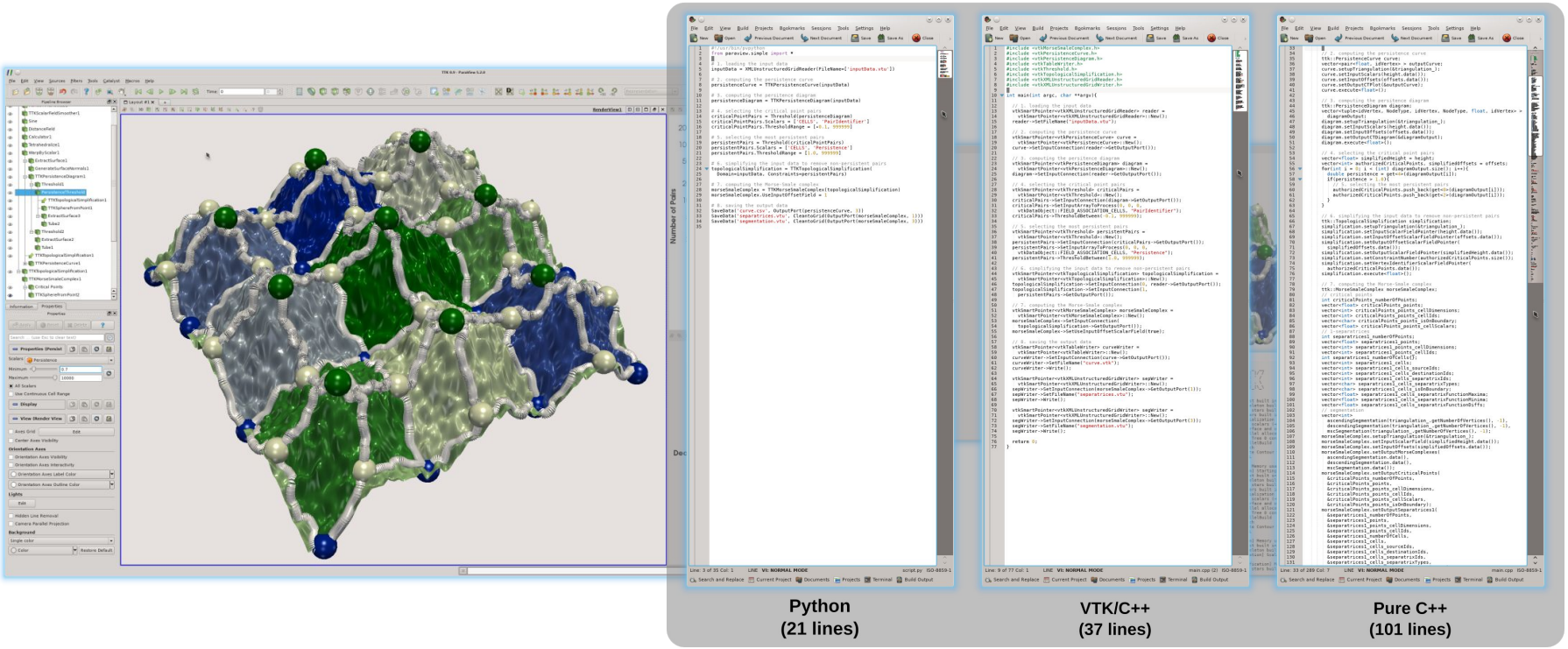


Python
(21 lines)

VTK/C++
(37 lines)

Pure C++
(101 lines)

User experience



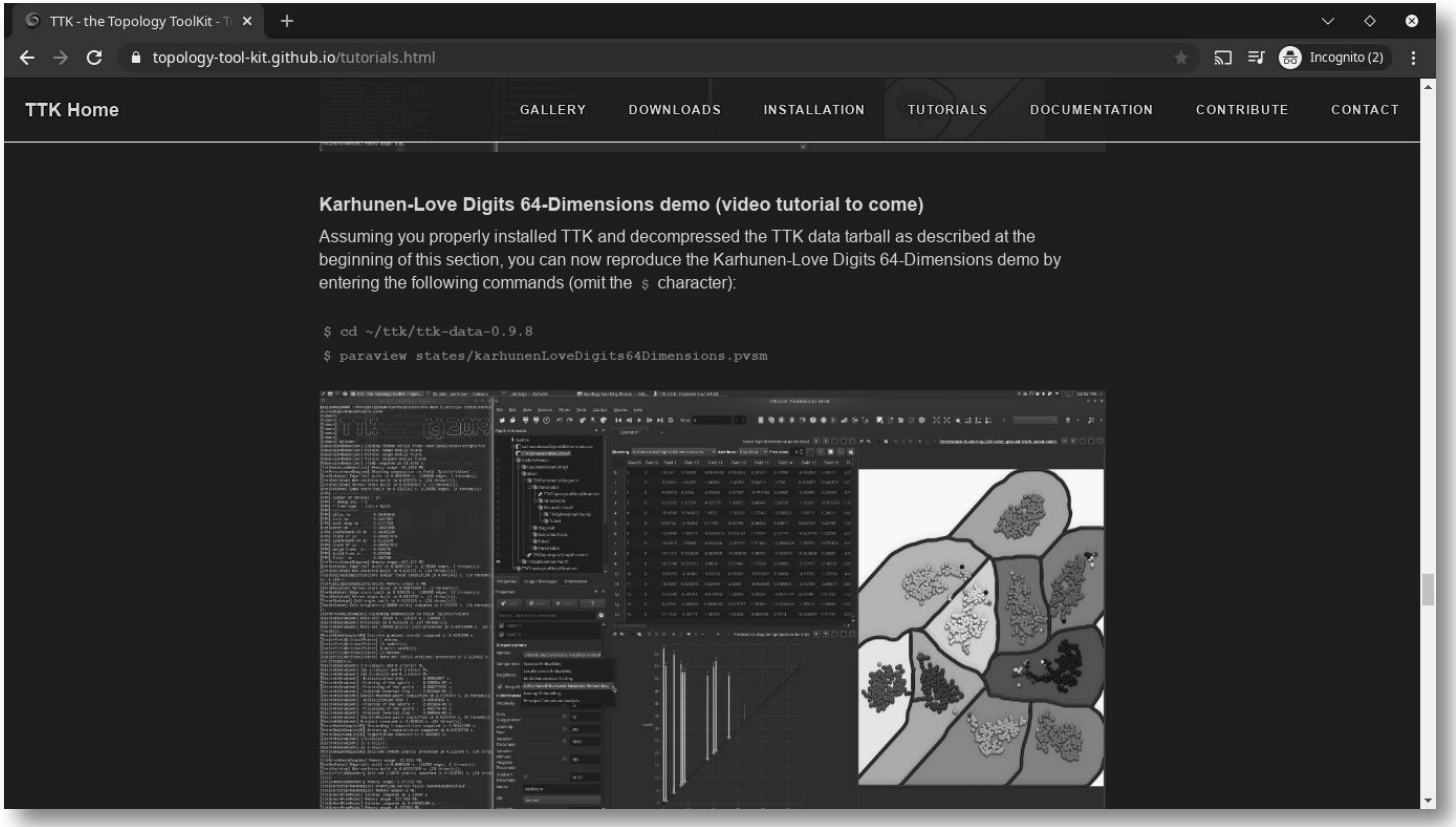
Python
(21 lines)

VTK/C++
(37 lines)

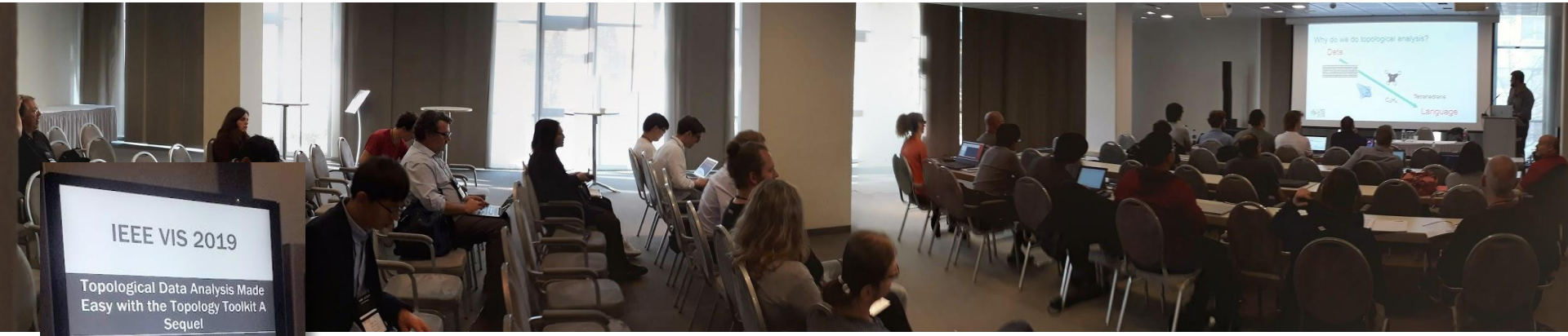
Pure C++
(101 lines)

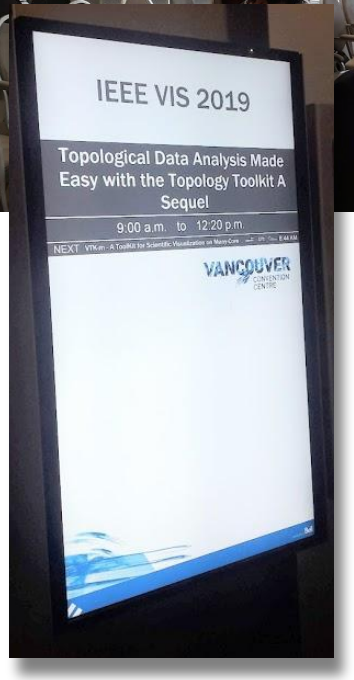
- https://github.com/topology-tool-kit/ttk/tree/dev/examples

TTK data & examples



IEEE VIS Tutorials 2018, 2019





- **Overview**
  - Presentations + hands-on
  - ~10 speakers, attendance: ~50
  - **https://topology-tool-kit.github.io/ieeeVisTutorial.html**
  - Slides, data, examples, pre-installed virtual machines, more

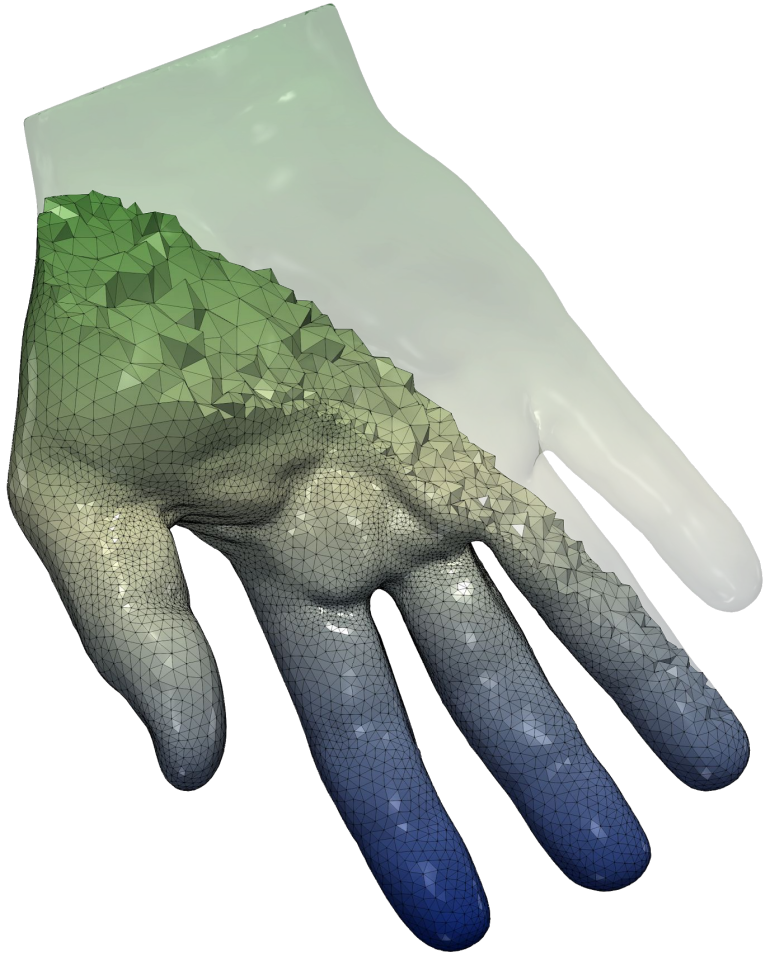
A (non-exhaustive) tour of TTK

Input data

- PL scalar field
  - $f : \mathcal{M} \to \mathbb{R}$

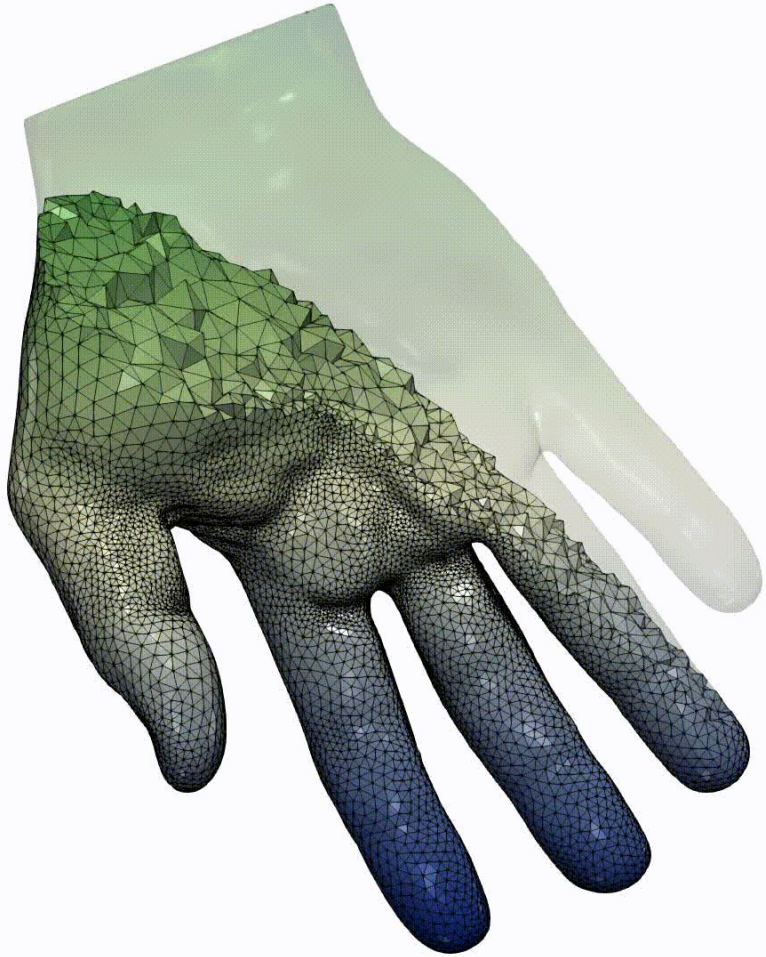
Input data

- **PL scalar field**
  - $f : \mathcal{M} \to \mathbb{R}$
  - $\mathcal{M}$: simplicial complex



Input data

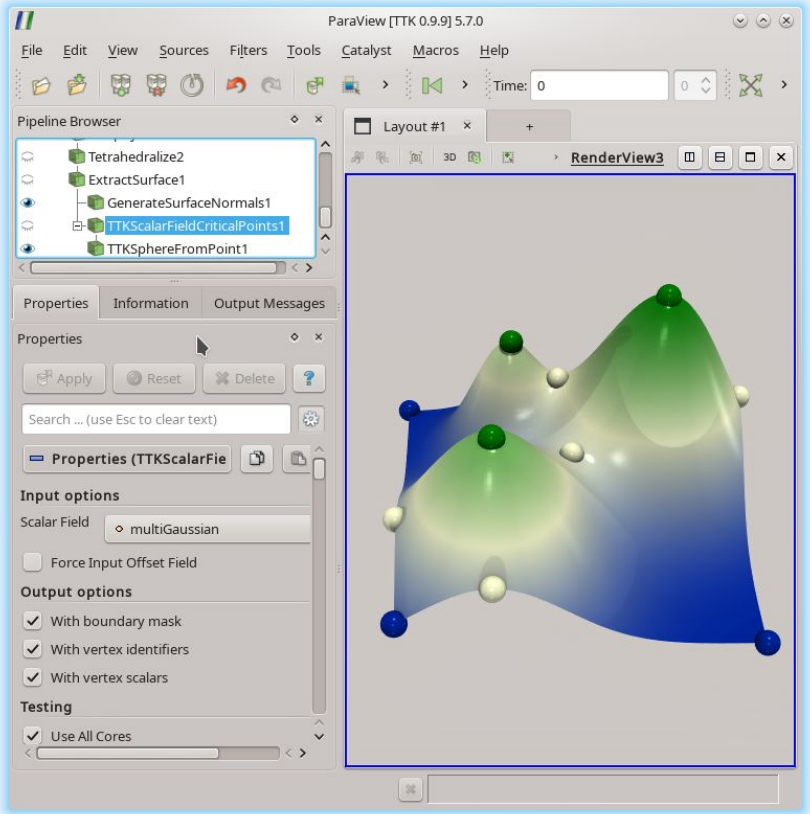
- **PL scalar field**
  - $f : \mathcal{M} \to \mathbb{R}$
  - $\mathcal{M}$: simplicial complex
    - Any mesh (`Tetrahedralize`)
      - Triangulated surface
      - Tetrahedral volume
    - Regular grids
      - 2D (pixel), 3D (voxel) images
      - Implicit triangulation
      - Periodicity support
    - Point clouds
      - Density estimates



Scalar data

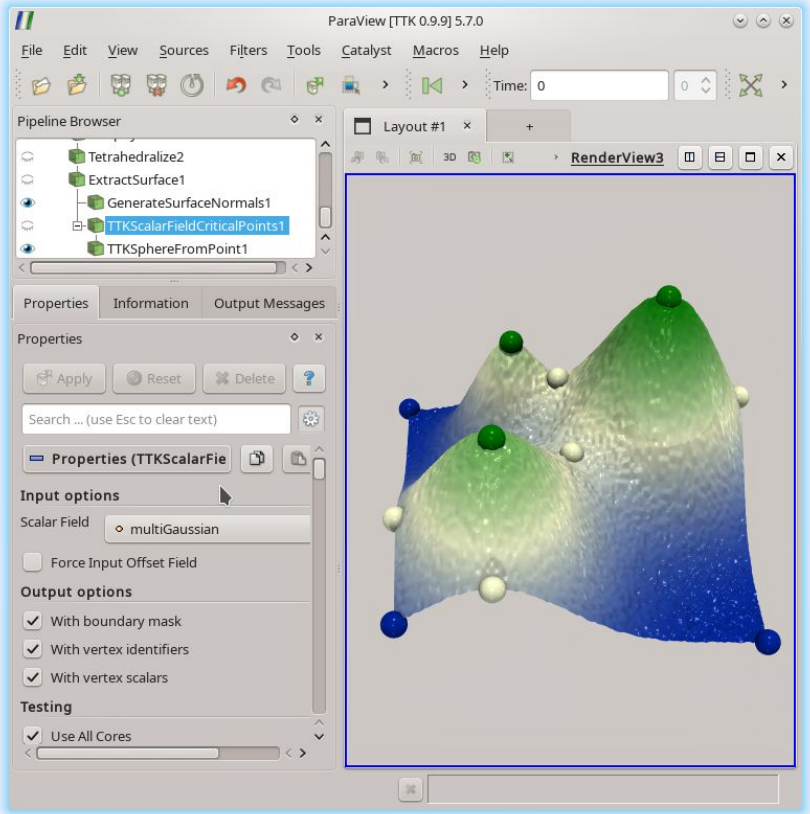
Critical points

- **Module**
  - `ScalarFieldCriticalPoints`
  - Extract the singularities and their index
- **Algorithm**
  - Banchoff, J. Diff. Geom. 1967
  - Linear time, trivially parallel
- **Output**
  - Points with index and vertex IDs



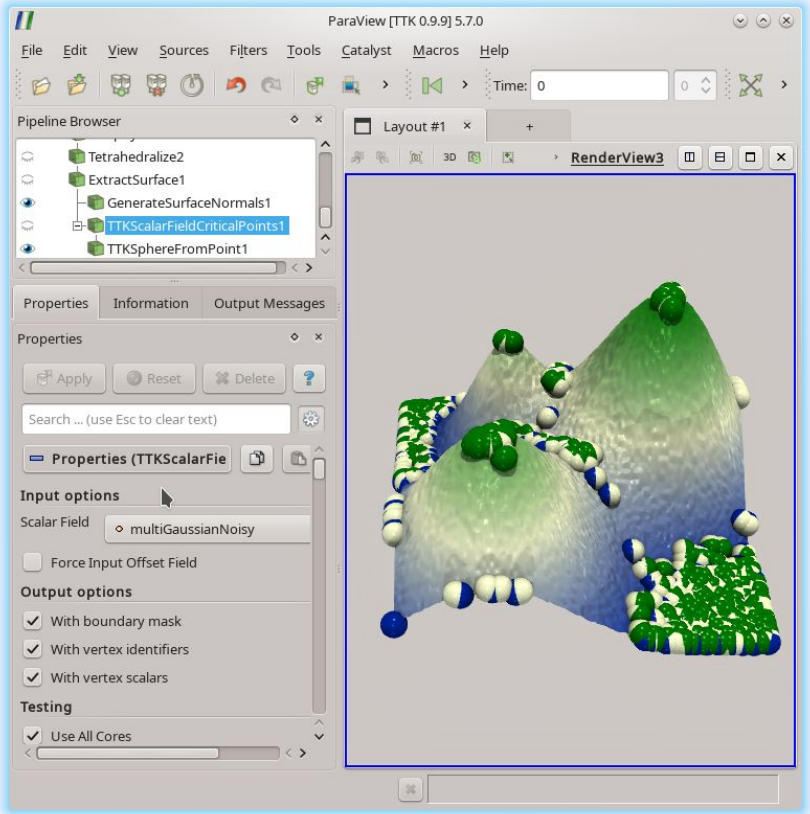
Critical points

- **Module**
  - `ScalarFieldCriticalPoints`
  - Extract the singularities and their index
- **Algorithm**
  - Banchoff, J. Diff. Geom. 1967
  - Linear time, trivially parallel
- **Output**
  - Points with index and vertex IDs



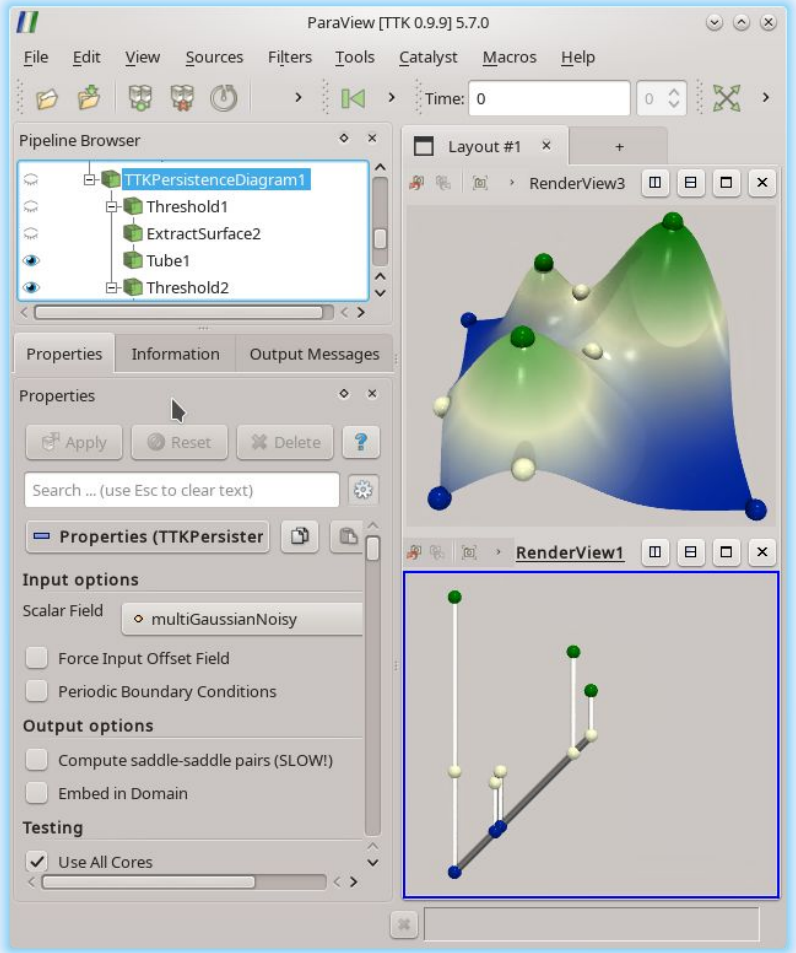
Critical points

- **Module**
  - `ScalarFieldCriticalPoints`
  - Extract the singularities and their index
- **Algorithm**
  - Banchoff, J. Diff. Geom. 1967
  - Linear time, trivially parallel
- **Output**
  - Points with index and vertex IDs



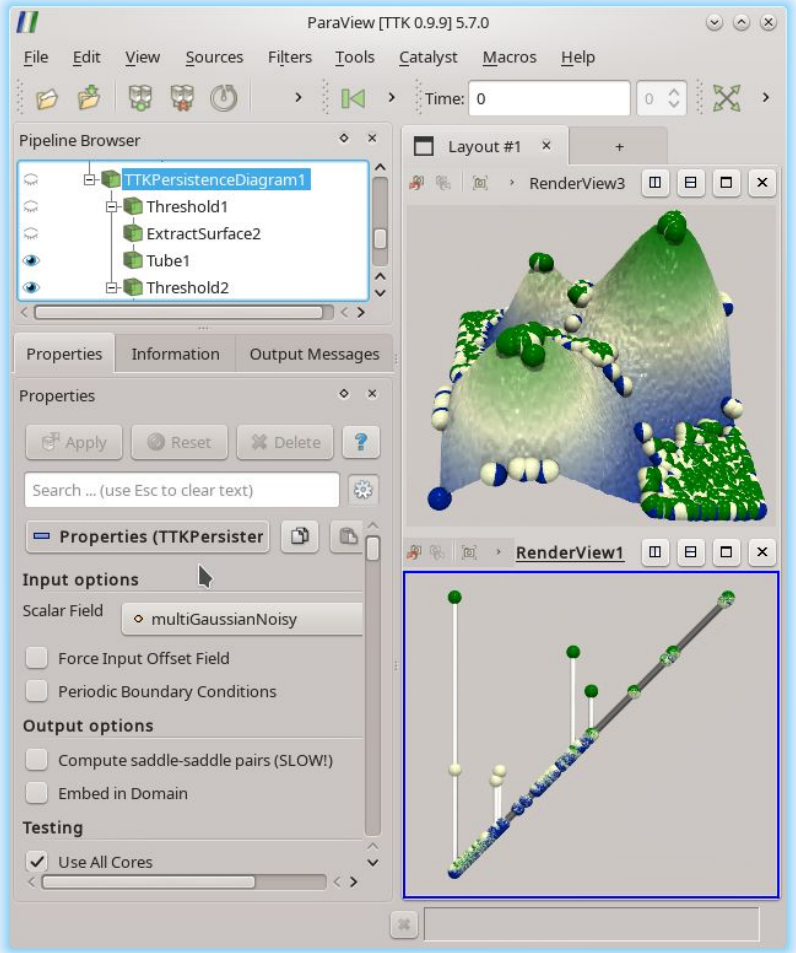
Persistence diagrams

- **Module**
  - **`PersistenceDiagram`**
  - **`PersistenceCurve`**
  - Specialized for dimensions 0, 1, 2
- **Algorithms**
  - Extremum/saddle pairs
    - Gueunet et al. IEEE TPDS 2019
    - Linearithmic time, efficient parallelization
  - Saddle/saddle pairs
    - Saddle connector reversal
- **Output**
  - One edge per pair (indices & vertex IDs)
  - One curve per pair type



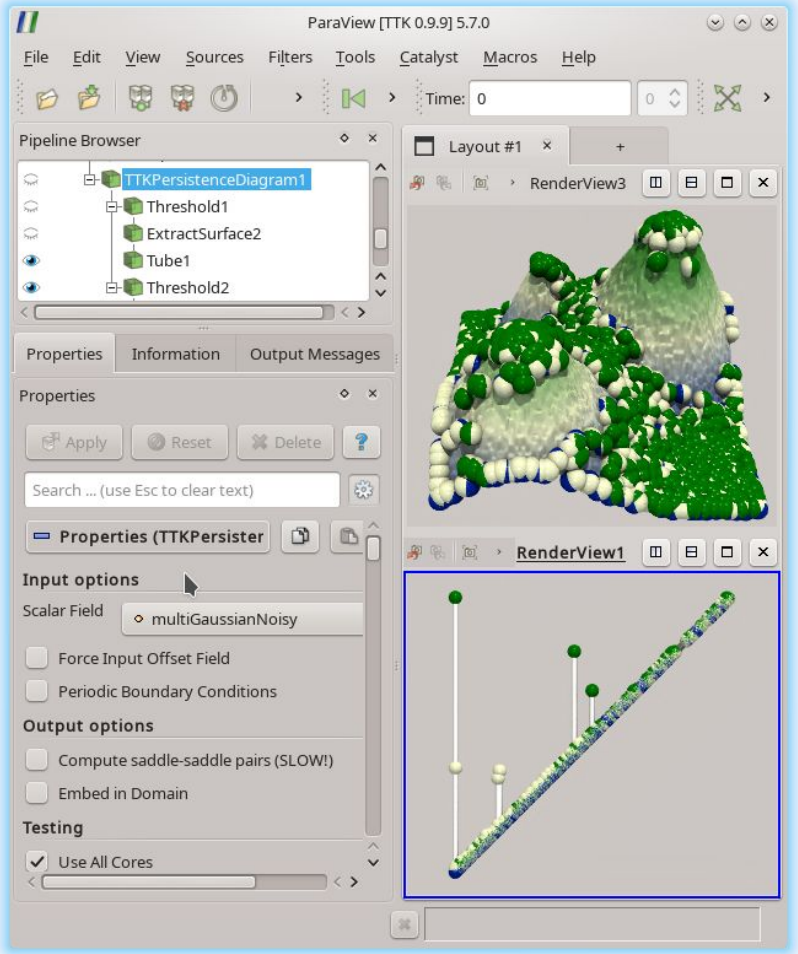
Persistence diagrams

- **Module**
  - `PersistenceDiagram`
  - `PersistenceCurve`
  - Specialized for dimensions 0, 1, 2
- **Algorithms**
  - Extremum/saddle pairs
    - Gueunet et al. IEEE TPDS 2019
    - Linearithmic time, efficient parallelization
  - Saddle/saddle pairs
    - Saddle connector reversal
- **Output**
  - One edge per pair (indices & vertex IDs)
  - One curve per pair type



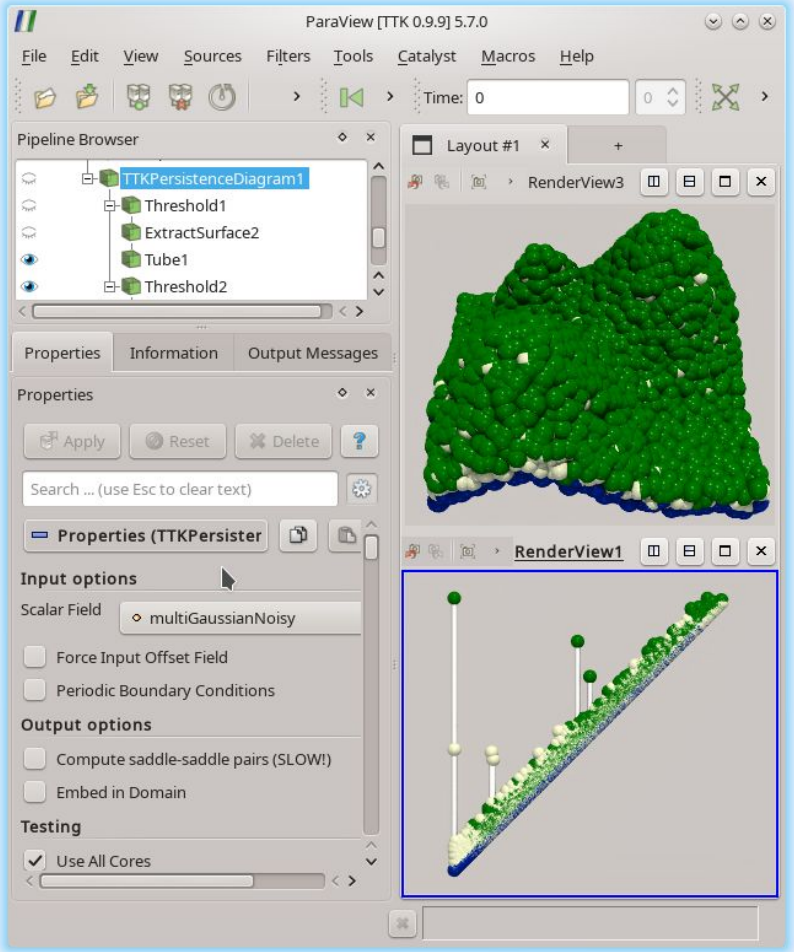
Persistence diagrams

- **Module**
  - **`PersistenceDiagram`**
  - **`PersistenceCurve`**
  - Specialized for dimensions 0, 1, 2
- **Algorithms**
  - Extremum/saddle pairs
    - Gueunet et al. IEEE TPDS 2019
    - Linearithmic time, efficient parallelization
  - Saddle/saddle pairs
    - Saddle connector reversal
- **Output**
  - One edge per pair (indices & vertex IDs)
  - One curve per pair type



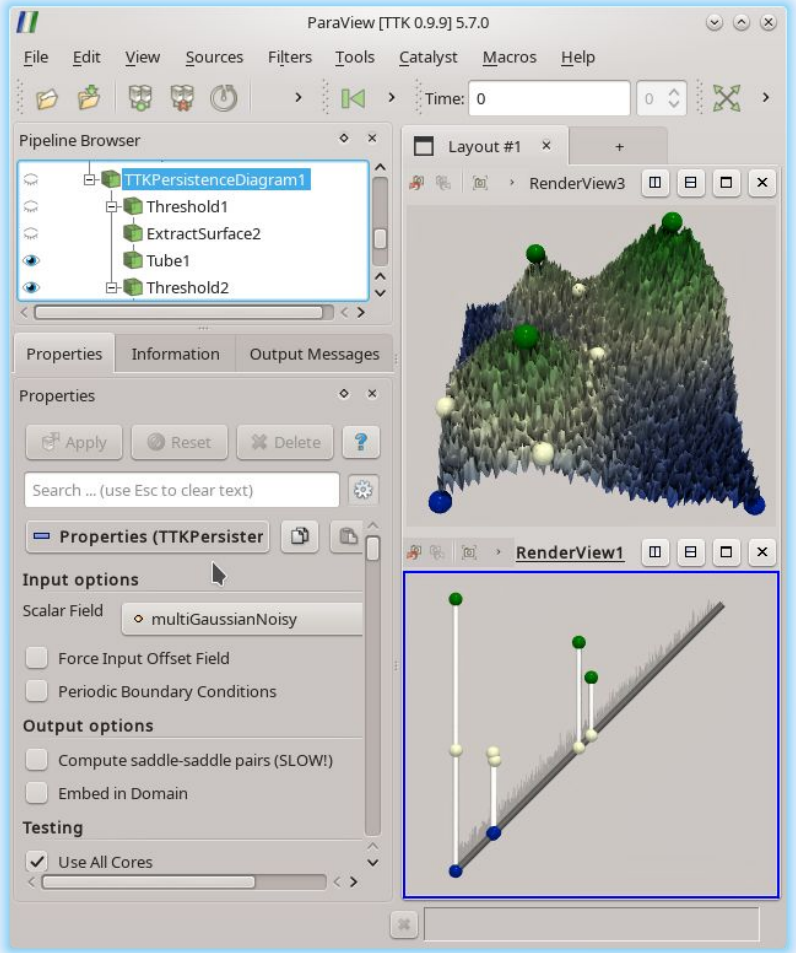
Persistence diagrams

- **Module**
  - `PersistenceDiagram`
  - `PersistenceCurve`
  - Specialized for dimensions 0, 1, 2
- **Algorithms**
  - Extremum/saddle pairs
    - Gueunet et al. IEEE TPDS 2019
    - Linearithmic time, efficient parallelization
  - Saddle/saddle pairs
    - Saddle connector reversal
- **Output**
  - One edge per pair (indices & vertex IDs)
  - One curve per pair type



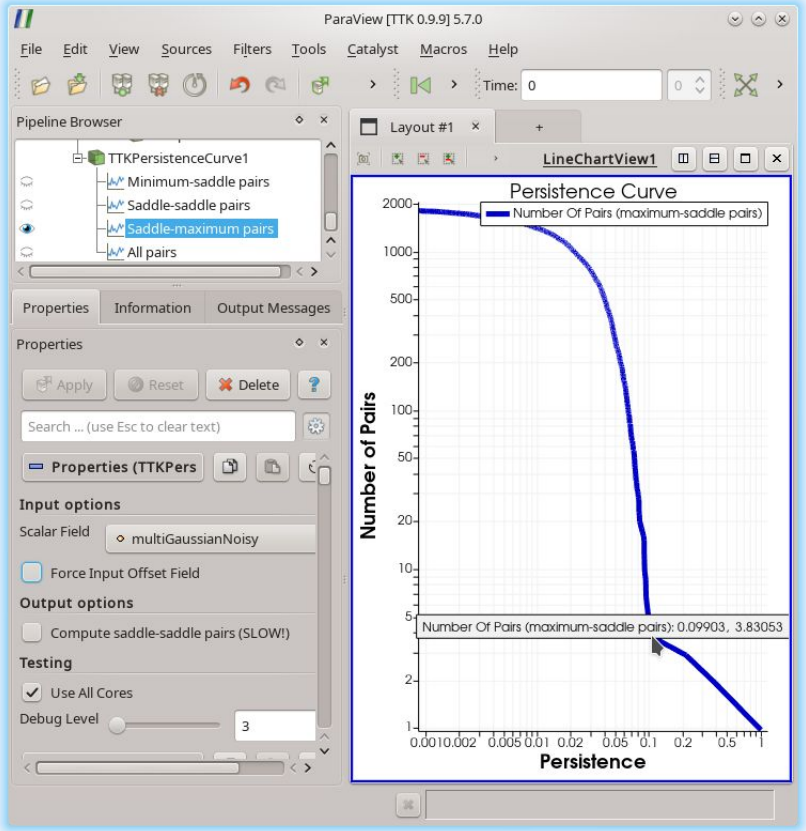
Persistence diagrams

- **Module**
  - **`PersistenceDiagram`**
  - **`PersistenceCurve`**
  - Specialized for dimensions 0, 1, 2
- **Algorithms**
  - Extremum/saddle pairs
    - Gueunet et al. IEEE TPDS 2019
    - Linearithmic time, efficient parallelization
  - Saddle/saddle pairs
    - Saddle connector reversal
- **Output**
  - One edge per pair (indices & vertex IDs)
  - One curve per pair type



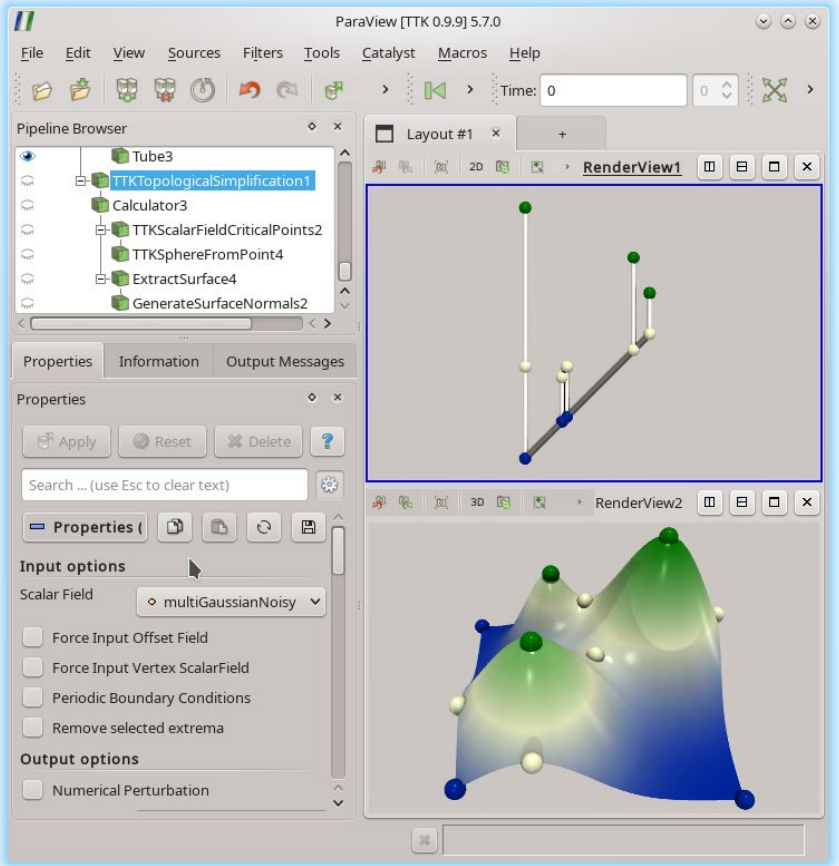
Persistence diagrams

- **Module**
  - `PersistenceDiagram`
  - `PersistenceCurve`
  - Specialized for dimensions 0, 1, 2
- **Algorithms**
  - Extremum/saddle pairs
    - Gueunet et al. IEEE TPDS 2019
    - Linearithmic time, efficient parallelization
  - Saddle/saddle pairs
    - Saddle connector reversal
- **Output**
  - One edge per pair (indices & vertex IDs)
  - One curve per pair type



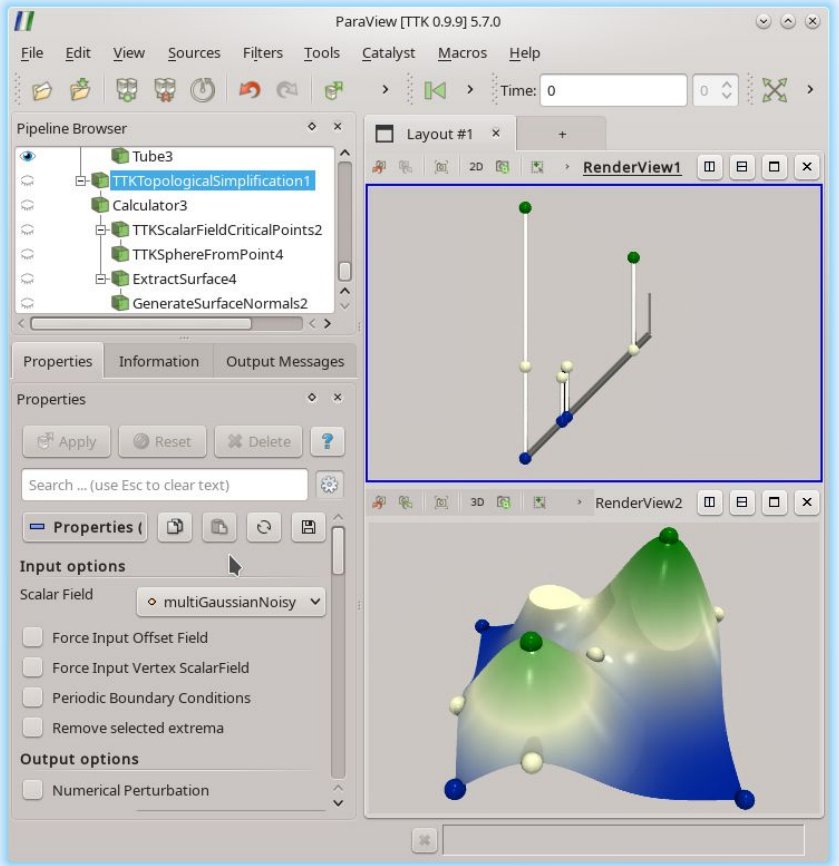
Topological data simplification

- **Module**
  - `TopologicalSimplification`
  - Extremum removal
  - Default multiscale mechanism
- **Algorithm**
  - Tierny & Pascucci IEEE TVCG 2012
  - Linearithmic time
- **Output**
  - “Flattened” data



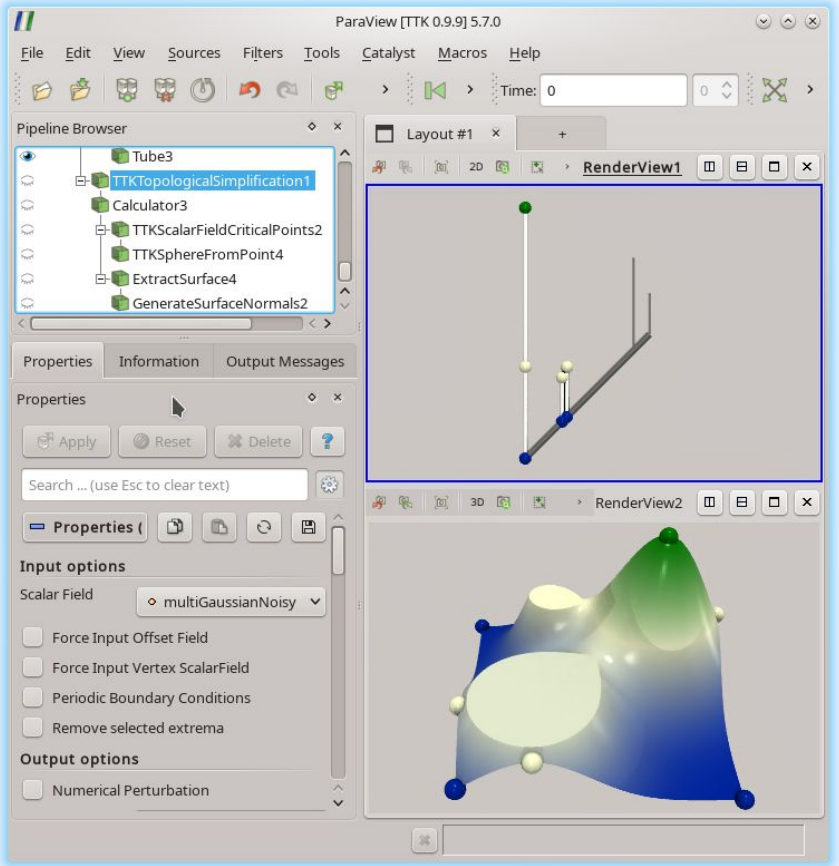
Topological data simplification

- **Module**
  - `TopologicalSimplification`
  - Extremum removal
  - Default multiscale mechanism
- **Algorithm**
  - Tierny & Pascucci IEEE TVCG 2012
  - Linearithmic time
- **Output**
  - “Flattened” data



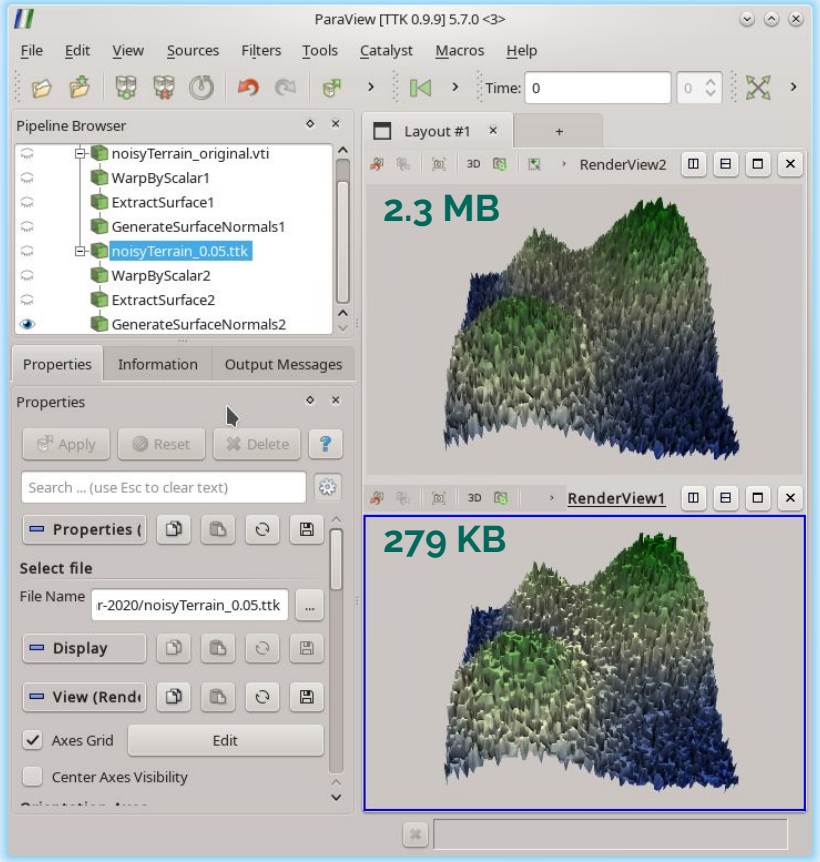
Topological data simplification

- **Module**
  - `TopologicalSimplification`
  - Extremum removal
  - Default multiscale mechanism
- **Algorithm**
  - Tierny & Pascucci IEEE TVCG 2012
  - Linearithmic time
- **Output**
  - “Flattened” data



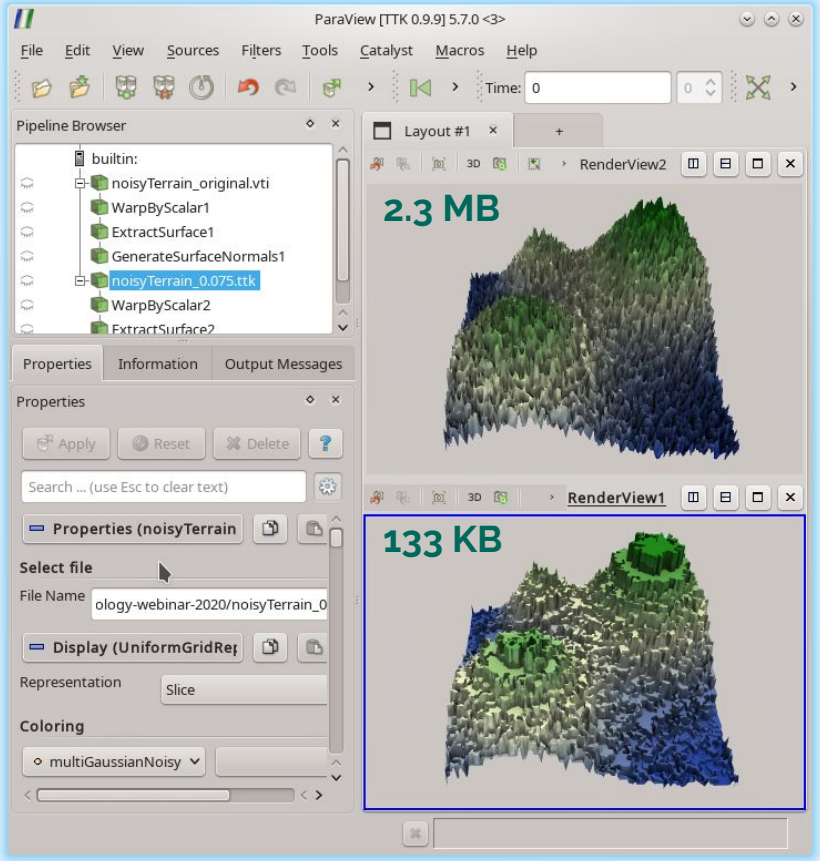
Topology aware compression

- **Module**
  - `TopologicalCompression (IO)`
  - Preserves the persistence diagram
    - Dimension 0 or (d-1)
- **Algorithm**
  - Soler et al. PacificViz 2018
  - Linearithmic time
- **Output**
  - Compressed image data
  - Geometry improvement with ZFP
    - Lindstrom IEEE TVCG 2014



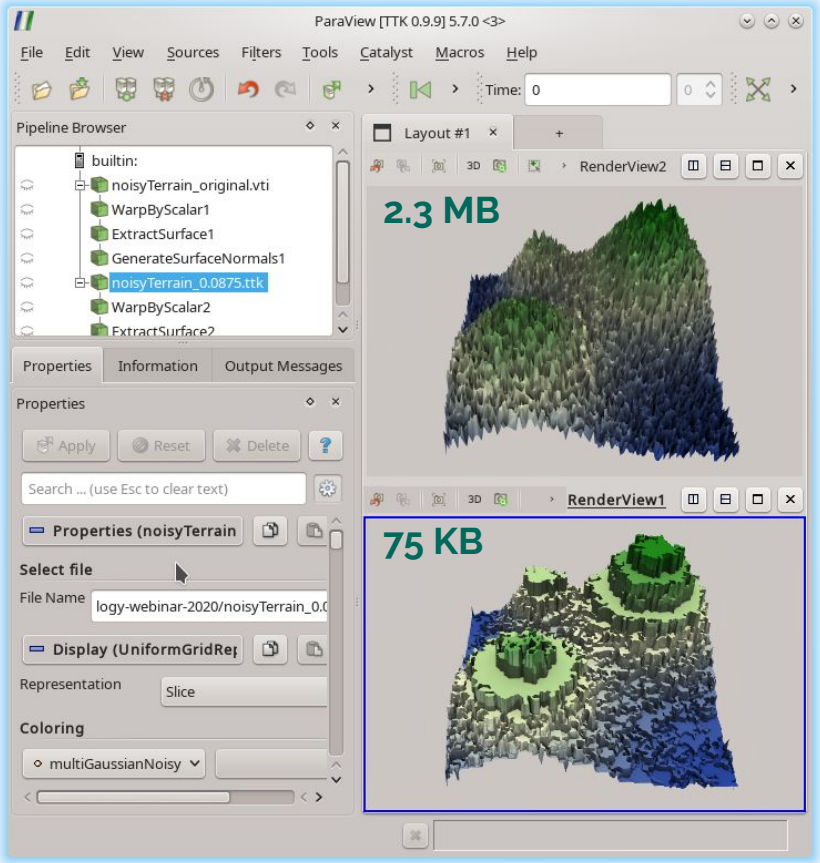
Topology aware compression

- **Module**
  - `TopologicalCompression (IO)`
  - Preserves the persistence diagram
    - Dimension 0 or (d-1)
- **Algorithm**
  - Soler et al. PacificViz 2018
  - Linearithmic time
- **Output**
  - Compressed image data
  - Geometry improvement with ZFP
    - Lindstrom IEEE TVCG 2014



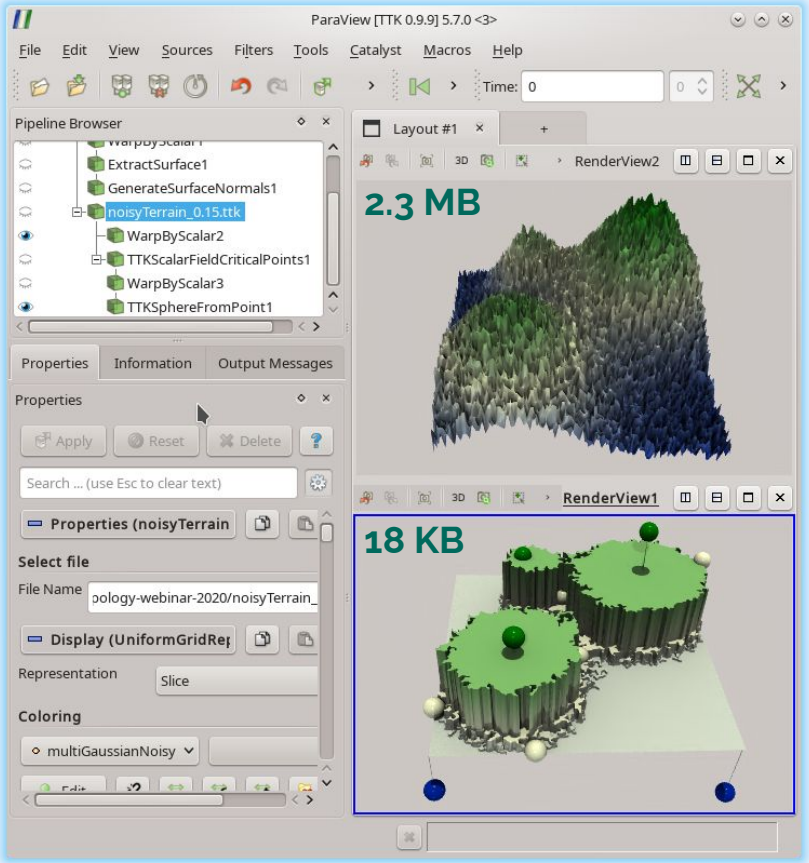
Topology aware compression

- **Module**
  - `TopologicalCompression` (IO)
  - Preserves the persistence diagram
    - Dimension 0 or (d-1)
- **Algorithm**
  - Soler et al. PacificViz 2018
  - Linearithmic time
- **Output**
  - Compressed image data
  - Geometry improvement with ZFP
    - Lindstrom IEEE TVCG 2014



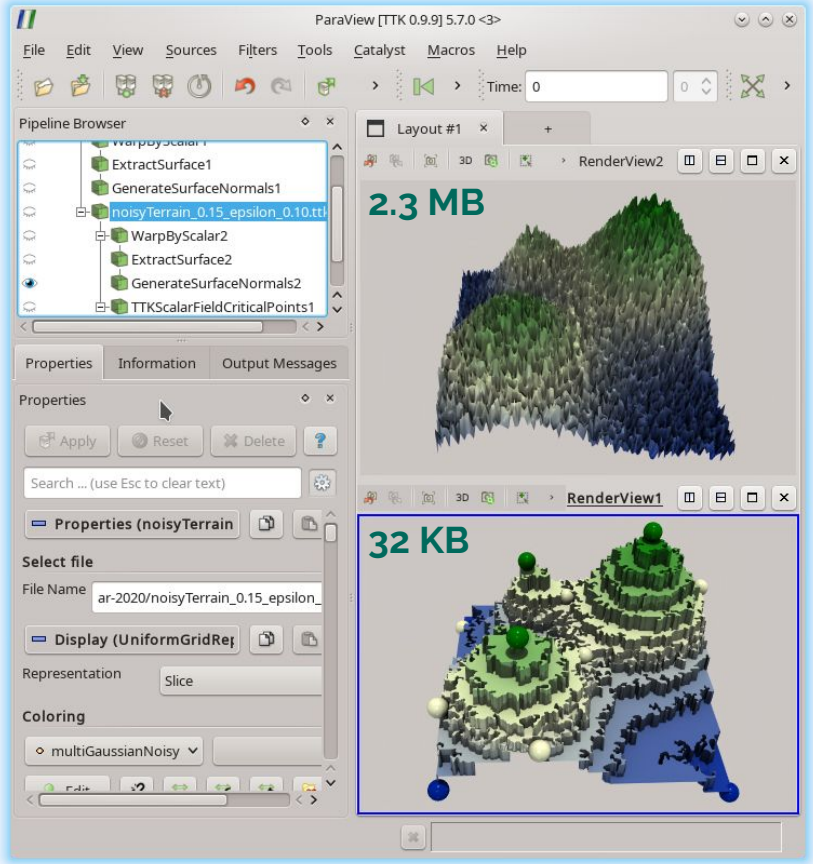
Topology aware compression

- **Module**
  - `TopologicalCompression` (IO)
  - Preserves the persistence diagram
    - Dimension 0 or (d-1)
- **Algorithm**
  - Soler et al. PacificViz 2018
  - Linearithmic time
- **Output**
  - Compressed image data
  - Geometry improvement with ZFP
    - Lindstrom IEEE TVCG 2014



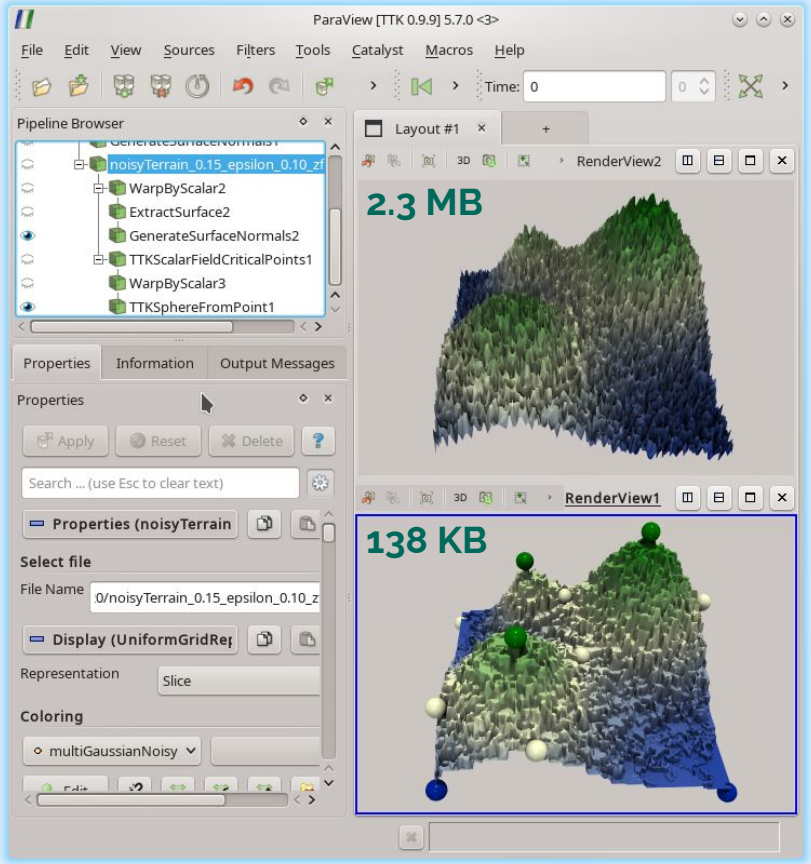
Topology aware compression

- **Module**
  - `TopologicalCompression (IO)`
  - Preserves the persistence diagram
    - Dimension 0 or (d-1)
- **Algorithm**
  - Soler et al. PacificViz 2018
  - Linearithmic time
- **Output**
  - Compressed image data
  - Geometry improvement with ZFP
    - Lindstrom IEEE TVCG 2014



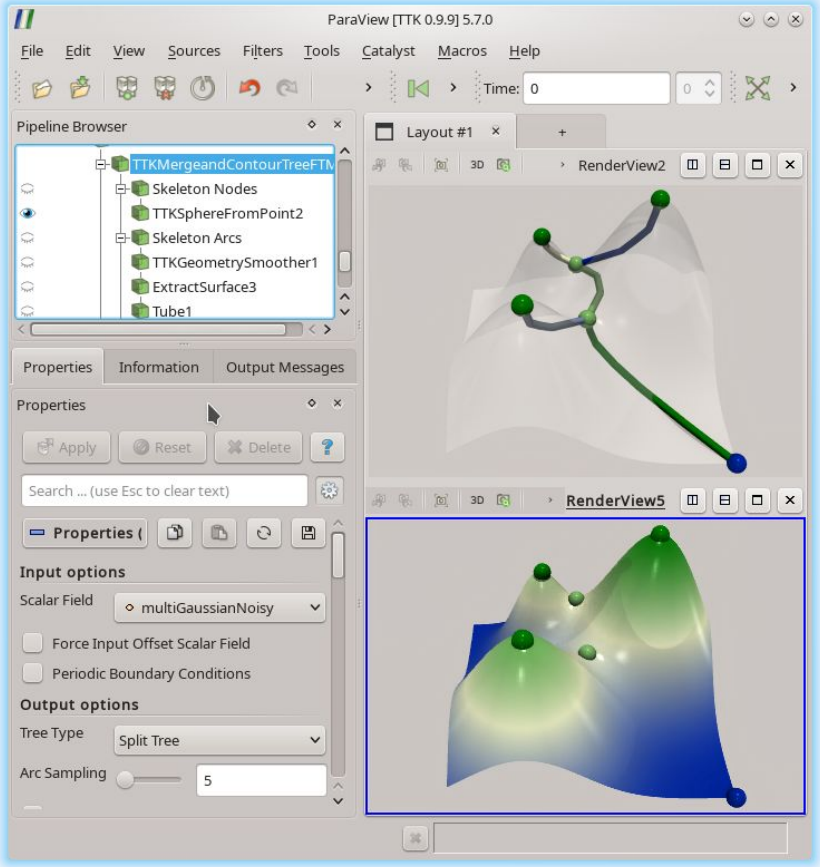
Topology aware compression

- **Module**
  - `TopologicalCompression` (IO)
  - Preserves the persistence diagram
    - Dimension 0 or (d-1)
- **Algorithm**
  - Soler et al. PacificViz 2018
  - Linearithmic time
- **Output**
  - Compressed image data
  - Geometry improvement with ZFP
    - Lindstrom IEEE TVCG 2014



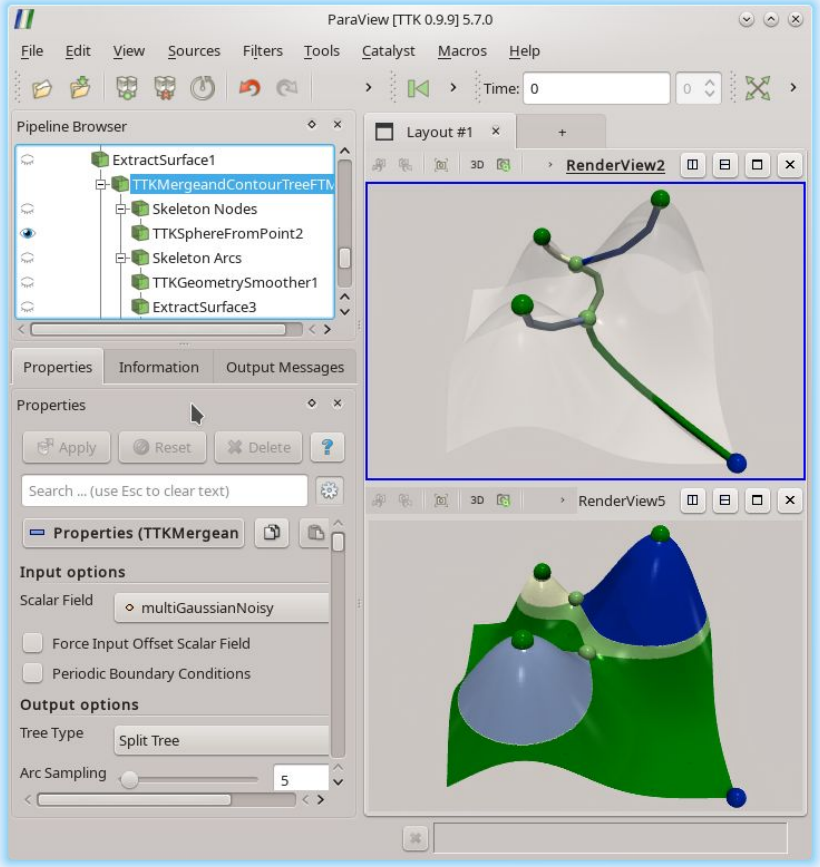
Merge & contour trees

- **Module**
  - **`FTMTree`**
  - Skeleton extraction
  - Level-set based segmentation
- **Algorithm**
  - Gueunet et al. IEEE TPDS 2019
  - Linearithmic time, efficient parallelization
- **Output**
  - Nodes of the trees
  - Arcs
  - Data segmentation



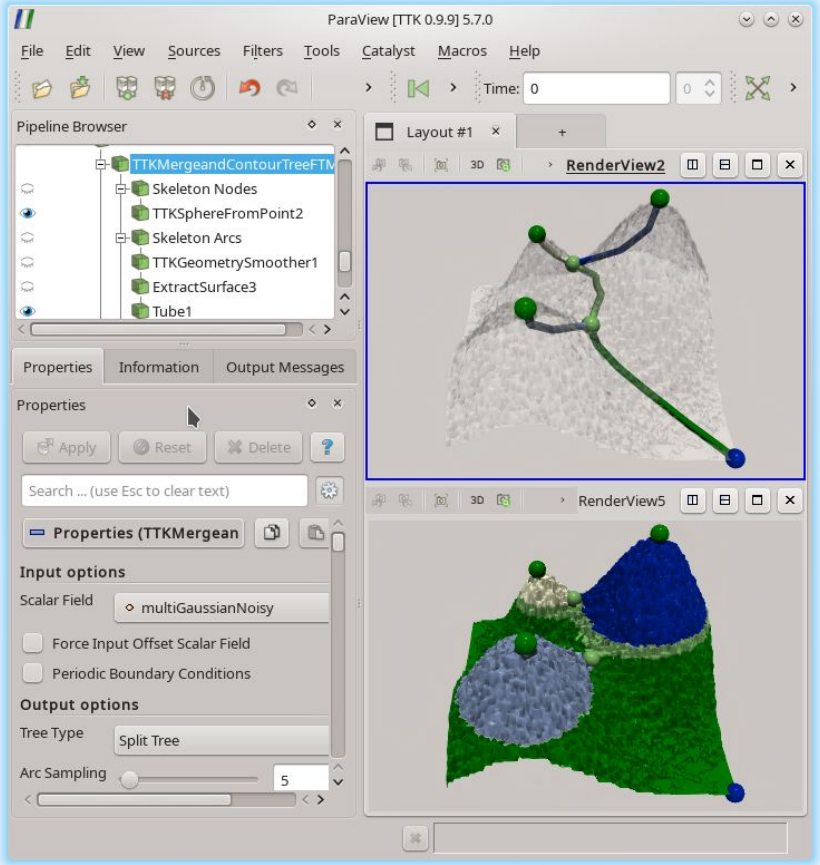
Merge & contour trees

- **Module**
  - **`FTMTree`**
  - Skeleton extraction
  - Level-set based segmentation
- **Algorithm**
  - Gueunet et al. IEEE TPDS 2019
  - Linearithmic time, efficient parallelization
- **Output**
  - Nodes of the trees
  - Arcs
  - Data segmentation

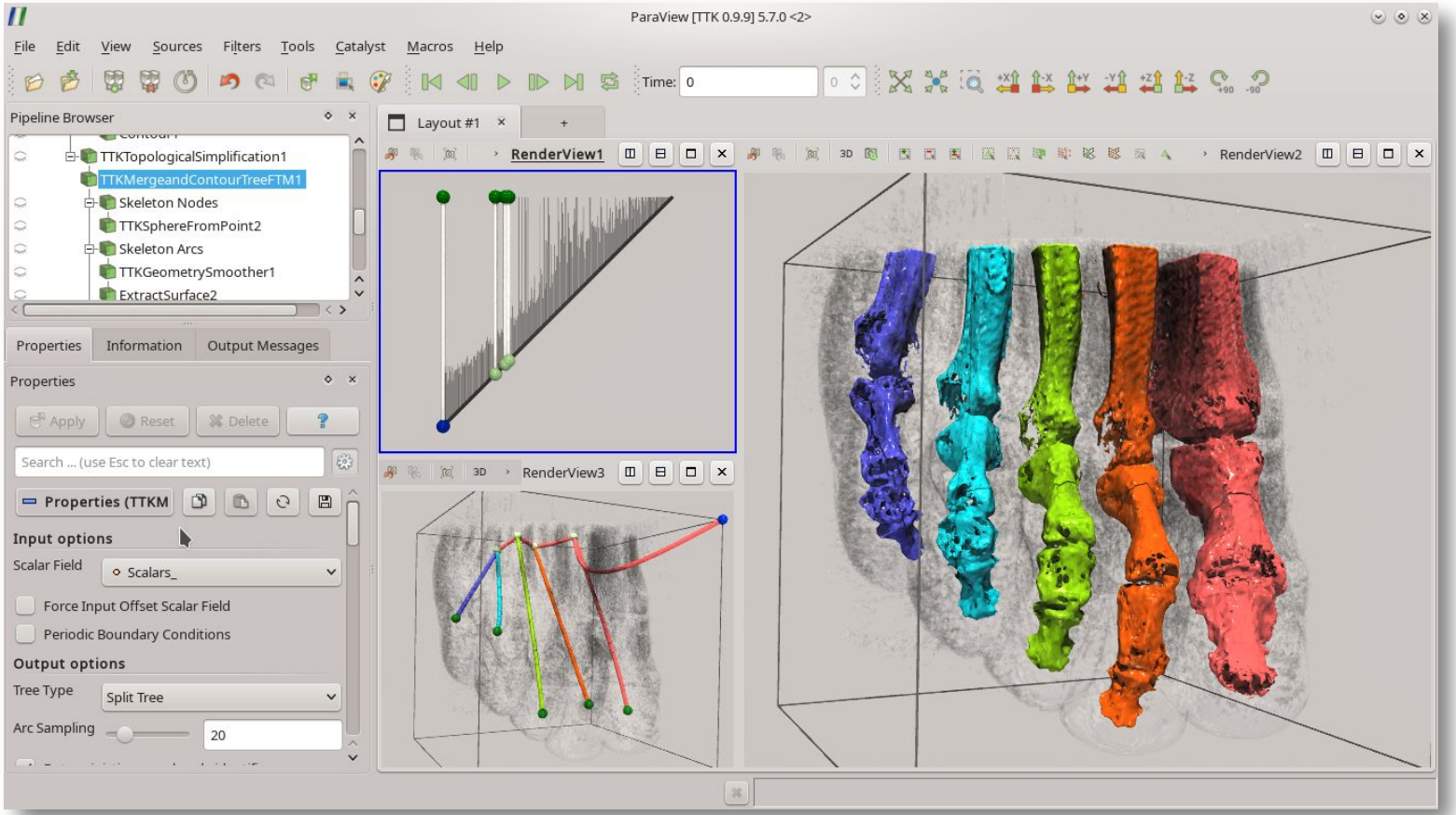


Merge & contour trees

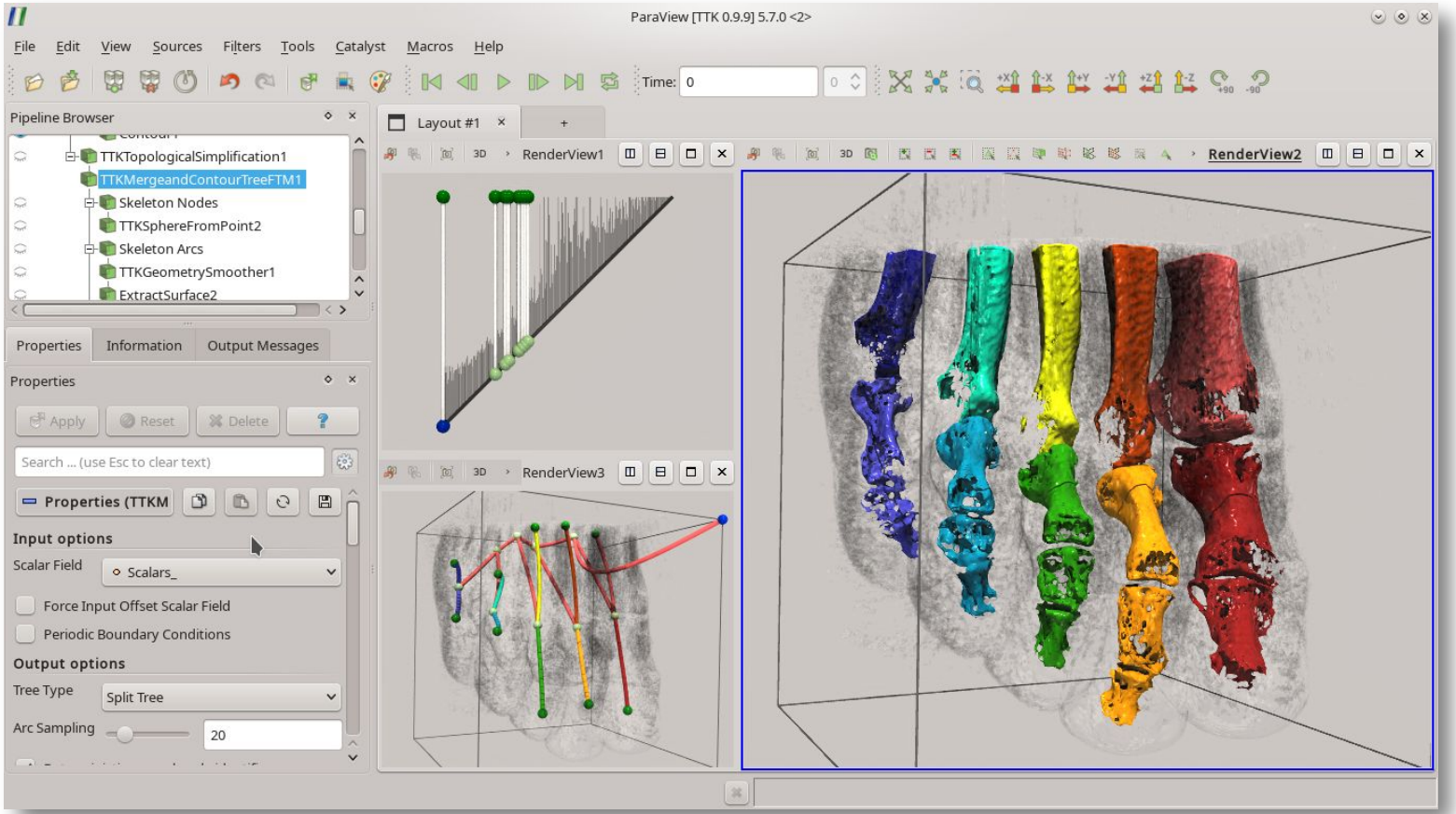
- **Module**
  - **`FTMTree`**
  - Skeleton extraction
  - Level-set based segmentation
- **Algorithm**
  - Gueunet et al. IEEE TPDS 2019
  - Linearithmic time, efficient parallelization
- **Output**
  - Nodes of the trees
  - Arcs
  - Data segmentation



Application to biomedical imaging

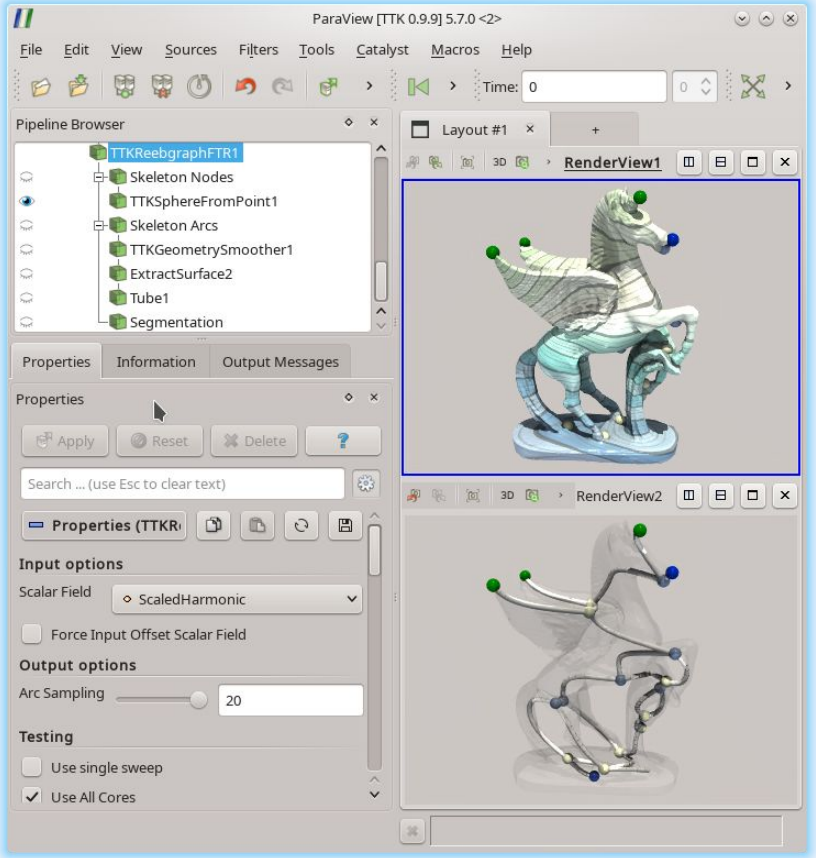


Application to biomedical imaging



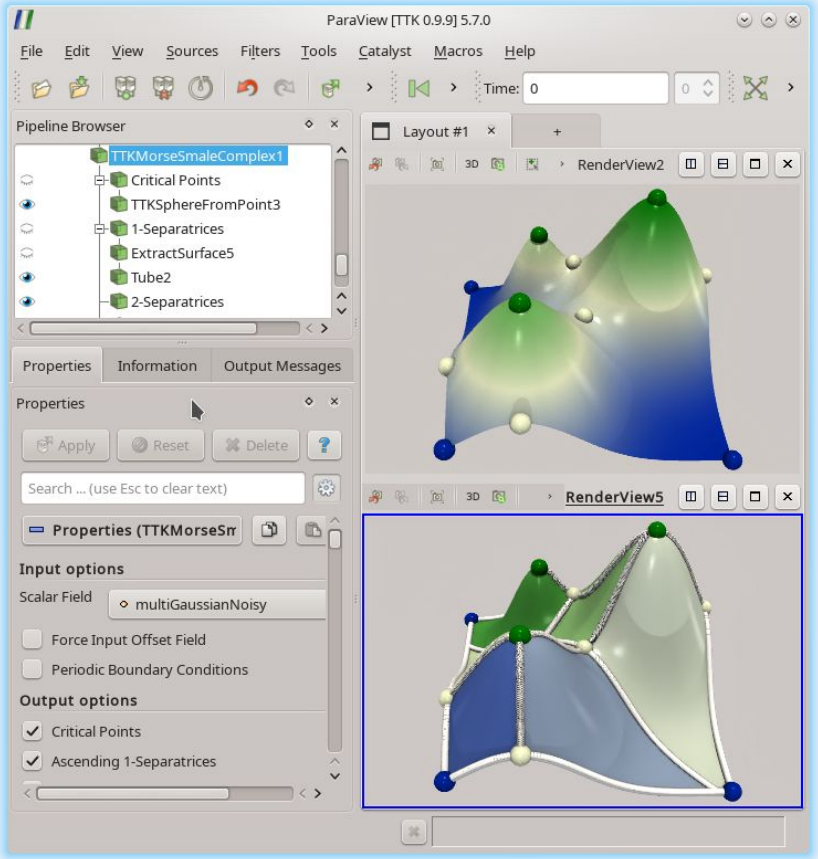
Reeb graphs

- **Module**
  - **`FTMGraph`**
  - Skeleton extraction
  - Level-set based segmentation
  - For non simply-connected domains
- **Algorithm**
  - Gueunet et al. EGPGV 2019
  - Linearithmic time
    - Fast Parallelization of Parsa SoCG 2012
- **Output**
  - Nodes of the trees
  - Arcs
  - Data segmentation



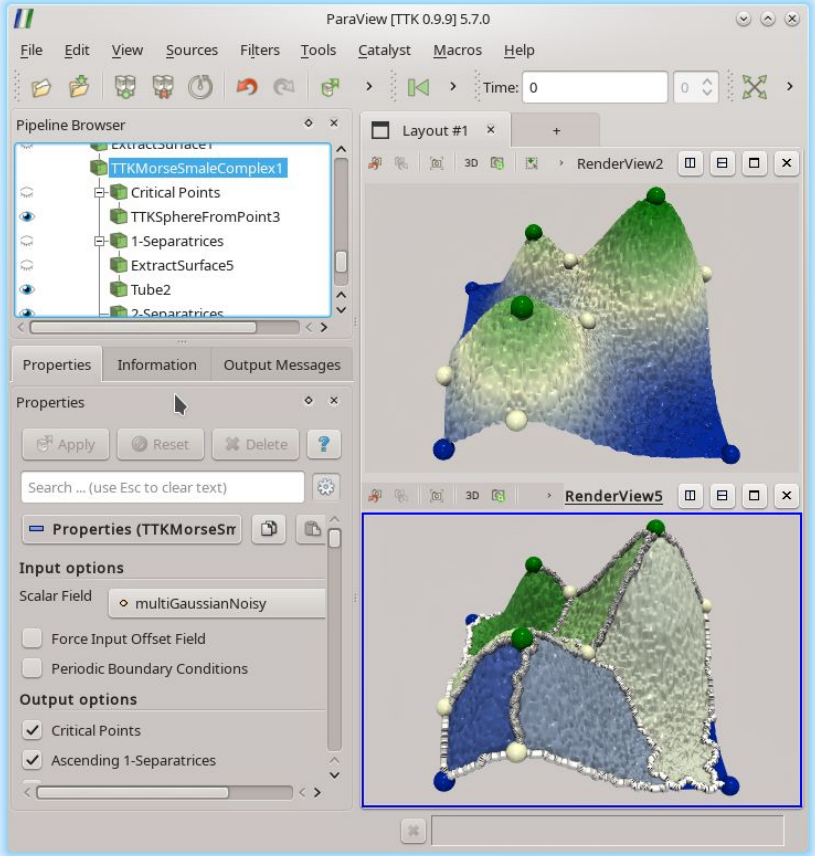
Morse-Smale complexes

- **Module**
  - **`MorseSmaleComplex`**
  - Extremal curves, separating surfaces
- **Algorithms**
  - Discrete Morse Theory, Forman SLC 2002
  - Complex extraction
    - Standard v-path collection (BFS)
    - Quadratic time (worst case)
  - Discrete gradient
    - Tierny et al. IEEE TVCG 2017
    - Robins et al. IEEE PAMI 2011
- **Output**
  - Critical points, 1D & 2D separatrices
  - Data segmentation

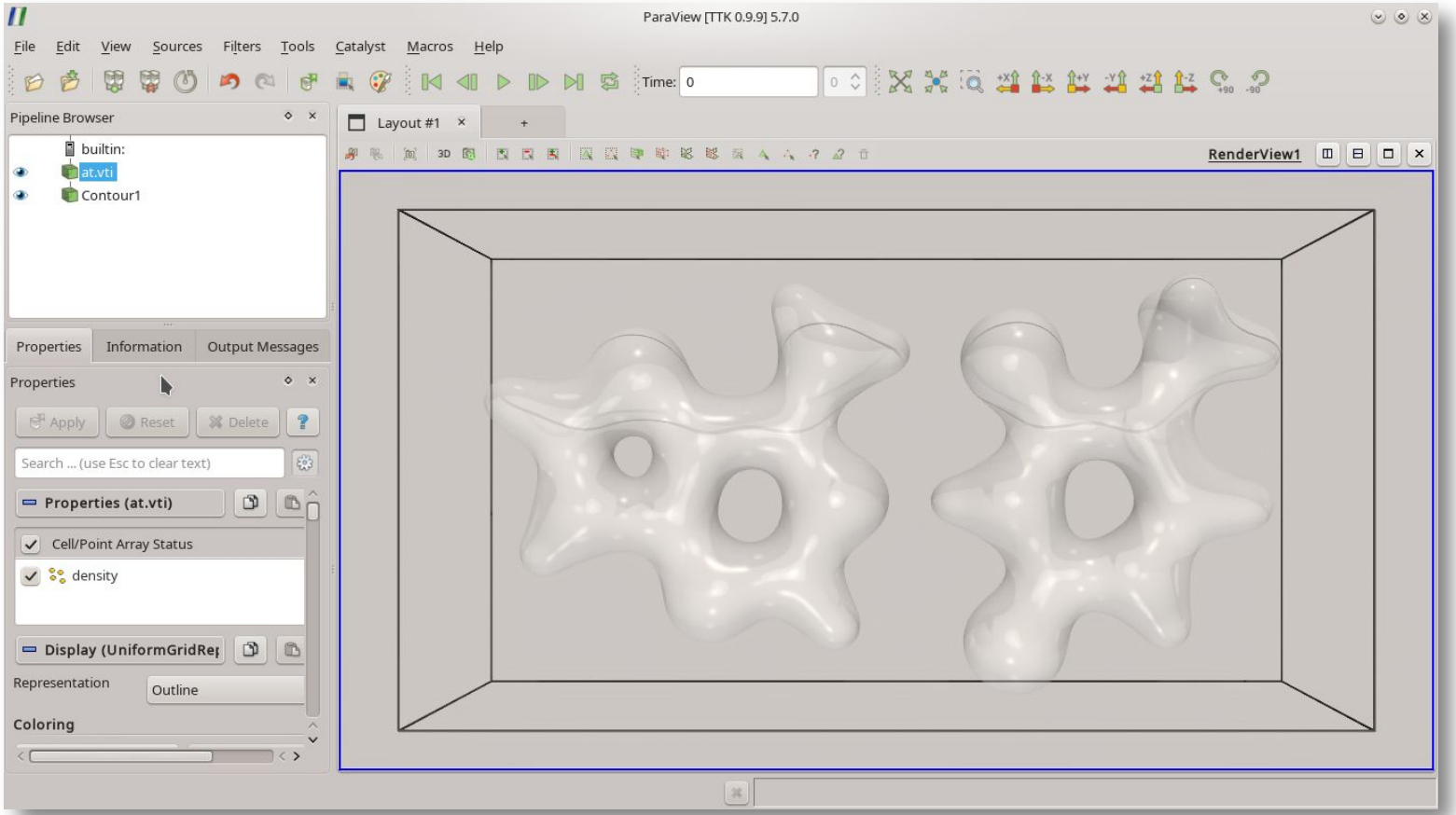


Morse-Smale complexes

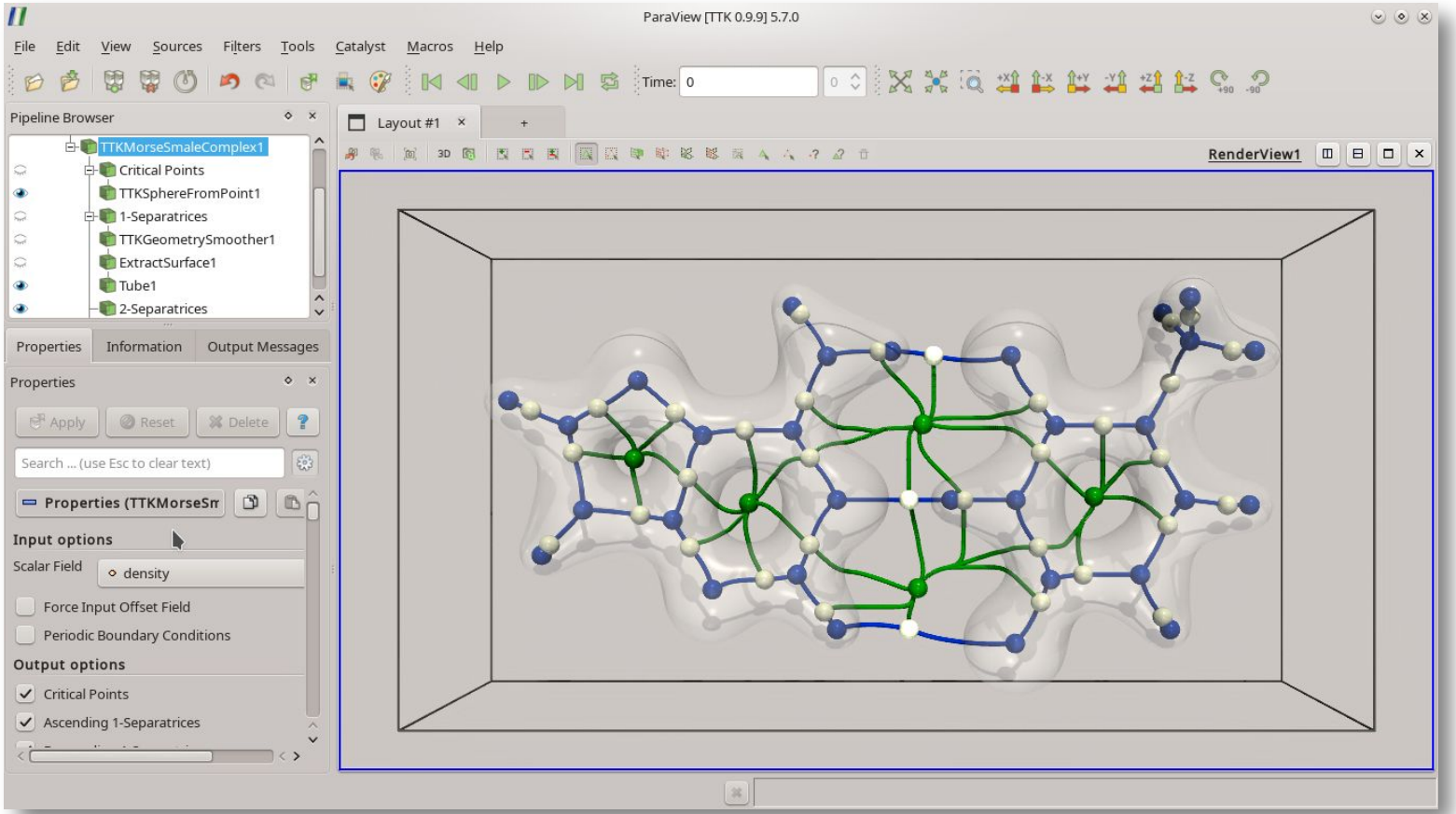
- **Module**
  - **`MorseSmaleComplex`**
  - Extremal curves, separating surfaces
- **Algorithms**
  - Discrete Morse Theory, Forman SLC 2002
  - Complex extraction
    - Standard v-path collection (BFS)
    - Quadratic time (worst case)
  - Discrete gradient
    - Tierny et al. IEEE TVCG 2017
    - Robins et al. IEEE PAMI 2011
- **Output**
  - Critical points, 1D & 2D separatrices
  - Data segmentation



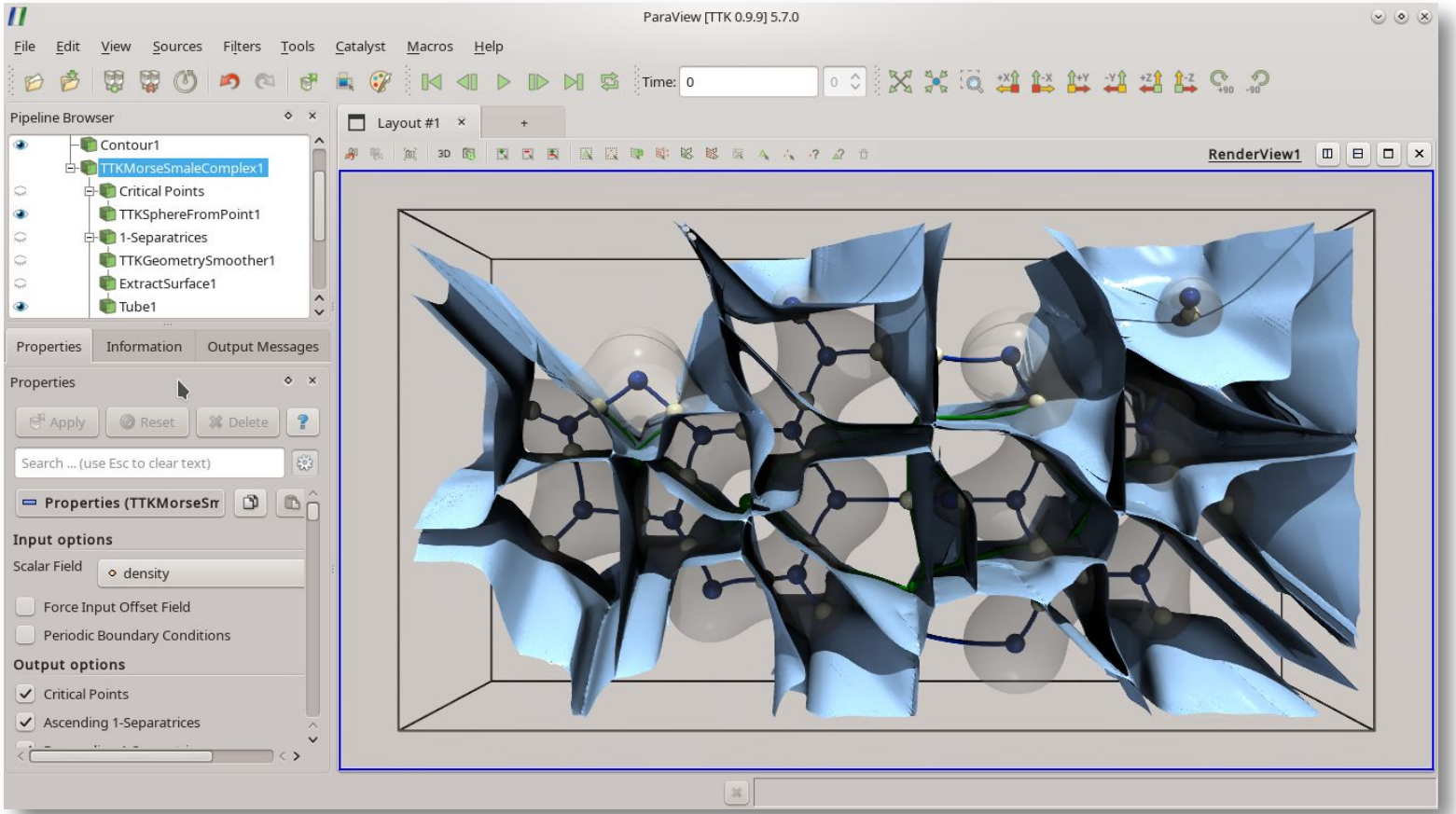
Application to quantum chemistry



Application to quantum chemistry

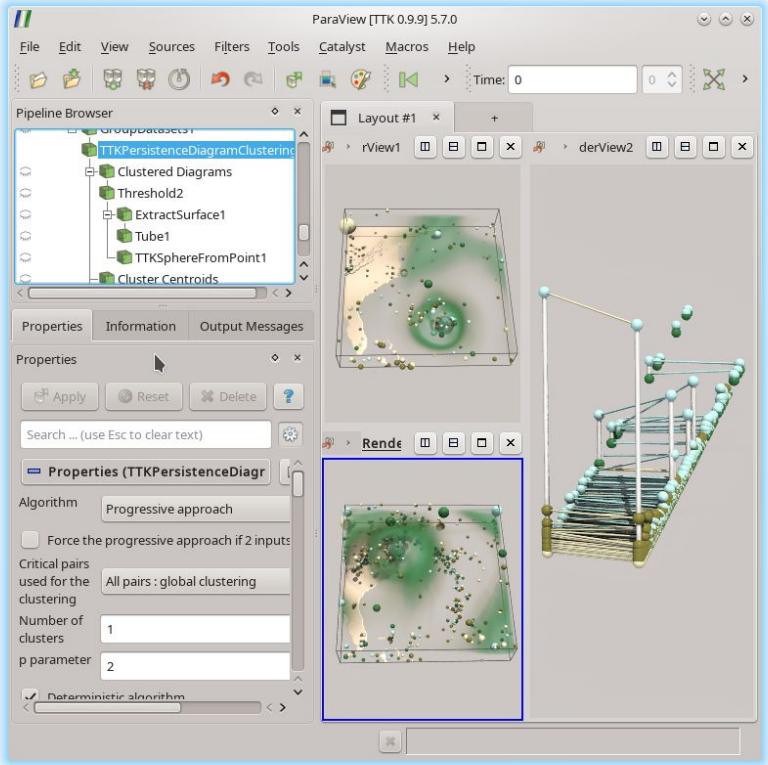


Application to quantum chemistry



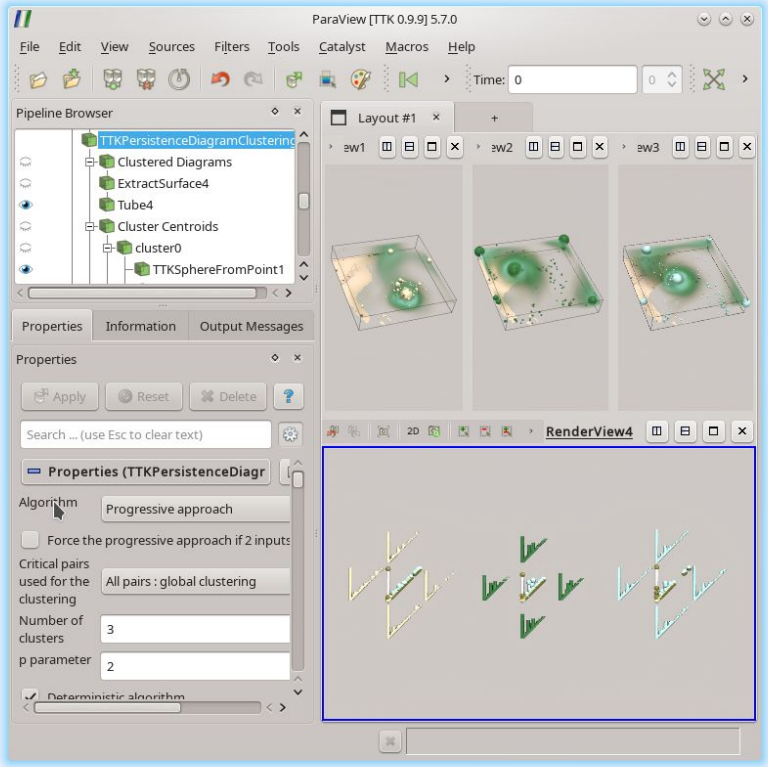
Distances between persistence diagrams

- **Module**
  - `Bottleneck`
  - `PersistenceDiagramClustering`
  - Bottleneck & Wasserstein
- **Algorithms**
  - Soler et al. IEEE LDAV 2018
    - Extends Bourgeois & Lassalle C. ACM 1971
  - Vidal et al. IEEE VIS 2019
    - Progressive Kerber et al. ACM JEA 2017
    - Computation time limits
- **Output**
  - Diagrams & Matching



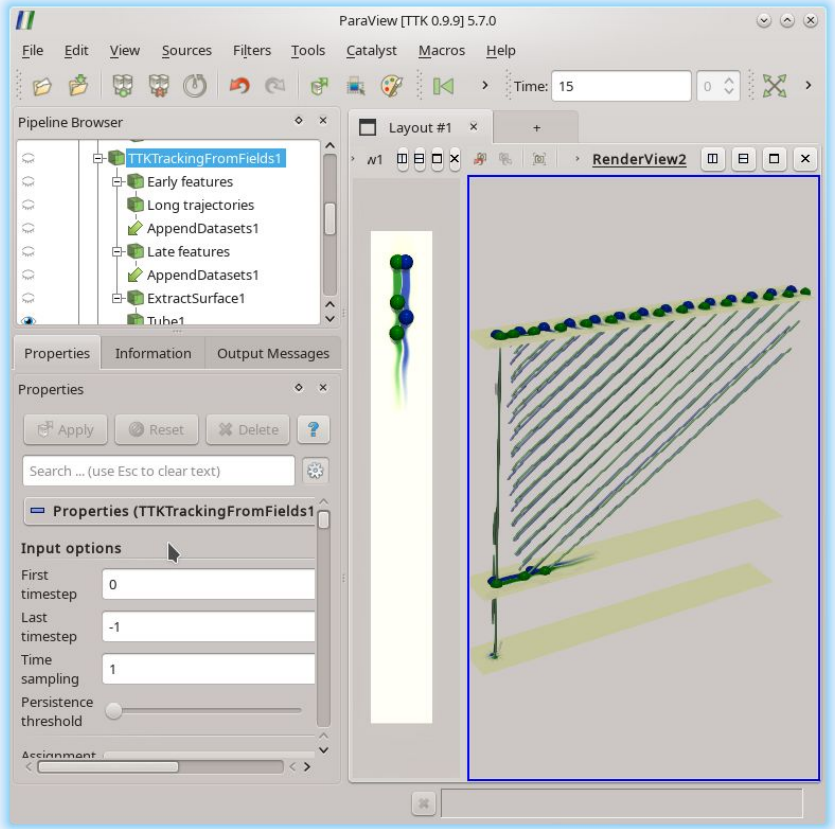
Barycenters/clusters of persistence diagrams

- **Module**
  - `PersistenceDiagramClustering`
  - Wasserstein barycenters + clustering
- **Algorithm**
  - Vidal et al. IEEE VIS 2019
    - Progressive approximation
      - Turner et al. DCG 2014
    - Fast convergence, parallelism
    - Computation time limits
- **Output**
  - Diagrams & Matching
  - Barycenters (cluster centers)



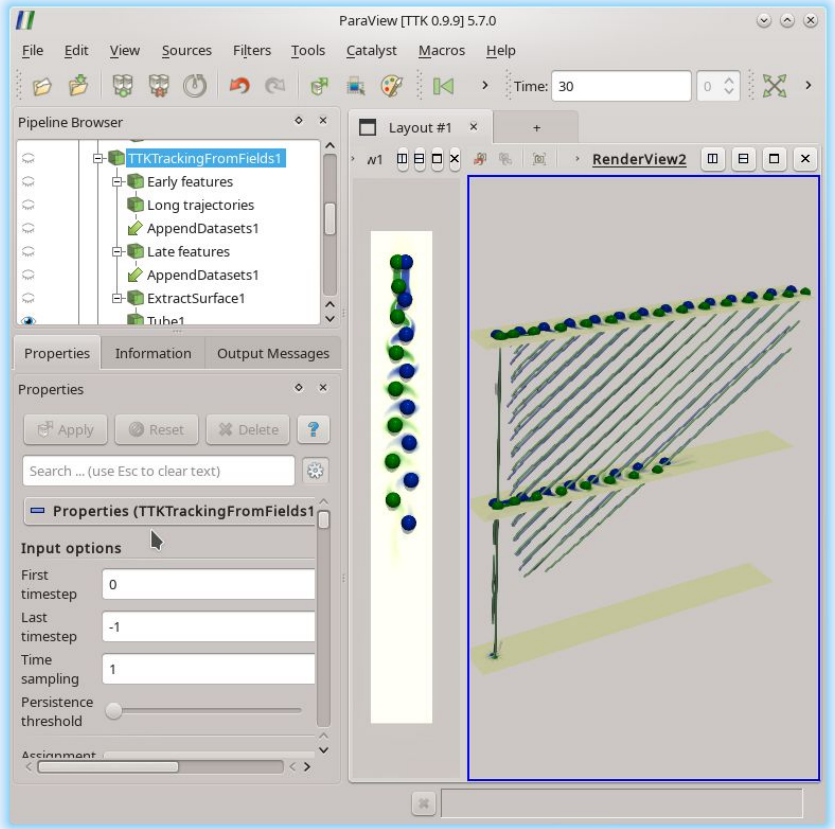
Time tracking

- **Module**
  - **`TrackingFromFields`**
  - **`TrackingFromPersistenceDiagrams`**
  - Minimizes Wasserstein distance
    - Lifted metric (geometry blending)
- **Algorithm**
  - Soler et al. IEEE LDAV 2018
- **Output**
  - Extremum temporal trajectories



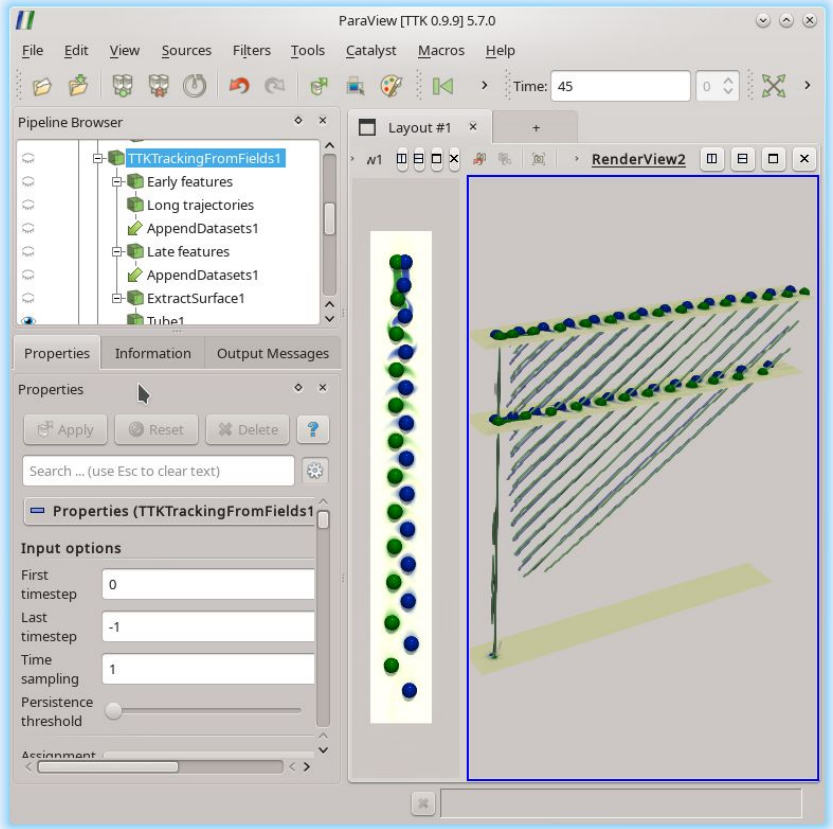
Time tracking

- **Module**
  - **`TrackingFromFields`**
  - **`TrackingFromPersistenceDiagrams`**
  - Minimizes Wasserstein distance
    - Lifted metric (geometry blending)
- **Algorithm**
  - Soler et al. IEEE LDAV 2018
- **Output**
  - Extremum temporal trajectories



Time tracking

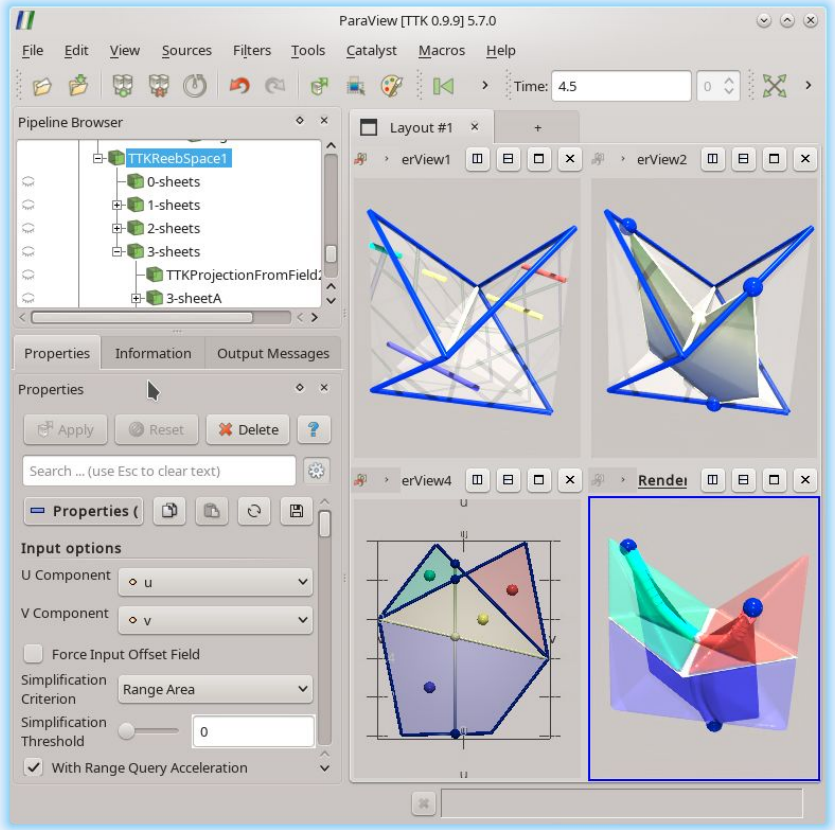
- **Module**
  - **`TrackingFromFields`**
  - **`TrackingFromPersistenceDiagrams`**
  - Minimizes Wasserstein distance
    - Lifted metric (geometry blending)
- **Algorithm**
  - Soler et al. IEEE LDAV 2018
- **Output**
  - Extremum temporal trajectories



Emerging data types

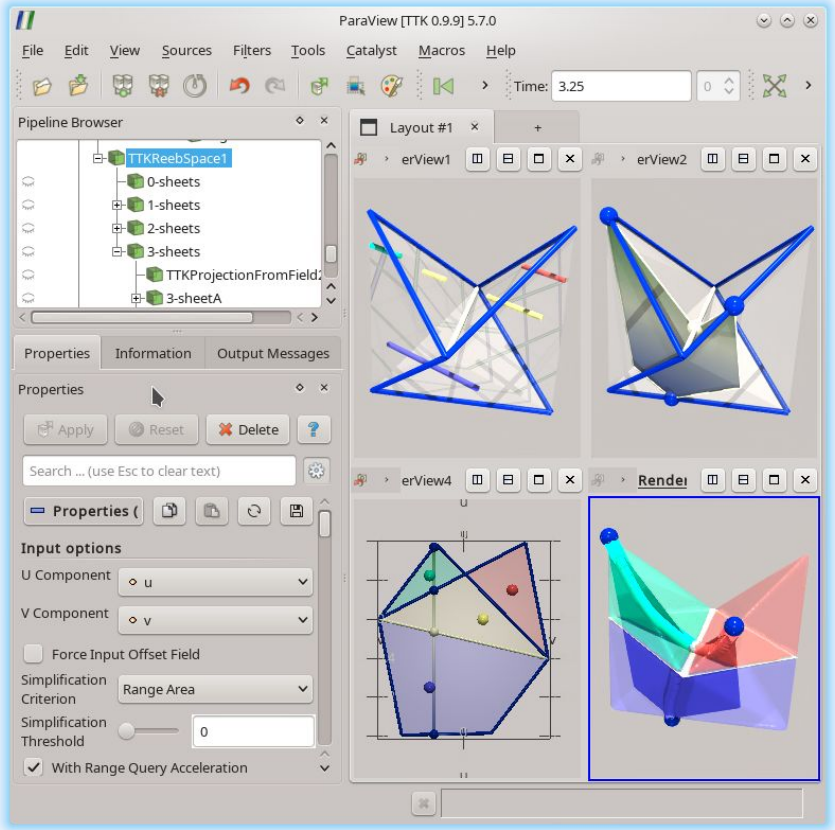
Bivariate data

- **Module**
  - `JacobiSet, ReebSpace`
  - `ContinuousScatterPlot`
  - Singularities extraction, segmentation
- **Algorithms**
  - Edelsbrunner & Harer FCM 2002
  - Bachthaler & Weiskopf IEEE TVCG 2008
  - Tierny & Carr IEEE TVCG 2016
- **Output**
  - Junction between Jacobi sets
  - Jacobi set curves (index and vertex IDs)
  - Sheets of the Reeb space & boundaries



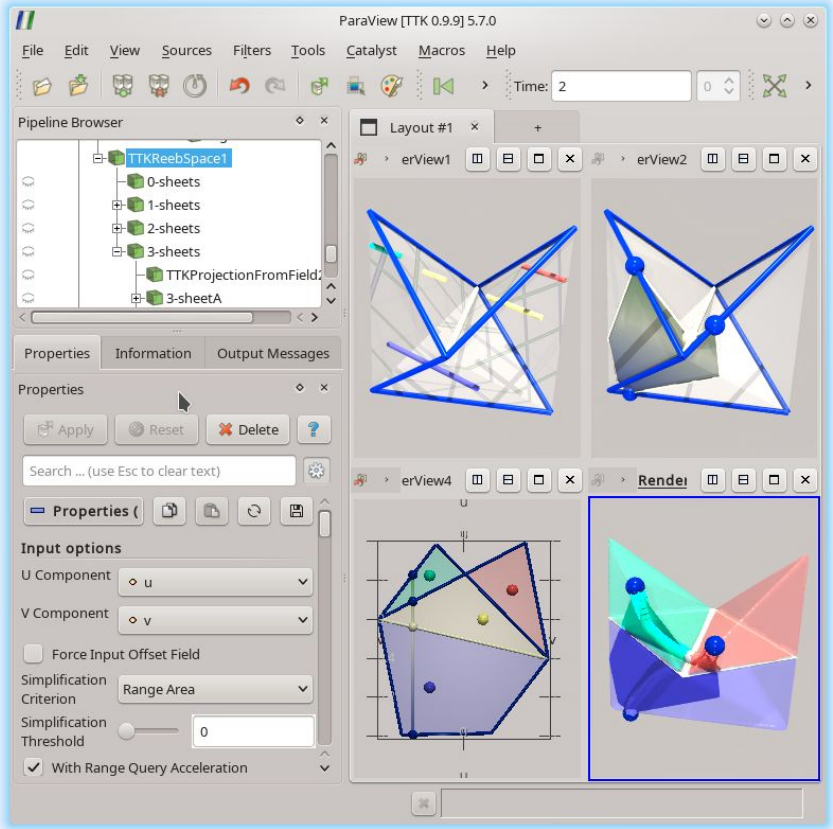
Bivariate data

- **Module**
  - `JacobiSet, ReebSpace`
  - `ContinuousScatterPlot`
  - Singularities extraction, segmentation
- **Algorithms**
  - Edelsbrunner & Harer FCM 2002
  - Bachthaler & Weiskopf IEEE TVCG 2008
  - Tierny & Carr IEEE TVCG 2016
- **Output**
  - Junction between Jacobi sets
  - Jacobi set curves (index and vertex IDs)
  - Sheets of the Reeb space & boundaries



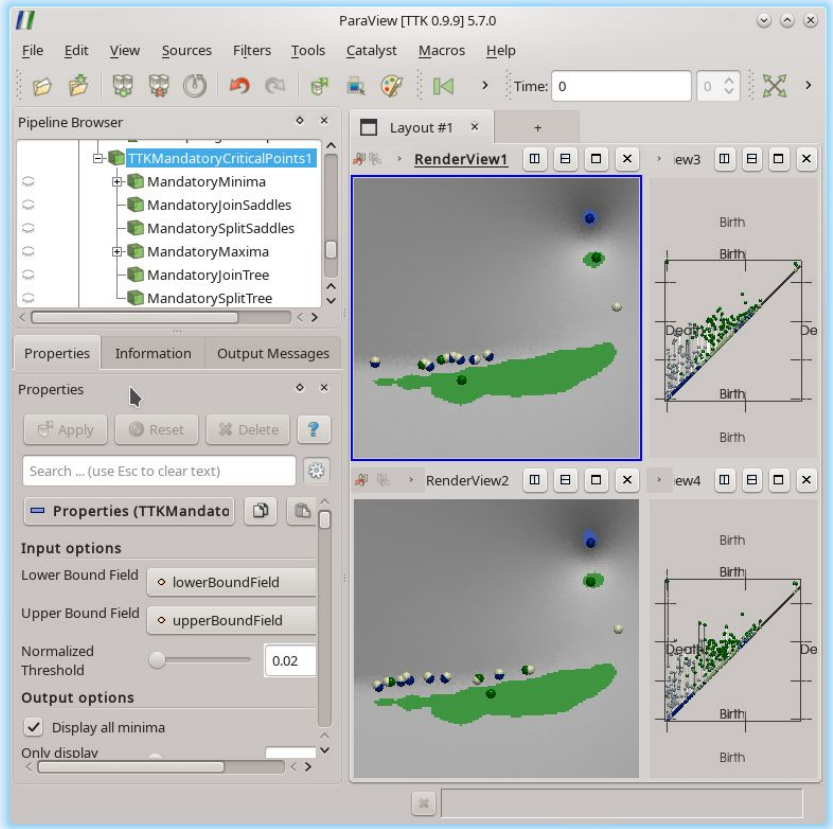
Bivariate data

- **Module**
  - `JacobiSet, ReebSpace`
  - `ContinuousScatterPlot`
  - Singularities extraction, segmentation
- **Algorithms**
  - Edelsbrunner & Harer FCM 2002
  - Bachthaler & Weiskopf IEEE TVCG 2008
  - Tierny & Carr IEEE TVCG 2016
- **Output**
  - Junction between Jacobi sets
  - Jacobi set curves (index and vertex IDs)
  - Sheets of the Reeb space & boundaries



Uncertain data

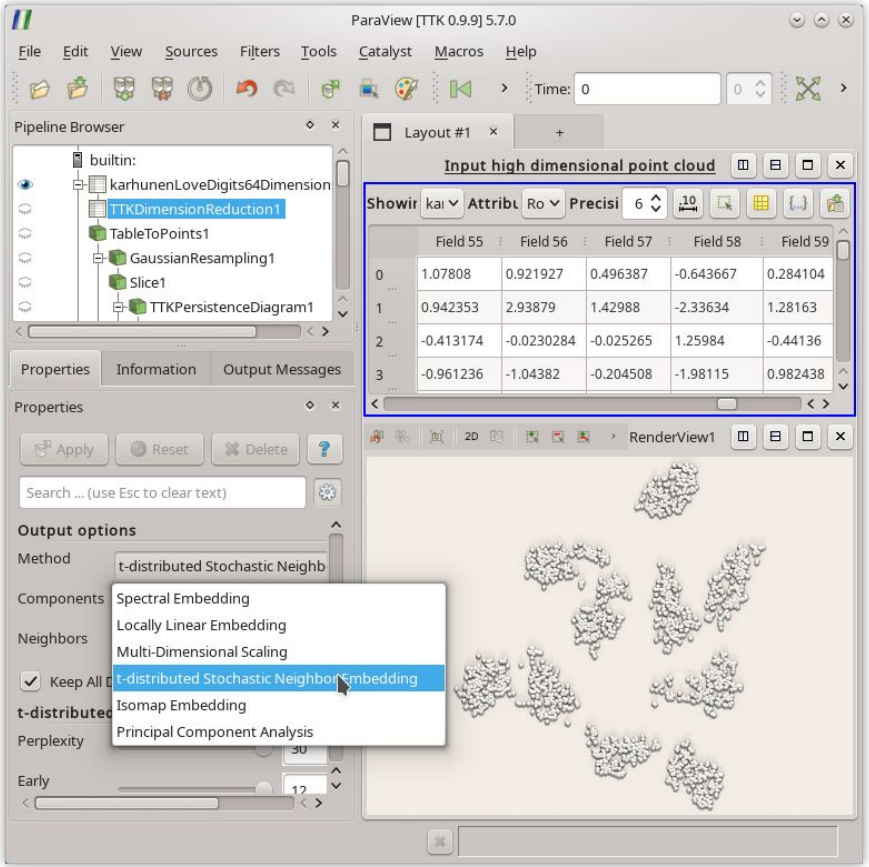
- **Module**
  - **`MandatoryCriticalPoints`**
  - Per-vertex error bounds
  - Critical points
    - Guaranteed regions of appearance
- **Algorithm**
  - Guenther et al. EuroVis 2014
- **Output**
  - Mandatory critical point regions
  - Mandatory merge trees



High dimensional point cloud data

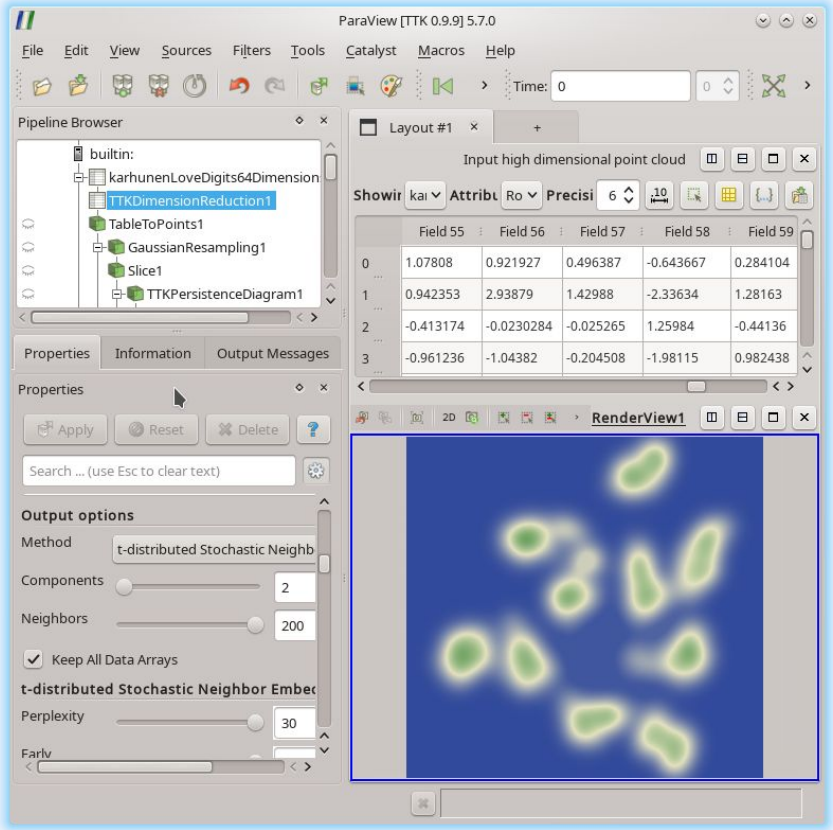
Dimension reduction

- **Module**
  - `DimensionReduction`
  - Wrapper to scikit-learn features
- **Algorithm**
  - Spectral Embedding
  - Locally Linear Embedding
  - Multi-Dimensional Scaling
  - t-Stochastic Neighbors Embedding
  - Isomap Embedding
  - Principal Component Analysis
- **Output**
  - Project point cloud (2D/3D)



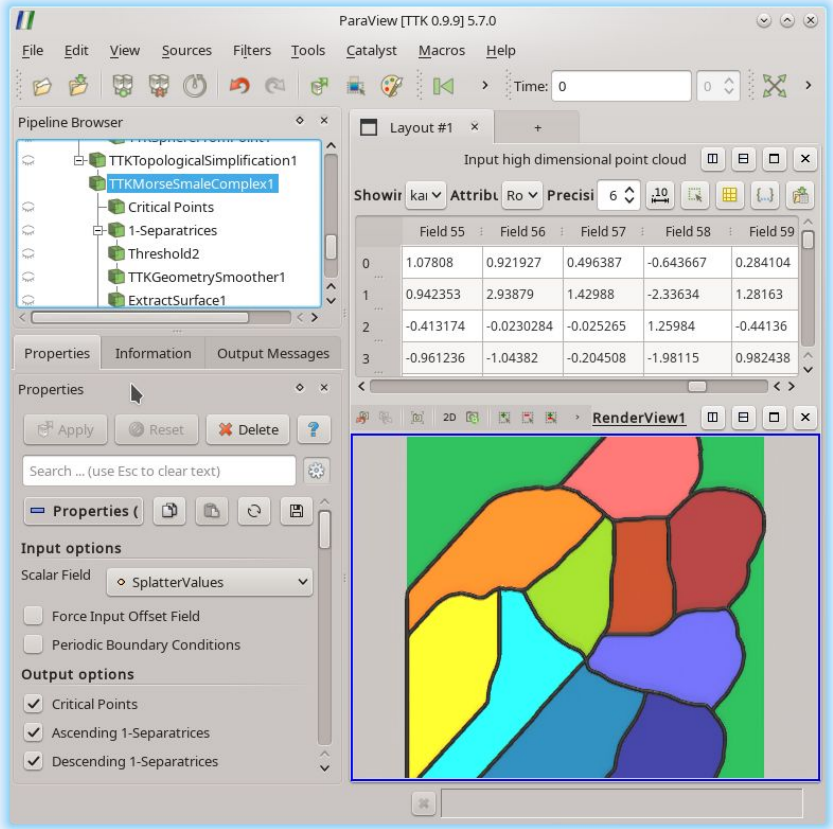
Persistence-based clustering

- **Module**
  - Combination of TTK modules
  - **`PersistenceDiagram`**
  - **`TopologicalSimplification`**
  - **`MorseSmaleComplex`**
- **Algorithm**
  - Chazal et al. SoCG 2011
- **Output**
  - Clustered point cloud



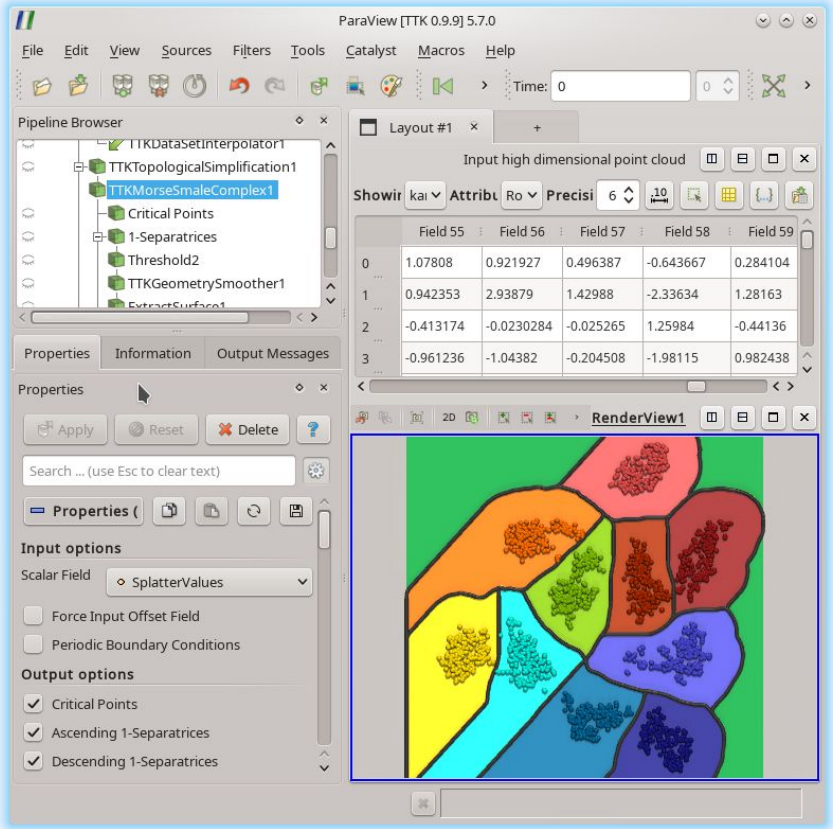
Persistence-based clustering

- **Module**
  - Combination of TTK modules
  - `PersistenceDiagram`
  - `TopologicalSimplification`
  - `MorseSmaleComplex`
- **Algorithm**
  - Chazal et al. SoCG 2011
- **Output**
  - Clustered point cloud



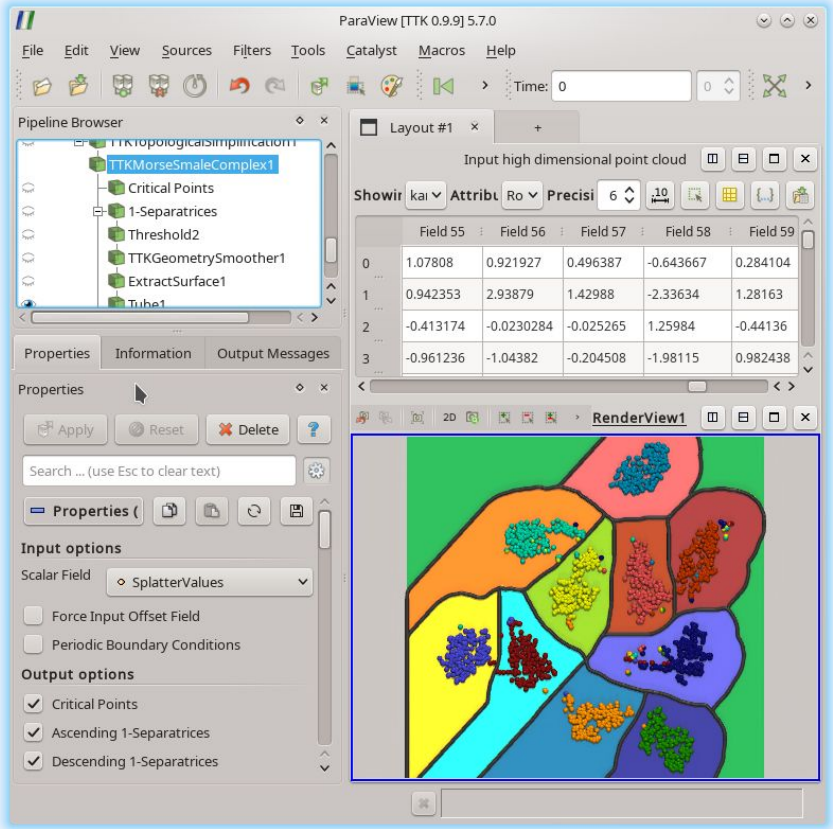
Persistence-based clustering

- **Module**
  - Combination of TTK modules
  - **`PersistenceDiagram`**
  - **`TopologicalSimplification`**
  - **`MorseSmaleComplex`**
- **Algorithm**
  - Chazal et al. SoCG 2011
- **Output**
  - Clustered point cloud



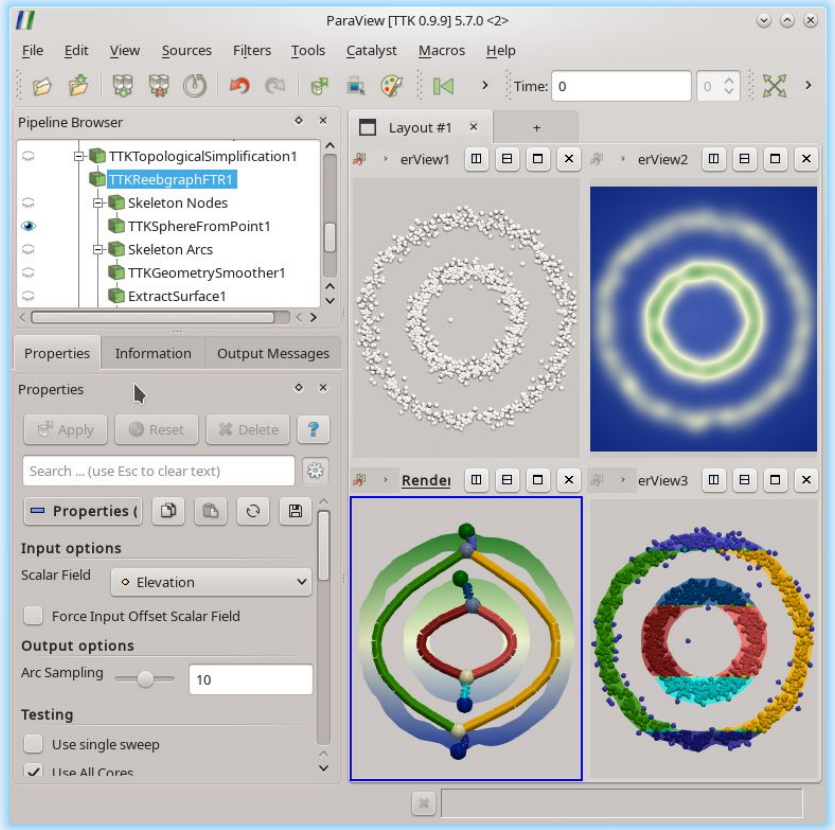
Persistence-based clustering

- **Module**
  - Combination of TTK modules
  - **`PersistenceDiagram`**
  - **`TopologicalSimplification`**
  - **`MorseSmaleComplex`**
- **Algorithm**
  - Chazal et al. SoCG 2011
- **Output**
  - Clustered point cloud



Mapper

- **Module**
  - **`FTRGraph`**
  - Density based Reeb graph
- **Algorithm**
  - Singh et al. EG PBG 2007
- **Output**
  - Graph
  - Clustering

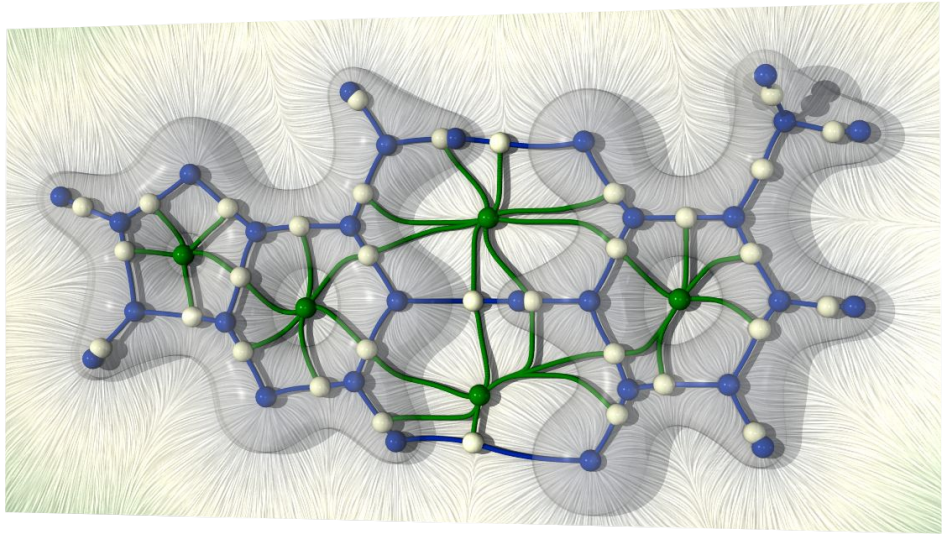


And many more!
https://topology-tool-kit.github.io/tutorials.html

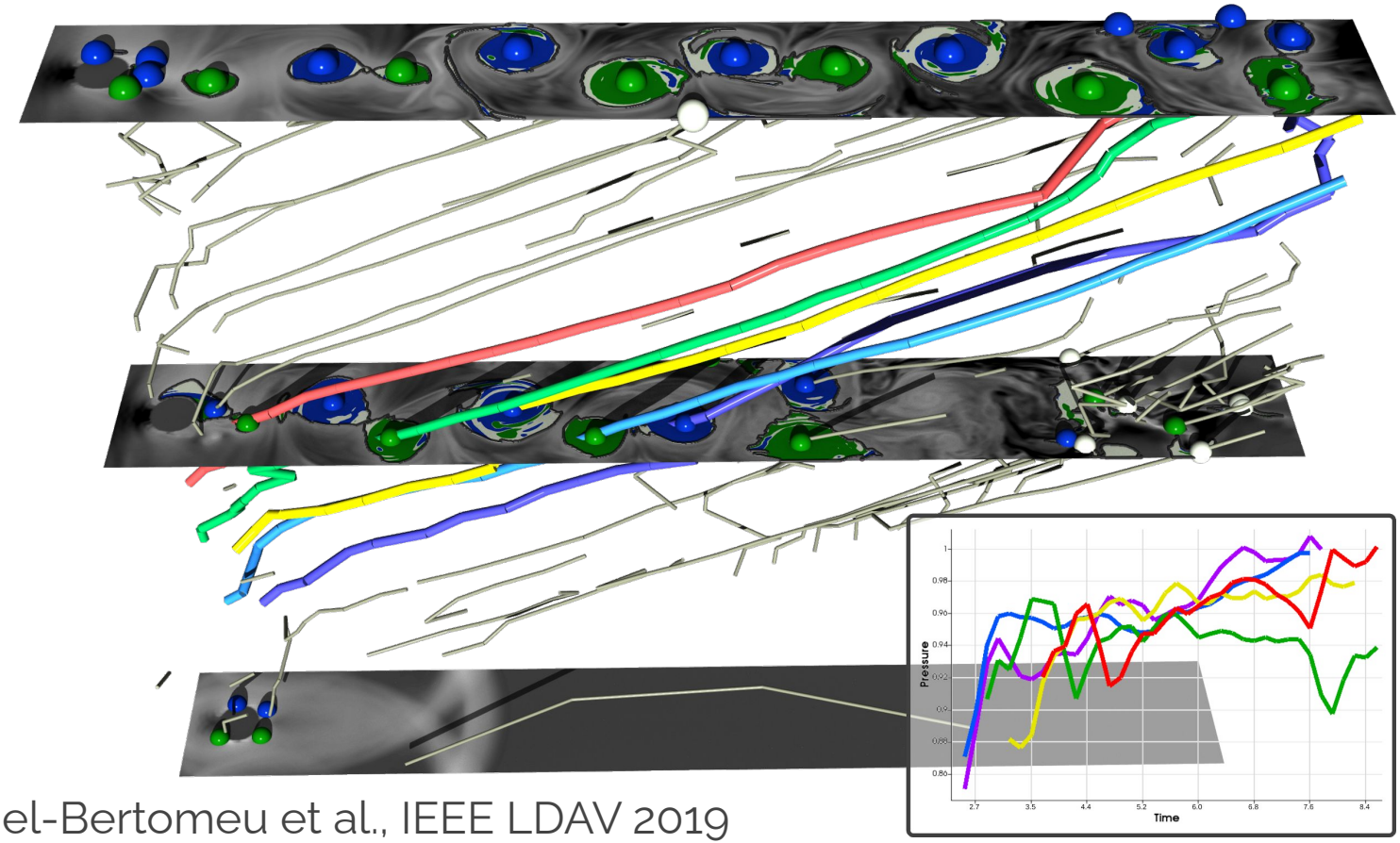
Examples of applications

Quantum chemistry

- **Non-covalent interactions**
  - Extract and analyze
  - Olejniczak et al. IJQC 2020
- **Other possible applications**
  - Bassin integration

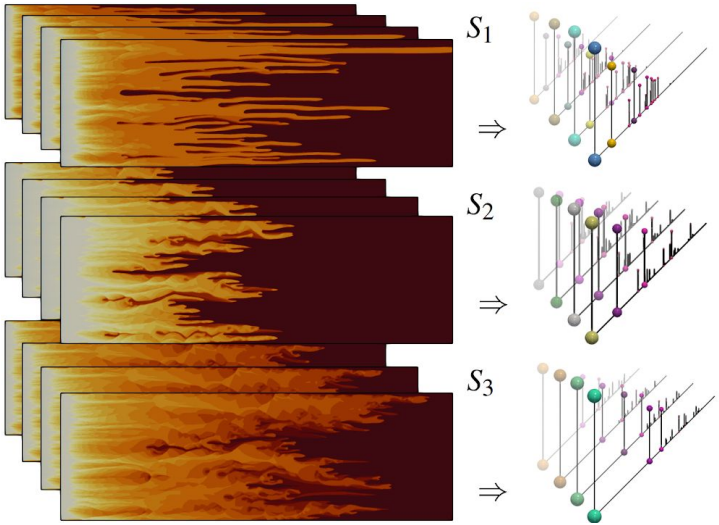


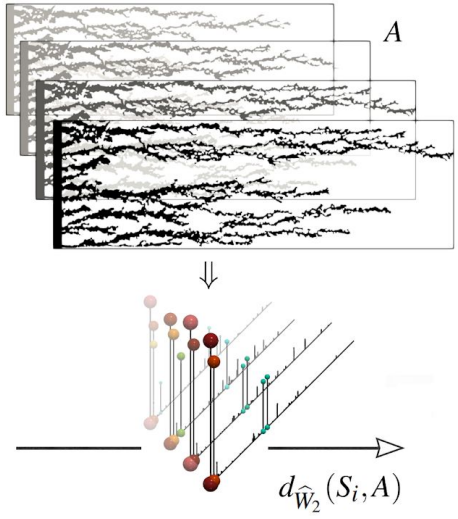
Turbulent flow analysis

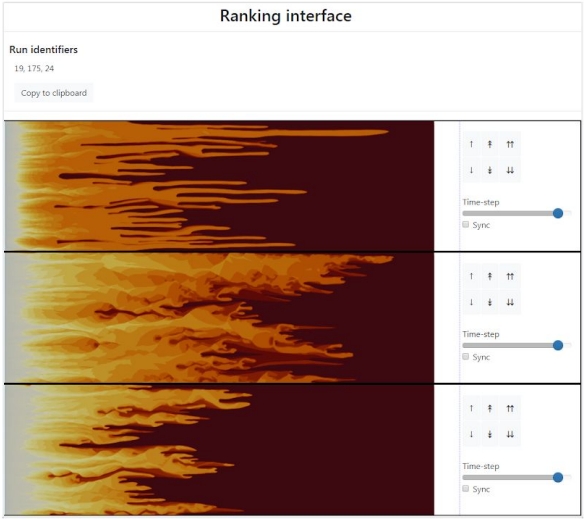


- Bridel-Bertomeu et al., IEEE LDAV 2019

Ensemble analysis (material sciences)







- Soler et al., IEEE LDAV 2019

Conclusion

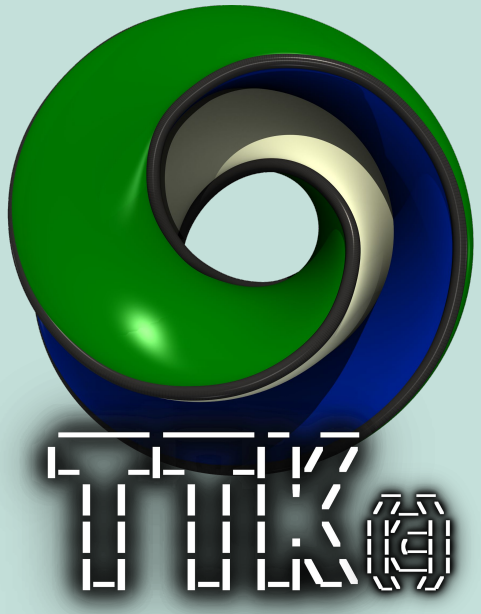
Join us!

- **For users**
  - Active user mailing list
  - Many tutorials, examples
  - Many available algorithms
  - Easy to interface
- **For developers**
  - TTK as a research platform
    - Rich IO support, advanced rendering
    - Scripting, efficient data structures
    - Leverage and combine
  - Easy module creation
- **Contribute!**
  - Everybody can help!
  - We welcome everyone :)

ttk-users@googlegroups.com

Conclusion

- **Roadmap**
  - Easier installation
    - Binary distributions
    - Integration with ParaView distribution
  - Extensions
    - Following research results
    - Vector / Tensor data?
    - High-dimensional data?
- **Community**
  - Data, example, tutorials
  - Mailing lists
  - Use TTK for your developments



Thanks!

- **Thanks to the TTK community!**
  - Developers & users

- **Code, documentation, tutorials, videos, etc.**
  - **http://topology-tool-kit.github.io**

- **We're hiring!**
  - 4 Ph.D. students, 2 post-docs, 2 engineers

- **Questions?**