Topological Data Analysis with TTK

A Four Years Post-Doc at LIP6

Pierre Guillou

Fontainebleau, March 20th, 2023

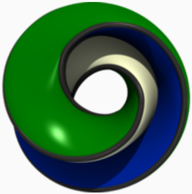
CNRS & Sorbonne Université

TTK : The Topology ToolKit

The Topology Toolkit (TTK)

A Topological Data Analysis software library

- `http://topology-tool-kit.github.io`
- open-source, BSD license
- ∼160k lines of C++, 7.5k commits
- several APIs
  - C++, VTK C++, VTK Python & ParaView Python
- ∼15 academic contributor
- Ubuntu packages & Windows installers
- TTK ⊂ ParaView official binaries! (since 2021)



Julien Tierny et al. “The Topology ToolKit”. In: *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS)* (2017), Talha Bin Masood et al. “An Overview of the Topology ToolKit”. In: *TopoInVis*. 2019

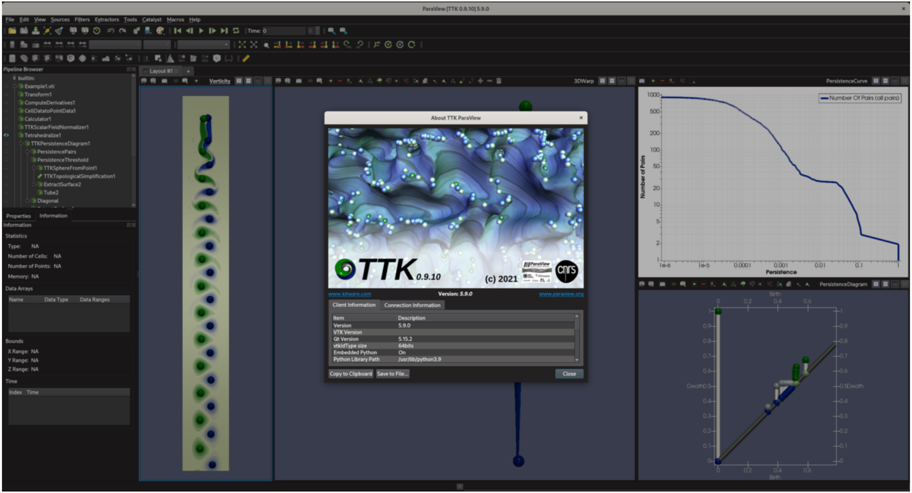
The ParaView/VTK ecosystem

VTK	Visualization ToolKit, a C++ software library
ParaView	Scientific Data Visualization application (C++/Qt)
CMake	Build system (mainly for C++ software)
Kitware, Inc.	N.Y.-based software company with a strong commitment on open-source software

How to use ParaView

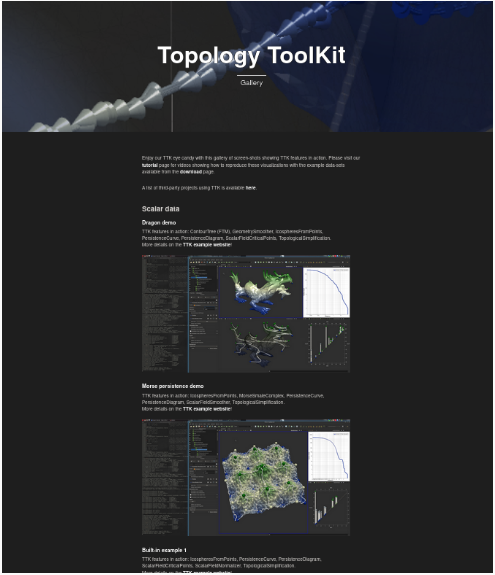
- Several types of views are proposed: *RenderView*, *SpreadsheetView*...
- *Filters* are applied on data, creating a *pipeline*
- Pipelines can be saved as Python scripts

TTK: a ParaView plugin



TTK Gallery / `ttk-data`

A collection of ParaView state files available on GitHub.

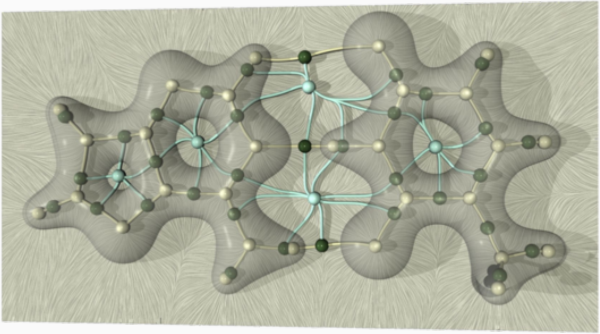


Outline

Topological Data Analysis: Key Concepts

A Basic Definition

Topological Data Analysis (TDA)



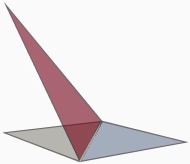
© Jules Vidal, « A Progressive Approach to Scalar Field Topology »

- studies *scalar fields* defined on a *domain*
  - regular grid, mesh...
- detects, measures & extracts *topological features*
  - peaks, valleys, cavities, cycles, noise...
- generates lightweight signatures used as proxies
  - persistence diagrams, merge trees...

On meshes

- A mesh = a set of polygons (2D) or polyhedron (3D)
- Usually 2D (squares or triangles) or 3D (cubes or tetrahedra).
- Basic topology: finding the connected components

Non-manifold meshes



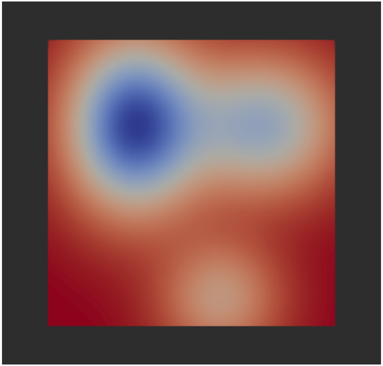
- Some meshes are more regular than others
- In 2D, a manifold mesh $\Rightarrow$ an edge must link either 1 or 2 squares/triangles

Complexes

- Ensemblist extension of a mesh
- If a cell $\in$ complex, then all its faces $\in$ complex
- Cubic complexes, simplicial complexes

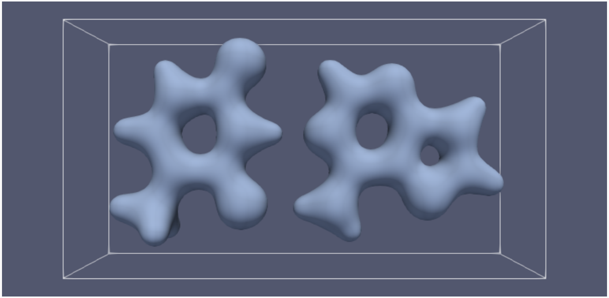
Example Data-Sets: Toy example

- 2D regular grid, 100×100
- $-1\times$ Sum of 3 gaussians
- Lower values, Higher values



Example Data-Sets: Molecular Simulation

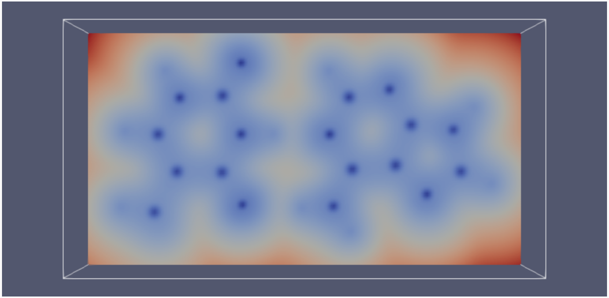
- 3D regular grid ($177 \times 95 \times 48$)
- Molecular bond between Adenine & Thymine (DNA nucleobases)
- Simulation of the electronic density probability ($-log()$)
- Lower values $\longrightarrow$ noyaux, Higher values on the boundary



Contour

Example Data-Sets: Molecular Simulation

- 3D regular grid ($177 \times 95 \times 48$)
- Molecular bond between Adenine & Thymine (DNA nucleobases)
- Simulation of the electronic density probability ($-log()$)
- Lower values $\longrightarrow$ noyaux, Higher values on the boundary



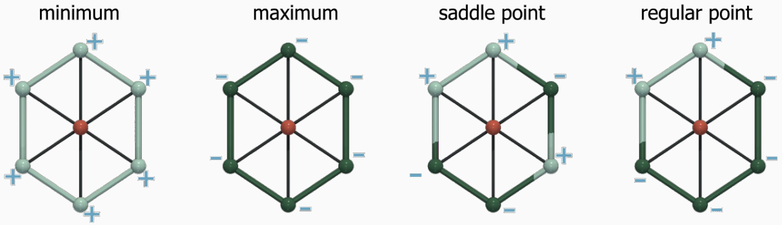
Slice

Topological Data Analysis: Key Concepts

Critical Points

Critical Points

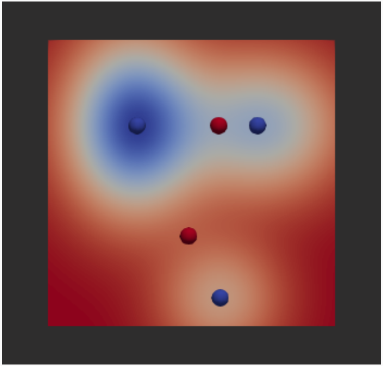
- Minima, maxima, saddle points
- Local characterization : counting the number of connected components in the lower/upper *link*
- 3D: 1-saddles & 2-saddles
- Behavior near the boundary?



© Jules Vidal, « A Progressive Approach to Scalar Field Topology »

Critical Points

- Minima, maxima, saddle points
- Local characterization : counting the number of connected components in the lower/upper *link*
- 3D: 1-saddles & 2-saddles
- Behavior near the boundary?



Minima, saddle points

Critical Points: Global Semantics & Topological Features

Tracking the connectivity of the sub-level sets

- **Minimum**: Birth of a connected component
- **Saddle**: Death of a connected component
  or Birth of a cycle / cavity
- **Maximum**: Death of a cycle / cavity

Elder Rule

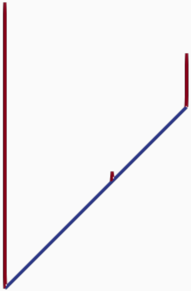
- The youngest topological feature dies (or emerges) in favor of the oldest

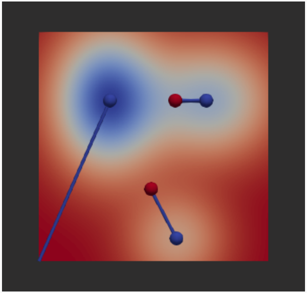
Topological Data Analysis: Key Concepts

Persistence Diagrams

Persistence Diagrams

- (Exclusive) Pairs of critical points
- A pair = one topological feature
  - min-saddle pairs ($\mathcal{D}_0(f)$): connected components
  - saddle-max pairs ($\mathcal{D}_{d-1}(f)$): cavities
  - saddle-saddle pairs ($\mathcal{D}_1(f)$): cycles
- Pair height = *persistence* of the feature
- Lightweight & stable representation
  - noisy feature have a small persistence

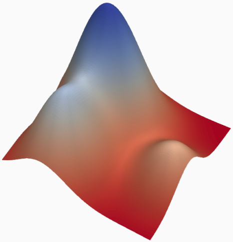


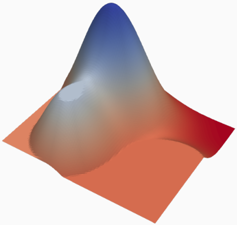


Application: Topological Simplification

With a given noisy scalar field:

1. the Persistence Diagram is computed
2. the persistence pairs under a given threshold are removed
3. a new scalar field is generated, corresponding to the simplified topology (levelling)





Julien Tierny and Valerio Pascucci. "Generalized Topological Simplification of Scalar Fields on Surfaces". In: *IEEE Transactions on Visualization and Computer Graphics* (Dec. 2012). URL: `https://hal.archives-ouvertes.fr/hal-01206877`, Jonas Lukasczyk et al. "Localized Topological Simplification of Scalar Data". In: *IEEE Transactions on Visualization and Computer Graphics* (Oct. 2020). URL: `https://hal.archives-ouvertes.fr/hal-02949278`

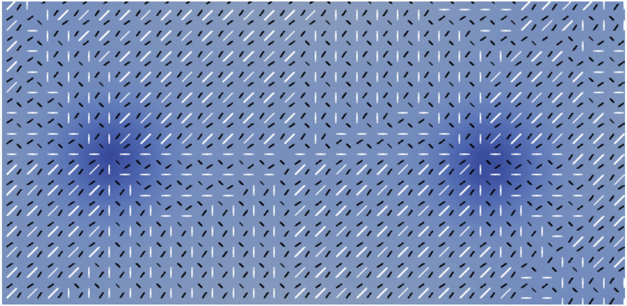
Topological Data Analysis: Key Concepts

Discrete Gradient & Morse-Smale Complex

Discrete Gradient

In a (cubic, simplicial) complex, pair each cell with

- either one of its faces,
- either one of its co-faces, (vertex $\longrightarrow$ edge towards the lowest neighbor)
- or it's a *critical* cell (its highest vertex is a critical point)

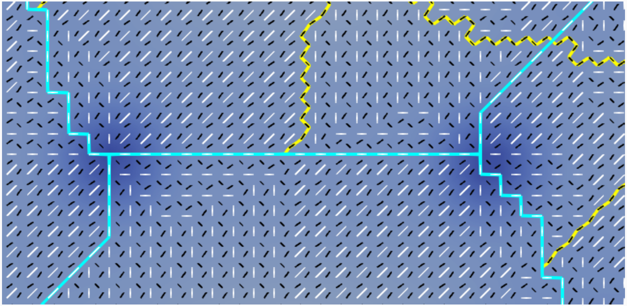


Vanessa Robins, Peter John Wood, and Adrian P. Sheppard. "Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011), pp. 1646–1658. DOI: 10.1109/TPAMI.2011.95

Discrete Gradient

In a (cubic, simplicial) complex, pair each cell with

- either one of its faces,
- either one of its co-faces, (vertex $\longrightarrow$ edge towards the lowest neighbor)
- or it's a *critical* cell (its highest vertex is a critical point)

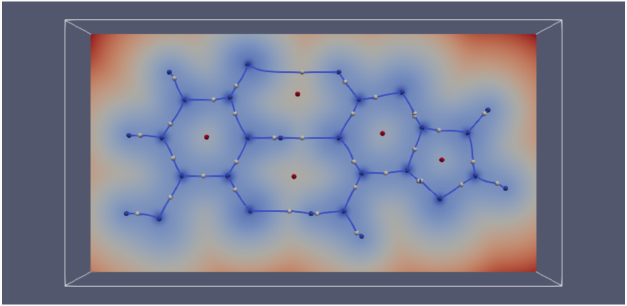


Vanessa Robins, Peter John Wood, and Adrian P. Sheppard. "Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011), pp. 1646–1658. DOI: 10.1109/TPAMI.2011.95

Morse-Smale Complex

critical cells from the Discrete Gradient

descending 1-separatrices follow the gradient downstream from the 1-saddles



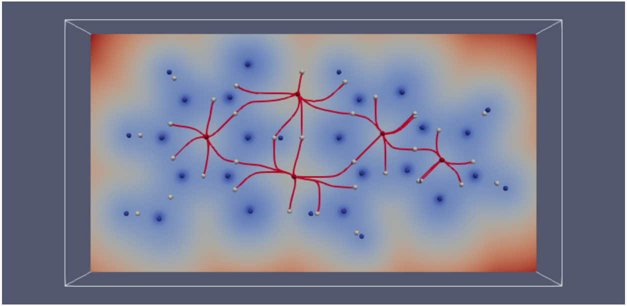
Descending 1-separatrices

Morse-Smale Complex

critical cells from the Discrete Gradient

descending 1-separatrices follow the gradient downstream from the 1-saddles

ascending 1-separatrices follow gradient upstream from the (D-1)-saddles



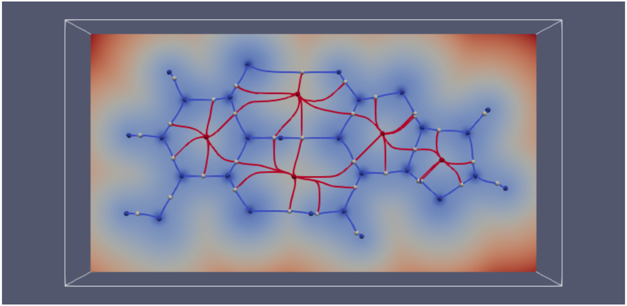
Ascending 1-separatricess

Morse-Smale Complex

critical cells from the Discrete Gradient

descending 1-separatrices follow the gradient downstream from the 1-saddles

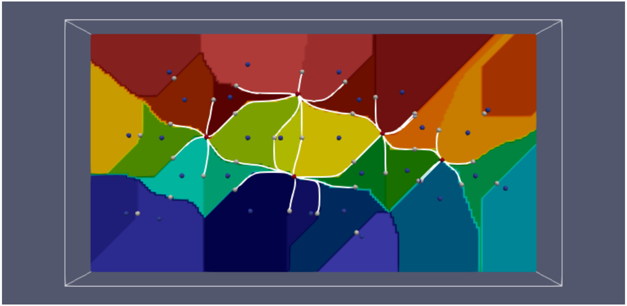
ascending 1-separatrices follow gradient upstream from the (D-1)-saddles



Ascending + descending 1-separatrices

Morse-Smale Complex

critical cells from the Discrete Gradient
descending 1-separatrices follow the gradient downstream from the 1-saddles
ascending 1-separatrices follow gradient upstream from the (D-1)-saddles
descending segmentation zone of influence of every minimum



Descending segmentation

Morse-Smale Complex

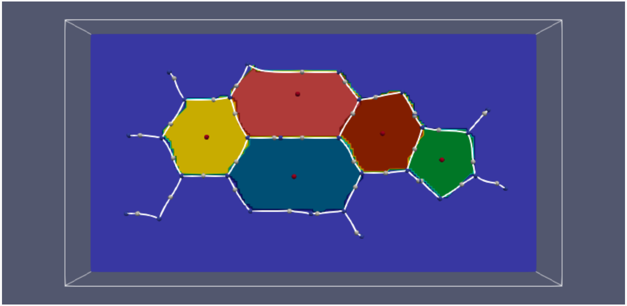
critical cells from the Discrete Gradient

descending 1-separatrices follow the gradient downstream from the 1-saddles

ascending 1-separatrices follow gradient upstream from the (D-1)-saddles

descending segmentation zone of influence of every minimum

ascending segmentation zone of influence of every maximum



Ascending segmentation

Morse-Smale Complex

critical cells from the Discrete Gradient

descending 1-separatrices follow the gradient downstream from the 1-saddles

ascending 1-separatrices follow gradient upstream from the (D-1)-saddles

descending segmentation zone of influence of every minimum

ascending segmentation zone of influence of every maximum

Morse-Smale segmentation ascending $\otimes$ descending segmentation

Morse-Smale segmentation

Morse-Smale Complex

critical cells from the Discrete Gradient

descending 1-separatrices follow the gradient downstream from the 1-saddles

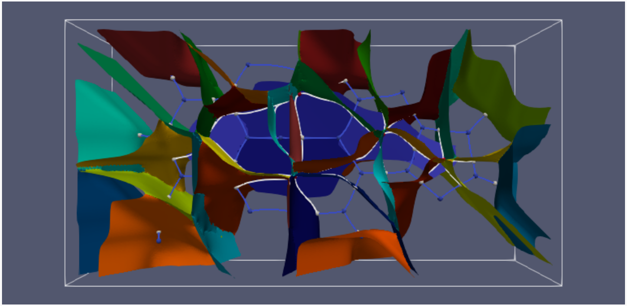
ascending 1-separatrices follow gradient upstream from the (D-1)-saddles

descending segmentation zone of influence of every minimum

ascending segmentation zone of influence of every maximum

Morse-Smale segmentation ascending $\otimes$ descending segmentation

2-separatrices (3D) boundary surfaces of each zone of influence



Ascending + descending 2-separatrices

Topological Data Analysis: Key Concepts

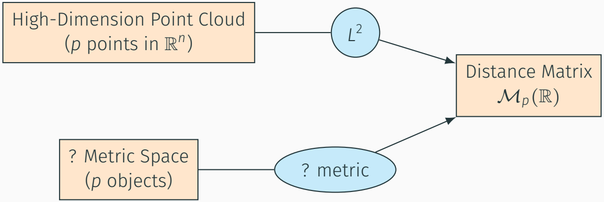
Topology of High-Dimension Data-Sets

Topology of High-Dimension Data-Sets

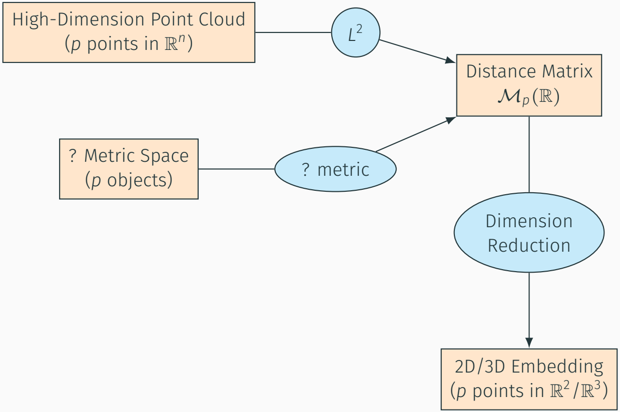
High-Dimension Point Cloud
(p points in $\mathbb{R}^n$)

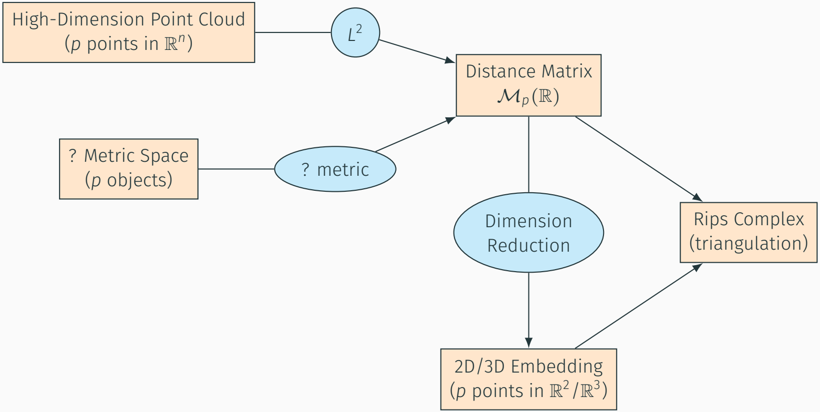
? Metric Space
(p objects)

Topology of High-Dimension Data-Sets



Topology of High-Dimension Data-Sets




High-Dimension Point Cloud
(p points in $\mathbb{R}^n$)
L^2
Distance Matrix
$\mathcal{M}_p(\mathbb{R})$
? Metric Space
(p objects)
? metric
Dimension
Reduction
Rips Complex
(triangulation)
2D/3D Embedding
(p points in $\mathbb{R}^2/\mathbb{R}^3$)

Rips Complex (2022)

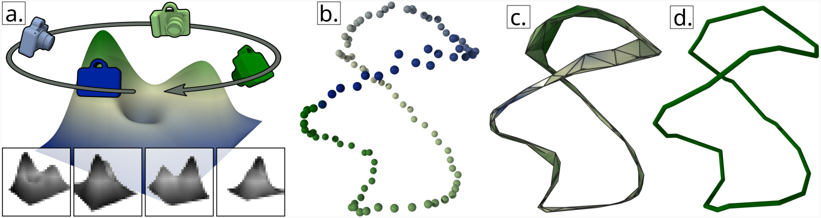
Generate a triangulation from a Distance Matrix

Algorithm 1: Find all (d+1)-uplets whose pairwise distance is $< \epsilon$

Input: d — Max simplex dimension
Input: ϵ — Max diameter
Input: Mat — Distance matrix
Input: E — 2D/3D Embedding
Output: Tr — Triangulation on embedding

$Tr \leftarrow \emptyset$;
foreach $(d+1)$-*uplet* $(p_0, .., p_d)$ **do**
    if $\forall i, j \in [0..d] Mat(p_i, p_j) < \epsilon$ **then**
        $Tr \leftarrow Tr \cup$ d-simplex from $(E(p_0), .., E(p_d))$;
    end if
end foreach
return Tr;

Illustration: Periodic Picture



DiscreteMorseSandwich

Fast computation of Persistence Diagrams (2021 – 2023)

Computing Persistence Diagrams from the Discrete Gradient

Discrete Morse Sandwich: Fast Computation of Persistence Diagrams for Scalar Data – An Algorithm and A Benchmark

Pierre Guillou, Jules Vidal, and Julien Tierny

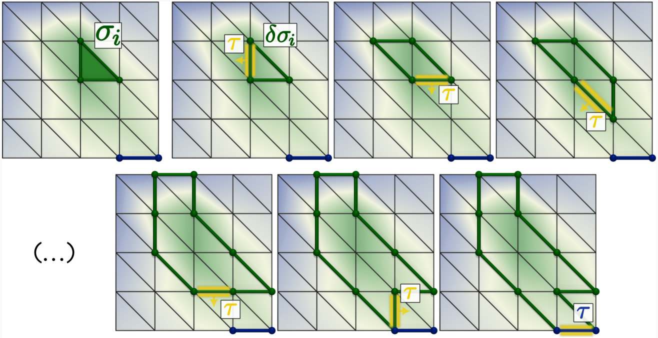
Abstract—This paper introduces an efficient algorithm for persistence diagram computation, given an input piecewise linear scalar field f defined on a d-dimensional simplicial complex $\mathcal{K}$, with $d \leq 3$. Our work revisits the seminal algorithm *"PairSimplices"* [31], [103] with discrete Morse theory (DMT) [34], [80], which greatly reduces the number of input simplices to consider. Further, we also extend to DMT and accelerate the stratification strategy described in *"PairSimplices"* [31], [103] for the fast computation of the 0^{th} and $(d-1)^{th}$ diagrams, noted $\mathcal{D}_0(f)$ and $\mathcal{D}_{d-1}(f)$. Minima-saddle persistence pairs ($\mathcal{D}_0(f)$) and saddle-maximum persistence pairs ($\mathcal{D}_{d-1}(f)$) are efficiently computed by processing, with a Union-Find, the unstable sets of 1-saddles and the stable sets of $(d-1)$-saddles. We provide a detailed description of the (optional) handling of the boundary component of $\mathcal{K}$ when processing $(d-1)$-saddles. This fast pre-computation for the dimensions 0 and $(d-1)$ enables an aggressive specialization of [4] to the 3D case, which results in a drastic reduction of the number of input simplices for the computation of $\mathcal{D}_1(f)$, the intermediate layer of the *sandwich*. Finally, we document several performance improvements via shared-memory parallelism. We provide an open-source implementation of our algorithm for reproducibility purposes. We also contribute a reproducible benchmark package, which exploits three-dimensional data from a public repository and compares our algorithm to a variety of publicly available implementations. Extensive experiments indicate that our algorithm improves by two orders of magnitude the time performance of the seminal *"PairSimplices"* algorithm it extends. Moreover, it also improves memory footprint and time performance over a selection of 14 competing approaches, with a substantial gain over the fastest available approaches, while producing a strictly identical output. We illustrate the utility of our contributions with an application to the fast and robust extraction of persistent 1-dimensional generators on surfaces, volume data and high-dimensional point clouds.

Index Terms—Topological data analysis, scalar data, persistence diagrams, discrete Morse theory.

Now the *default* Persistence Diagram algorithm in TTK!

DiscreteMorseSandwich: Key ideas

- Use the Discrete Gradient to detect critical cells;
- Detect first the min-saddle and saddle-max pairs
    - less saddles to consider for the saddle-saddle pairs
- *Boundary expansions* following the Discrete Gradient
    - Specific accelerations for the min-saddle and the saddle-max pairs
    - *Counterintuitive behavior* on non-manifold surfaces!

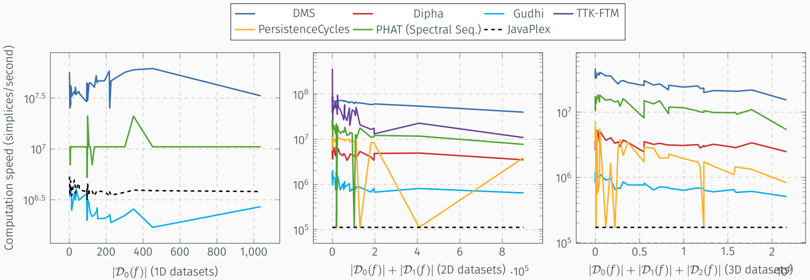


A Benchmark

Implementation	Ref.	Category	Language	Parallelism	Distance
DiscreteMorseSandwich	[9]	Discrete Morse Theory	C++	Controllable	**0.0**
PairSimplices	[7, 25]	Explicit Propagation	C++	No	**0.0**
TTK-FTM	[8]	Merge-Tree (2D)	C++	Controllable	122.5×10^6
PersistenceCycles	[11]	Discrete Morse Theory	C++	Controllable	97.5×10^3
Dionysus2	[15]	Boundary Matrix	C++	No	**0.0**
DIPHA	[2]	Boundary Matrix	C++	Controllable	**0.0**
Eirene.jl	[10]	Boundary Matrix	Julia	No	9.0×10^3
Gudhi	[14]	Boundary Matrix	C++	Observed	15.3×10^3
Javaplex	[19]	Boundary Matrix	Java	Observed	**0.0**
PHAT (Spectral Seq.)	[3]	Boundary Matrix	C++	Controllable	466.6×10^3
Ripser.py	[1, 22]	Boundary Matrix	C++	No	NA
CubicalRipser	[12]	Boundary Matrix	C++	No	NA
Oineus	[17]	Boundary Matrix	C++	Controllable	NA
Perseus	[16]	Discrete Morse Theory	C++	No	NA
Diamorse	[5]	Discrete Morse Theory	C++	No	NA

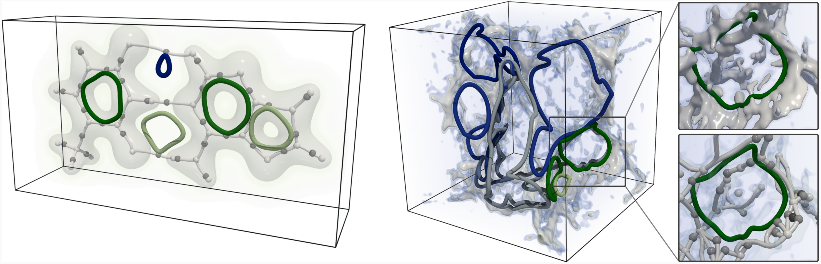
$$\sim 15 \text{ implementations} \times 36 \text{ datasets} \times \{1D, 2D, 3D\} \times$$
$$\{\text{regular, explicit}\} \times \{\text{sequential, parallel}\}$$

A Benchmark



Parallel results (desktop computer): we are better!

By-products: Cycle Generators



Expanded 2-saddles boundaries → cycle generators

TDA Applications

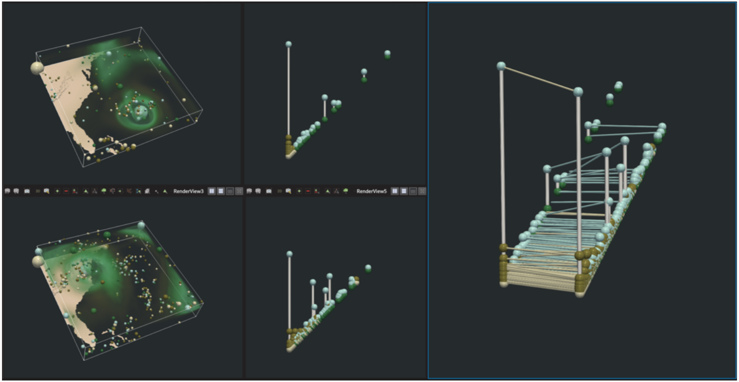
Ensembles Analysis (2019 – 2022)

The VESTEC Project

- European-funded research Project
- 9 academic & industrial partners (inc. Sorbonne Université)
- To build methods, infrastructure & interfaces for **Urgent Decision-Making**
- Using ensemble simulations on HPC clusters
- Using visualization software
  - ParaView, CosmoScout VR
- 3 Use-Cases
  1. Forest Fire (Tecnosylva, Spain)
  2. Mosquito-Borne Diseases (FBK, Italy)
  3. *Space Weather* (KTH, Sweden)

TTK Tools for Ensemble Analysis

Distance between Persistence Diagrams



Jules Vidal, Joseph Budin, and Julien Tierny. “Progressive Wasserstein Barycenters of Persistence Diagrams”. In: *IEEE Transactions on Visualization and Computer Graphics* (Oct. 2019). Accepted to IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS 2019). URL: `https://hal.archives-ouvertes.fr/hal-02179674`

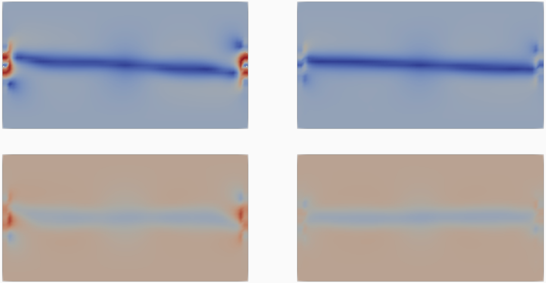
TTK’s Role in VESTEC

Analyze ensemble simulations

1. Generate persistence diagrams at every simulation cycle
2. At the end of the simulation, compute a distance matrix from all the diagrams
3. Use Dimension Reduction to reduce the distance matrix to a point cloud
4. Visualize & manipulate the results with ParaView
5. (Opt.) Cluster the persistence diagrams
6. (Opt.) Generate a Rips Complex to extract topological features

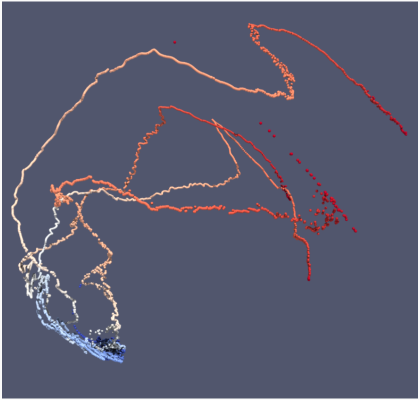
VESTEC Use-Case 3: Space Weather

- Particle-In-Cell simulator developed at KTH (Stockholm)
- Magnetic (vector) field in the Earth magnetosphere
- What's important: magnetic reconnection (instability phenomenon)
- One persistence diagram on the magnitude of the magnetic field $\times$ 2500 cycles $\times$ 4 simulations

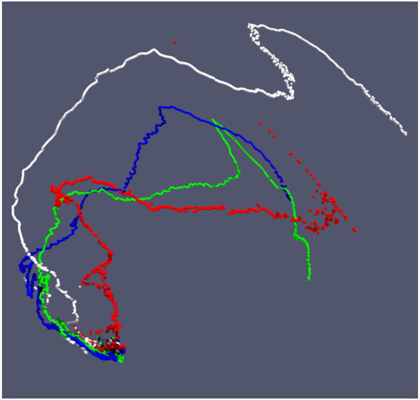


Four simulations, same cycle, different input parameters

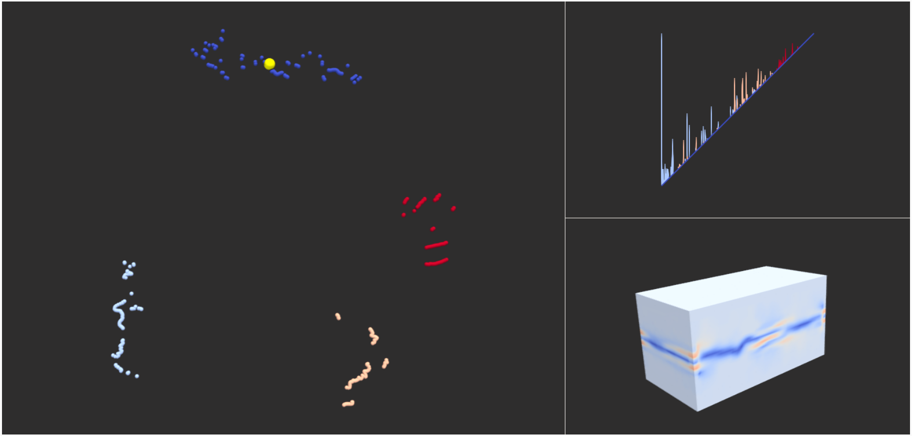
Simulation Results after Dimension Reduction



Color:
simulation cycle
(0 to 2500)



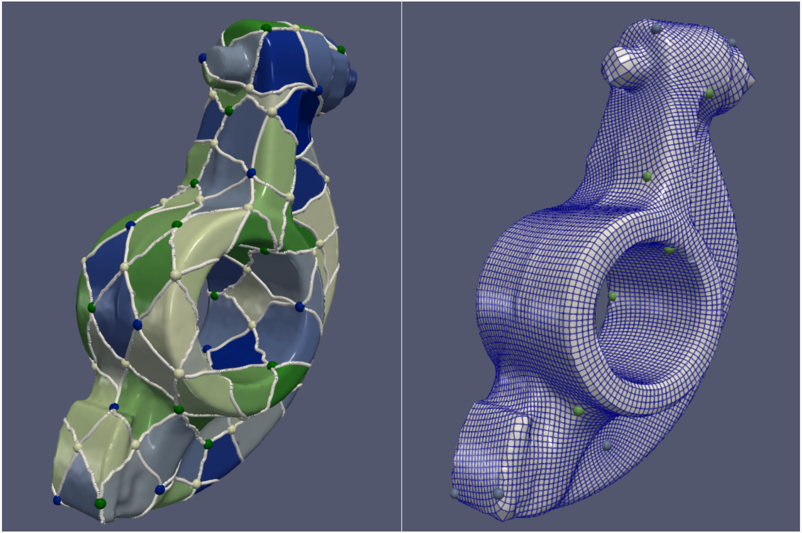
Color:
simulation parameters
(4 simulations)



TDA Applications

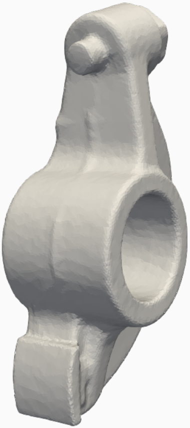
Surface Quadrangulation (2019)

Surface Quadrangulation using the Morse-Smale Complex



Pipeline

From a triangular, closed surface

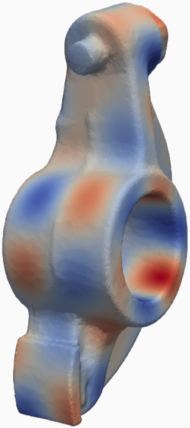


Shen Dong et al. “Spectral Surface Quadrangulation”. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1057–1066. ISSN: 0730-0301. DOI: 10.1145/1141911.1141993. URL: https://doi.org/10.1145/1141911.1141993

Pipeline

From a triangular, closed surface

1. we generate a scalar field that alternates the critical points on the surface

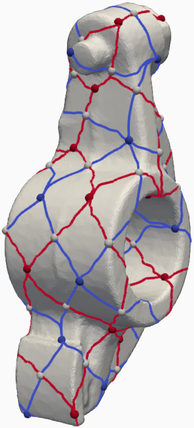


Shen Dong et al. “Spectral Surface Quadrangulation”. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1057–1066. ISSN: 0730-0301. DOI: 10.1145/1141911.1141993. URL: https://doi.org/10.1145/1141911.1141993

Pipeline

From a triangular, closed surface

1. we generate a scalar field that alternates the critical points on the surface
2. we compute the Morse-Smale Complex

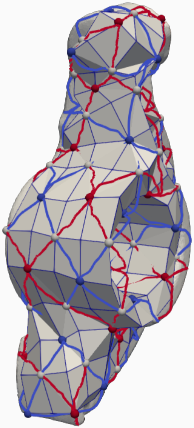


Shen Dong et al. “Spectral Surface Quadrangulation”. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1057–1066. ISSN: 0730-0301. DOI: 10.1145/1141911.1141993. URL: https://doi.org/10.1145/1141911.1141993

Pipeline

From a triangular, closed surface

1. we generate a scalar field that
   alternates the critical points on the surface
2. we compute the Morse-Smale Complex
3. coarse quadrangulation around
   the saddle points (4 neighbors)

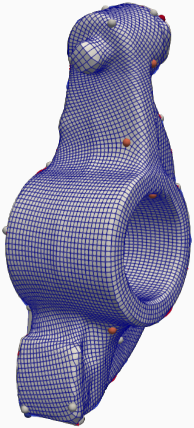


Shen Dong et al. “Spectral Surface Quadrangulation”. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1057–1066. ISSN: 0730-0301. DOI: 10.1145/1141911.1141993. URL: https://doi.org/10.1145/1141911.1141993

Pipeline

From a triangular, closed surface

1. we generate a scalar field that alternates the critical points on the surface
2. we compute the Morse-Smale Complex
3. coarse quadrangulation around the saddle points (4 neighbors)
4. subdivision then projection/relaxation iterations to refine the quadrangulation



Shen Dong et al. "Spectral Surface Quadrangulation". In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1057–1066. ISSN: 0730-0301. DOI: 10.1145/1141911.1141993. URL: https://doi.org/10.1145/1141911.1141993

Input Scalar Field

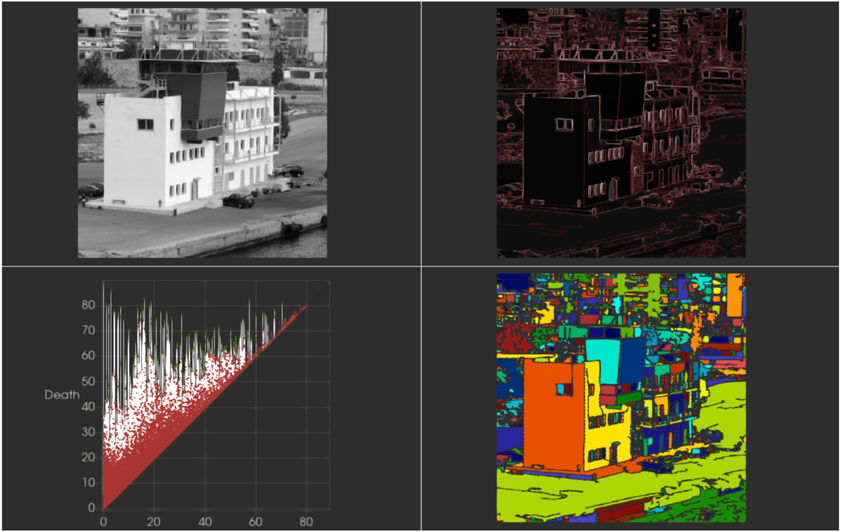
One eigenfunction of the triangulation laplacian

- Triangulation edges $\rightarrow$ adjacency relationship between vertices
- Laplacian matrix $=$ Degree matrix $-$ Adjacency matrix
- Use spectralib to get the eigenvectors associated with the highest eigenvalues (magnitude)
- An eigenvector $=$ a value per vertex $=$ a scalar field
  - Minima & maxima are well distributed on the input domain
  - Eigenvalue magnitude $\searrow$ #critical points $\nearrow$

TDA Applications

Image Segmentation

Image Segmentation using the Morse-Smale Complex



From a PNG image

Pipeline

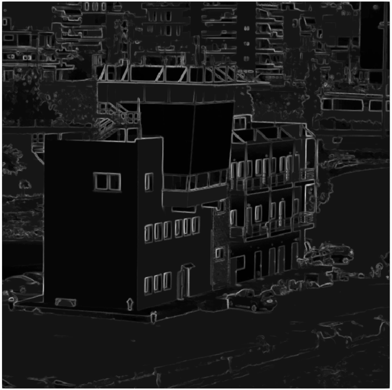
From a PNG image

1. ParaView computes the gradient

Pipeline

From a PNG image

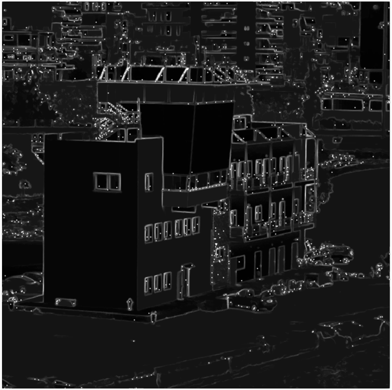
1. ParaView computes the gradient
2. topological Simplification



Pipeline

From a PNG image

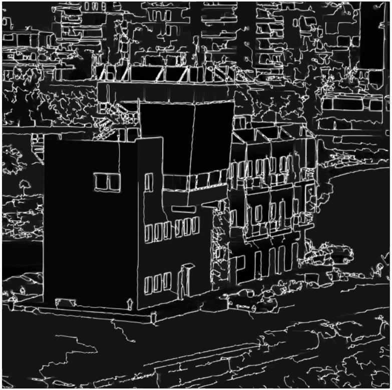
1. ParaView computes the gradient
2. topological Simplification
3. Morse-Smale Complex
    - the minima are the markers



Pipeline

From a PNG image

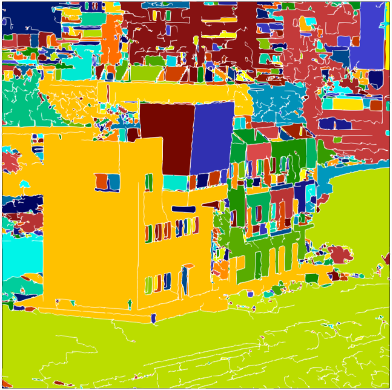
1. ParaView computes the gradient
2. topological Simplification
3. Morse-Smale Complex
   - the minima are the markers
   - the ascending separatrices are the boundaries



Pipeline

From a PNG image

1. ParaView computes the gradient
2. topological Simplification
3. Morse-Smale Complex
   - the minima are the markers
   - the ascending separatrices are the boundaries
   - the minima basins form the segmentation

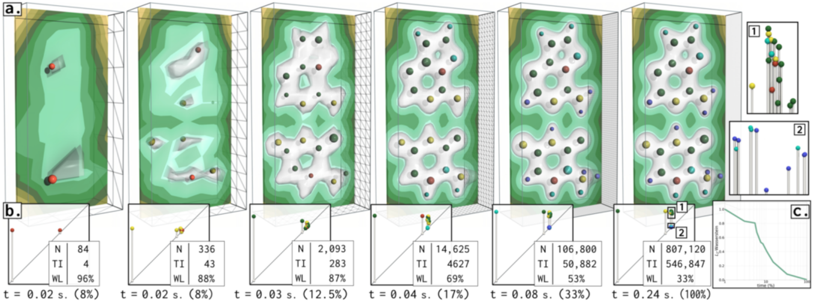


Software Engineering Work

Progressive Persistence Diagrams (2020)

Progressive Persistence Diagrams (2020)

Using a *hierarchical decimation* of regular grids



Jules Vidal, Pierre Guillou, and Julien Tierny. "A Progressive Approach to Scalar Field Topology". In: *IEEE Transactions on Visualization and Computer Graphics* 27.6 (June 2021), pp. 2833–2850. ISSN: 2160-9306. DOI: 10.1109/tvcg.2021.3060500. URL: http://dx.doi.org/10.1109/TVCG.2021.3060500

Contributions: **performance**, **timer**, **restart**

Software Engineering Work

MPI support in TTK’s Triangulation (2022)

MPI support in TTK's triangulation (2022)

ParaView auto distributes regular grids with MPI

Eve Le Guillou: on-going PhD thesis to distribute TTK algorithms

- first, the internal data structures (Triangulation) ✓
- then, *ScalarFieldSmoother*, *DiscreteGradient* ✓
- finally, *DiscreteMorseSandwich*...

Triangulation work

- local $\leftrightarrow$ global simplex identifiers
- regular grids: use a per-process virtual representation of the global grid
- explicit triangulations: enumerate edges & triangles inside contiguous (global) ranges of tetrahedra

Software Engineering Work

Performance & Quality Improvements

Performance Improvements

Morse-Smale Complex OpenMP

- Rework the (map-reduce) parallelism for the 2-Separatrices
- Speedup: $\times 4$ (GitHub PR)

Explicit Triangulation data structures

- all relationships between simplices are stored explicitly
  - ex: edges per triangle $\rightarrow$ `std::vector<std::vector<int>>`
- `std::vector<std::vector<int>>` used everywhere
  - non-cache friendly, lots of allocations/deallocations
- replace with `std::vector<std::array<int, N>>`
  - one contiguous cache-friendly memory block
  - ex: edges per triangle $\rightarrow$ `std::vector<std::array<int, 3>>`
  - Discrete Gradient speedups from +30% to +90% (GitHub PR)
- something similar can be done to the non-fixed relationships
  - ex: number of neighbors per vertex

Code Quality Improvements

GitHub Actions workflows

- first, for generating binary packages
  - Ubuntu `.deb`
  - Windows installers
- then, to test the build at each PR (Ubuntu, macOS, Windows)
  - with `ccache`/`sccache` to cache the build artifacts
- then, to test the state files from `ttk-data`
- use tools to maintain a high-quality source code
  - `clang-format` makes the code uniformly readable
  - `clang-check` quickly checks if the code compiles in a variety of configurations (MPI, OpenMP, Debug vs Release)
  - `clang-tidy` enforces more complex rules (static analysis)

Conclusion

Conclusion

TTK (and TDA in general) provides useful tools to help understanding scalar field on meshes

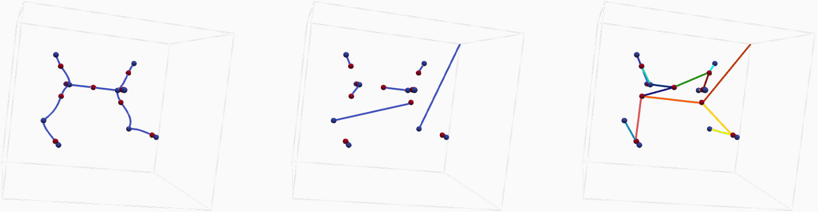
- Topology-preserving reduced representations
  - persistence diagrams
- Statistical analyses on these reduced representations
  - distance, clustering, dimension reduction
- Easy manipulations & visualizations
  - ParaView integration
- Various applications
  - ensemble analysis, quadrangulation, image segmentation...
- Great performance & code quality
  - hopefully it remains the same after my departure...

Other topological abstractions

- **Merge trees** “augmented” persistence diagrams with parent relationships between pairs: *distance, clustering, geodesics*

On-going work

- **Distribution** (MPI) of the algorithms
- Include **Machine-Learning** methods in our pipelines



Morse-Smale separatrices — Embedded persistence diagram — Join tree (Merge tree)

Topological Data Analysis with TTK

A Four Years Post-Doc at LIP6

Pierre Guillou

Fontainebleau, March 20th, 2023

CNRS & Sorbonne Université

References

[1] Ulrich Bauer. *Ripser: efficient computation of Vietoris-Rips persistence barcodes.* https://github.com/Ripser/ripser/. 2019. eprint: 1908.02518.

[2] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. “Distributed computation of persistent homology”. In: *Alg. Eng. & Exp.* 2014.

[3] Ulrich Bauer et al. “PHAT - Persistent Homology Algorithms Toolbox”. In: *J. Symb. Comput.* (2017).

[4] Talha Bin Masood et al. “An Overview of the Topology ToolKit”. In: *TopoInVis*. 2019.

[5] O. Delgado-Friedrichs et al. *Digital image analysis using discrete Morse theory and persistent homology*. `https://github.com/AppliedMathematicsANU/diamorse/`. Accessed: 2021-05-18. 2020.

[6] Shen Dong et al. “Spectral Surface Quadrangulation”. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1057–1066. ISSN: 0730-0301. DOI: `10.1145/1141911.1141993`. URL: `https://doi.org/10.1145/1141911.1141993`.

[7] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. “Topological Persistence and Simplification”. In: *DCG* (2002).

[8] Charles Gueunet et al. “Task-Based Augmented Contour Trees with Fibonacci Heaps”. In: *TPDS* (2019).

[9] Pierre Guillou, Jules Vidal, and Julien Tierny. *Discrete Morse Sandwich: Fast Computation of Persistence Diagrams for Scalar Data – An Algorithm and A Benchmark*. 2022. DOI: 10.48550/ARXIV.2206.13932. URL: https://arxiv.org/abs/2206.13932.

[10] Gregory Henselman-Petrusek. *Eirene.jl package for Homological Algebra*. https://github.com/Eetion/Eirene.jl. Accessed: 2021-05-19. 2018.

[11] Federico Iuricich. "Persistence cycles for visual exploration of persistent homology". In: *TVCG* (2021). https://github.com/IuricichF/PersistenceCycles.

[12] Shizuo Kaji, Takeki Sudo, and Kazushi Ahara. *Cubical Ripser: Software for computing persistent homology of image and volume data.* `https://github.com/CubicalRipser/`. 2020.

[13] Jonas Lukasczyk et al. “Localized Topological Simplification of Scalar Data”. In: *IEEE Transactions on Visualization and Computer Graphics* (Oct. 2020). URL: `https://hal.archives-ouvertes.fr/hal-02949278`.

[14] Clément Maria et al. “The Gudhi Library: Simplicial Complexes and Persistent Homology”. In: *Mathematical Software.* `https://github.com/GUDHI/`. 2014.

[15] Dmitriy Morozov. *Dionysus2.* `http://www.mrzv.org/software/dionysus2`. Accessed: 2021-05-19. 2017.

[16] Vidit Nanda. *Perseus, the Persistent Homology Software.* `http://people.maths.ox.ac.uk/nanda/perseus/`. Accessed: 2021-05-18. 2021.

[17] Arnur Nigmetov and Dmitriy Morozov. *Oineus*. `https://github.com/grey-narn/oineus`. Accessed: 2021-05-19. 2021.

[18] Vanessa Robins, Peter John Wood, and Adrian P. Sheppard. "Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011), pp. 1646–1658. DOI: `10.1109/TPAMI.2011.95`.

[19] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. “JavaPlex: A research software package for persistent (co)homology”. In: *ICMS*. http://appliedtopology.github.io/javaplex/. 2014.

[20] Julien Tierny and Valerio Pascucci. “Generalized Topological Simplification of Scalar Fields on Surfaces”. In: *IEEE Transactions on Visualization and Computer Graphics* (Dec. 2012). URL: https://hal.archives-ouvertes.fr/hal-01206877.

[21] Julien Tierny et al. “The Topology ToolKit”. In: *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS)* (2017).

[22] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. "Ripser.py: A Lean Persistent Homology Library for Python". In: *The Journal of Open Source Software* 3.29 (Sept. 2018), p. 925. DOI: `10.21105/joss.00925`. URL: `https://doi.org/10.21105/joss.00925`.

[23] Jules Vidal, Joseph Budin, and Julien Tierny. "Progressive Wasserstein Barycenters of Persistence Diagrams". In: *IEEE Transactions on Visualization and Computer Graphics* (Oct. 2019). Accepted to IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS 2019). URL: `https://hal.archives-ouvertes.fr/hal-02179674`.

[24] Jules Vidal, Pierre Guillou, and Julien Tierny. "A Progressive Approach to Scalar Field Topology". In: *IEEE Transactions on Visualization and Computer Graphics* 27.6 (June 2021), pp. 2833–2850. ISSN: 2160-9306. DOI: 10.1109/tvcg.2021.3060500. URL: http://dx.doi.org/10.1109/TVCG.2021.3060500.

[25] Afra J. Zomorodian. "Topology for Computing". In: *Algorithms and Theory of Computation Handbook (Second Edition)*. Ed. by Mikhail J. Atallah and Marina Blanton. CRC Press, 2010. Chap. 3, pp. 82–112.