Applications of Algebraic Topology

Mikael Vejdemo-Johansson

August 14, 2011

Outline

Wherefore topology?

Topology gives qualitative information

Topology is about geometric properties not dependent on a metric:

How many pieces? Are there holes? Bubbles? Can you get turned around while moving?

Topology captures continuity, connectedness, nearness

In particular, the non-dependence on a metric helps if metrics are ill-motivated, eg phylo-genetics.

From geometry to algebra

Nice enough spaces can be triangulated: decomposed into vertices, edges, triangles, and higher-order *simplices*.

Chain complexes

From a triangulated space $X = X_0 \sqcup X_1 \sqcup \cdots \sqcup X_n$, we construct a vector space $CX = k^{|X_0|} \oplus k^{|X_1|} \oplus \cdots \oplus k^{|X_n|}$ with a basis element for each building block of X.

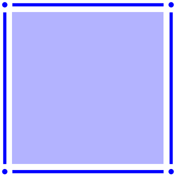
Mapping a simplex to (a signed sum of) its boundary components yields a linear map $\partial : CX \to CX$.

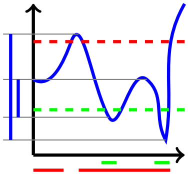
Done right, this map encodes geometry: closed circuits corresponds to the null space, filled in closed circuits to the image space.

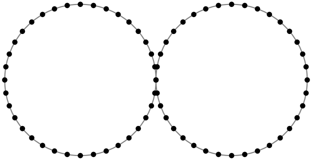
Applying algebraic topology

Three major genres:

1. Cubical methods: homology of collections of pixels/voxels. Useful for bitmaps et c. Very efficient in 2-3 dimensions, falls of radically with dimensionality.
2. Persistence:
   - 2.1 Function scheme. Function $M \to \mathbb{R}$ on manifold. Measure topology of sublevel sets.
   - 2.2 Point clouds. Collection $X \subset \mathbb{R}^d$. Measure topology of object X is sampled from.







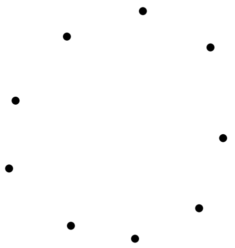
Outline

Algebraic topology of point clouds

Definition

The Vietoris-Rips complex is an abstract simplicial complex $VR_\epsilon(X)$ for $\epsilon \in \mathbb{R}_+$ and X a finite metric space:

- Contains one vertex for each element in X.
- Contains a simplex $(x_0, \ldots, x_k)$ exactly when $d(x_i, d_j) < \epsilon$ for all $i, j \in [k]$.

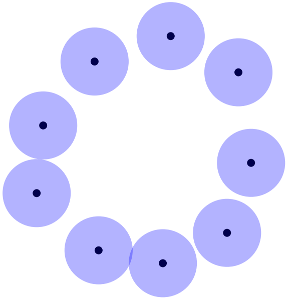


Algebraic topology of point clouds

Definition

The Vietoris-Rips complex is an abstract simplicial complex $VR_\epsilon(X)$ for $\epsilon \in \mathbb{R}_+$ and X a finite metric space:

- Contains one vertex for each element in X.
- Contains a simplex $(x_0, \ldots, x_k)$ exactly when $d(x_i, d_j) < \epsilon$ for all $i, j \in [k]$.

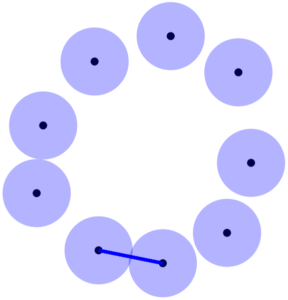


Algebraic topology of point clouds

Definition

The Vietoris-Rips complex is an abstract simplicial complex $VR_\epsilon(X)$ for $\epsilon \in \mathbb{R}_+$ and X a finite metric space:

- Contains one vertex for each element in X.
- Contains a simplex $(x_0, \ldots, x_k)$ exactly when $d(x_i, d_j) < \epsilon$ for all $i, j \in [k]$.

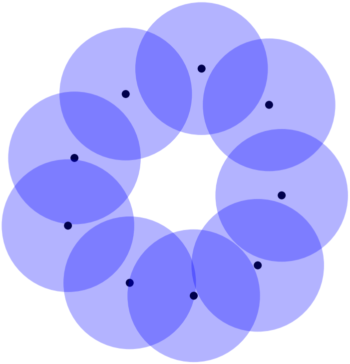


Algebraic topology of point clouds

Definition

The Vietoris-Rips complex is an abstract simplicial complex $VR_\epsilon(X)$ for $\epsilon \in \mathbb{R}_+$ and X a finite metric space:

- Contains one vertex for each element in X.
- Contains a simplex $(x_0, \ldots, x_k)$ exactly when $d(x_i, d_j) < \epsilon$ for all $i, j \in [k]$.

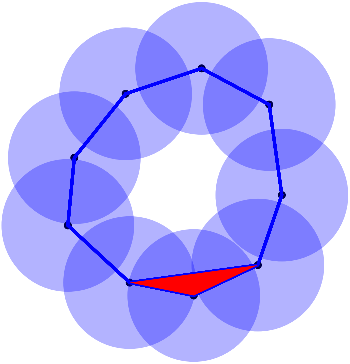


Algebraic topology of point clouds

Definition

The Vietoris-Rips complex is an abstract simplicial complex $VR_\epsilon(X)$ for $\epsilon \in \mathbb{R}_+$ and X a finite metric space:

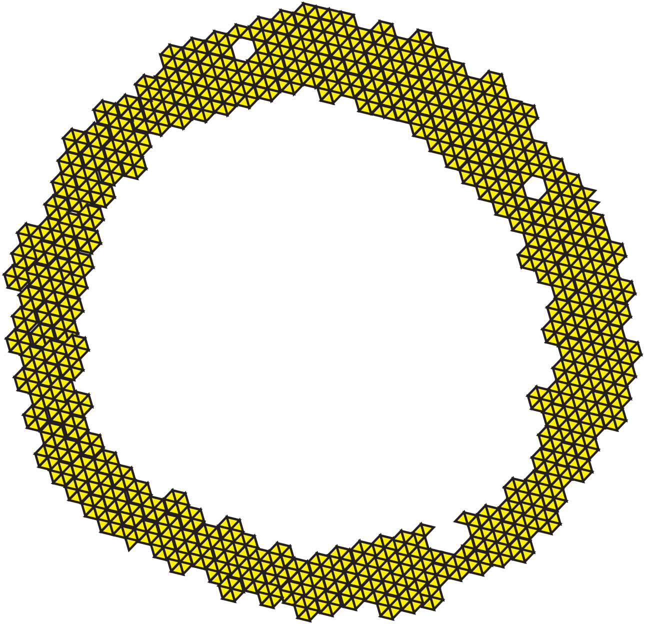
- Contains one vertex for each element in X.
- Contains a simplex $(x_0, \ldots, x_k)$ exactly when $d(x_i, d_j) < \epsilon$ for all $i, j \in [k]$.



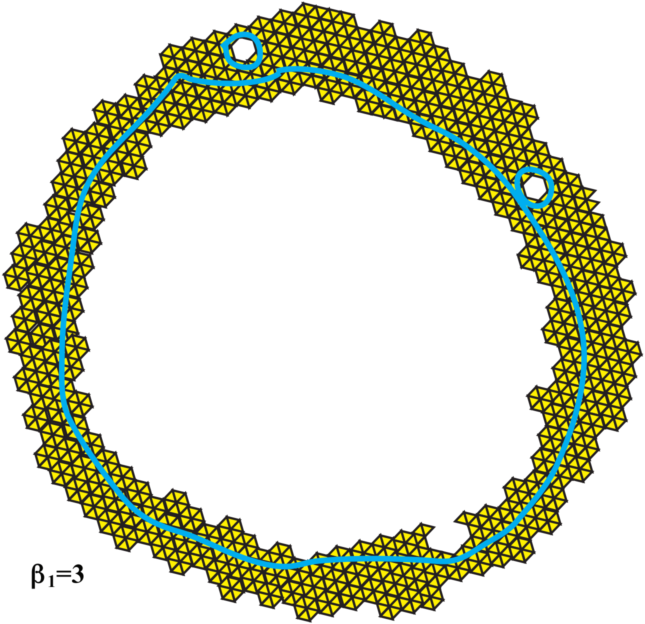
How do we choose parameters?

Wrong question: we could pick a parameter, but it would be less illustrative. Right answer is to use *persistence*. Grow the parameter, and measure *timespans* for each topological feature.

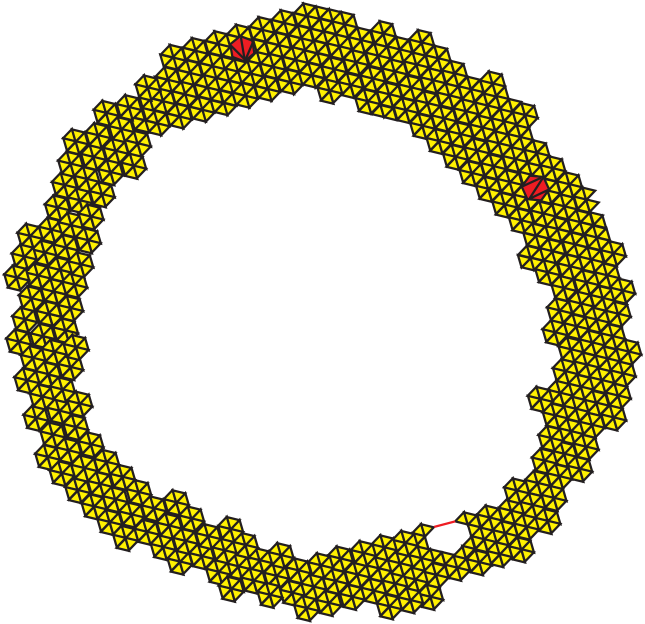
How do we choose parameters?



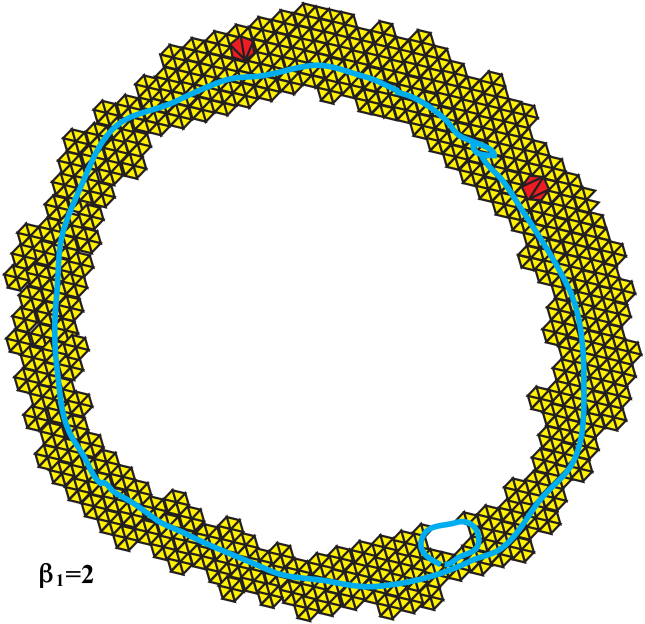
How do we choose parameters?



How do we choose parameters?



How do we choose parameters?



$\beta_1 = 2$

Outline

Plex, jPlex, javaPlex

All developed by the Stanford group.

Plex

- Discontinued.
- Consumes a lot of memory.
- C++ / MEX, with Matlab interface.

jPlex

- Discontinued.
- Highly optimized for computing barcodes.
- Difficult to extend.
- Java, with Matlab interface.

javaPlex

- Recently developed.
- Easy to extend.
- Java, with Matlab interface.

`http://comptop.stanford.edu`

`http://code.google.com/p/javaplex`

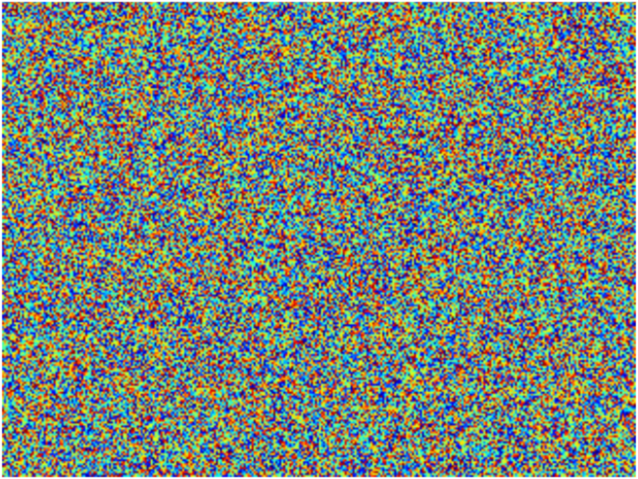
Dionysus

- Alternative implementation by Dmitriy Morozov.
- C++/Python. Python interface.
- High-speed computation and easy to extend.
- Implements more techniques than Plex/jPlex/javaPlex.
- Library, less focused on ease-of-use.

`http://www.mrzv.org/software/dionysus/`

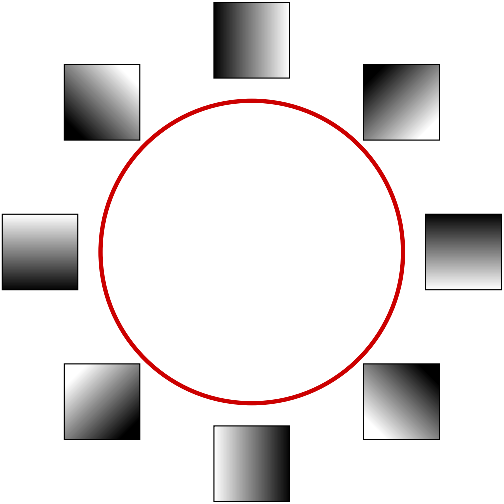
Outline

Klein bottle of natural images



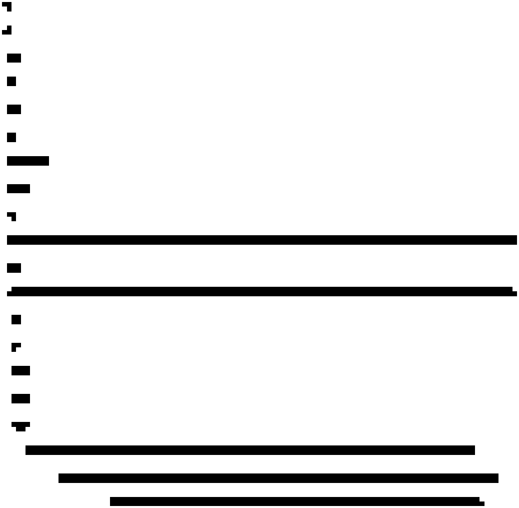
Klein bottle of natural images

Klein bottle of natural images

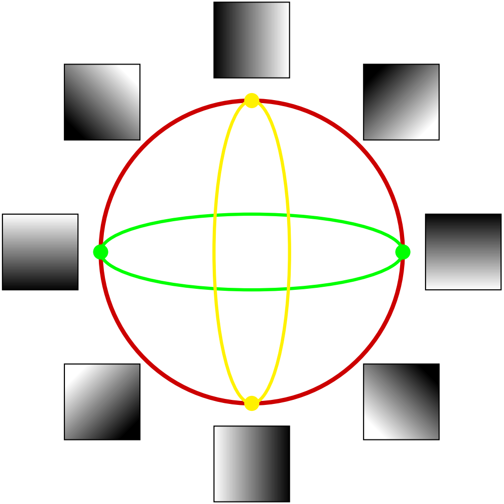


Klein bottle of natural images

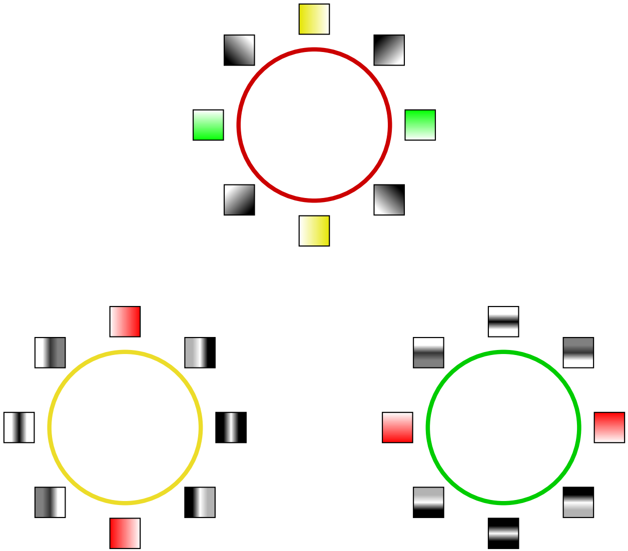
Klein bottle of natural images



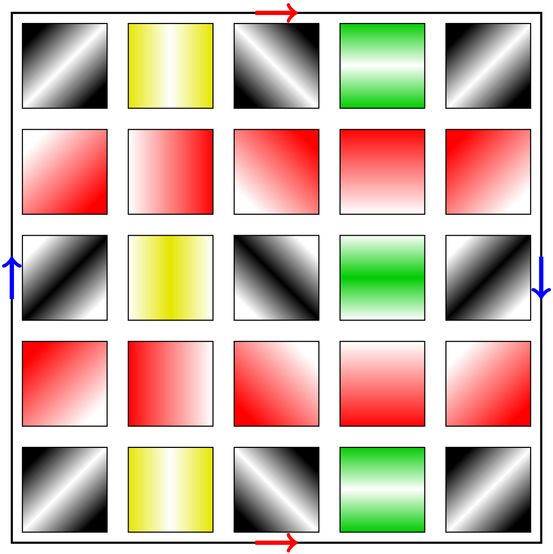
Klein bottle of natural images



Klein bottle of natural images



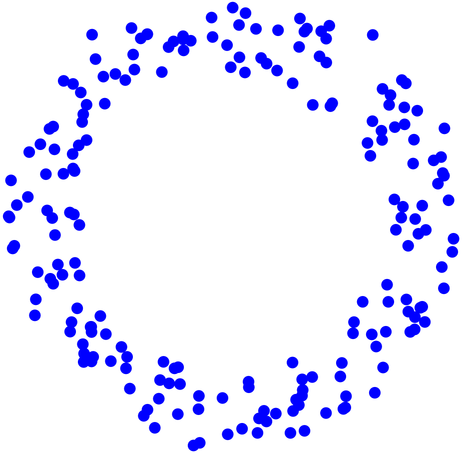
Klein bottle of natural images



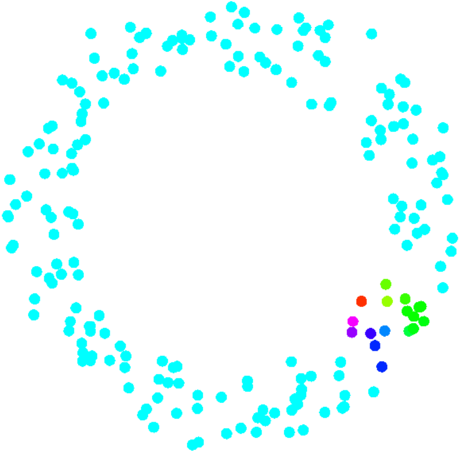
Periodicity and persistent cohomology

- Fact from topology: $[X; S^1] = H^1(X, \mathbb{Z})$.
- Morozov – de Silva – Vejdemo-Johansson (2009) demonstrates how to compute $H^1(X, \mathbb{Z})$ persistently.
- Yields *coordinate functions* $X \to S^1$ with circle-valued coordinates.
- Current research project into how to apply this to analyzing periodic systems.
- Example: unrolling a circular coordinate, followed by linear regression, measures periodicity.

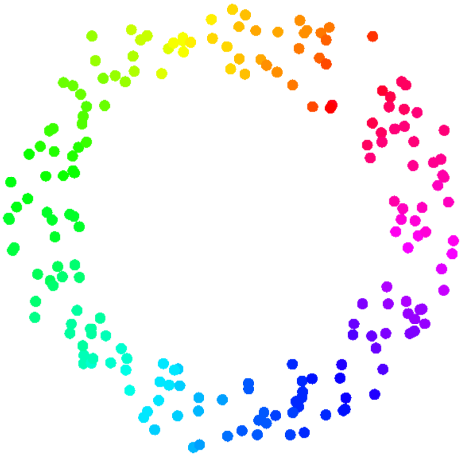
Persistence shows relevance of coordinate



Persistence shows relevance of coordinate



Persistence shows relevance of coordinate



Configuration spaces of linkages

Preliminary from a project joint with J. Hauenstein (Texas A&M), D. Eklund (KTH), M. Scolamieri (KTH), P. Skraba (Joszef Stefan institute, Ljubljana), and C. Peterson (Colorado State University).

Numerical algebraic geometry

Homotopy continuation solves, numerically, systems of polynomial equations. In effect sampling algebraic varieties (solution spaces), such as configuration spaces of linkages.

Persistence

Persistent homology can estimate topological features of point clouds – such as these samples.

Mayer-Vietoris can piece together easy-to-compute pieces and stretch computational capacity.

Configuration spaces of linkages

Proposed workflow

1. Generate a point cloud of valid configurations.
2. Find (w/ algebraic geometry) singular points.
3. Remove singular points and their neighbours.
4. Compute topological description of the singular locus.
5. Compute topological description of each connected component (cluster) of the non-singular points.
6. Merge the information from these steps into a topological description of the entire structure.

Example

Configuration space of cyclo-octane has been found to be a Klein bottle, union with a sphere, intersecting in two disjoint circles.

Questions?