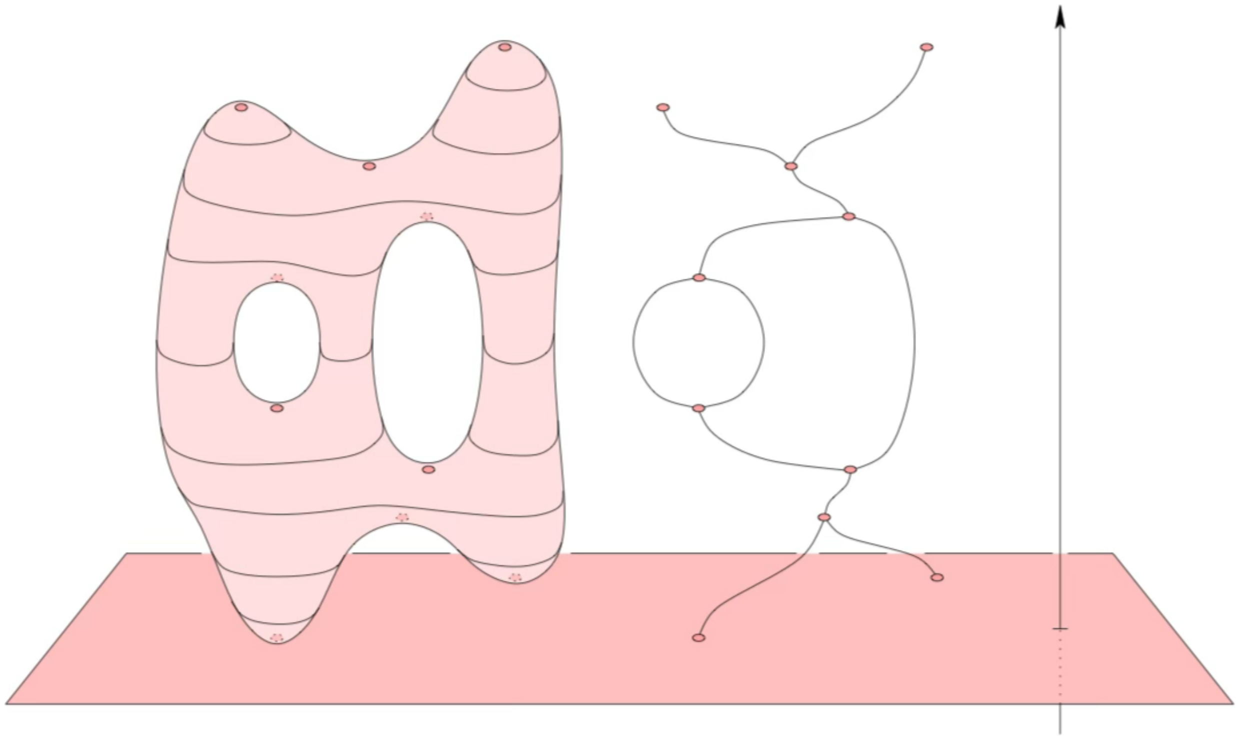
MATH412/COMPSCI434/MATH713
Fall 2025
Topological Data Analysis

Topic 9: Reeb graph, contour trees

Instructor: Ling Zhou

Reeb graphs

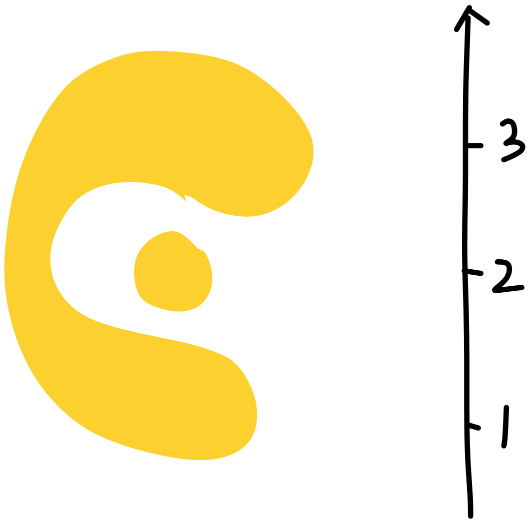
Reeb graph: applications of function-induced persistence



Data		Sublevel Set		Data		Reeb graph		Sublevel Set
(X, f)	$\longrightarrow$	$f^{-1}((-\infty, t])$		(X, f)	$\longrightarrow$	(R_f, f)	$\longrightarrow$	$f^{-1}((-\infty, t])$

Reeb Graph

- Given a topological space X and function $f: X \to R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a



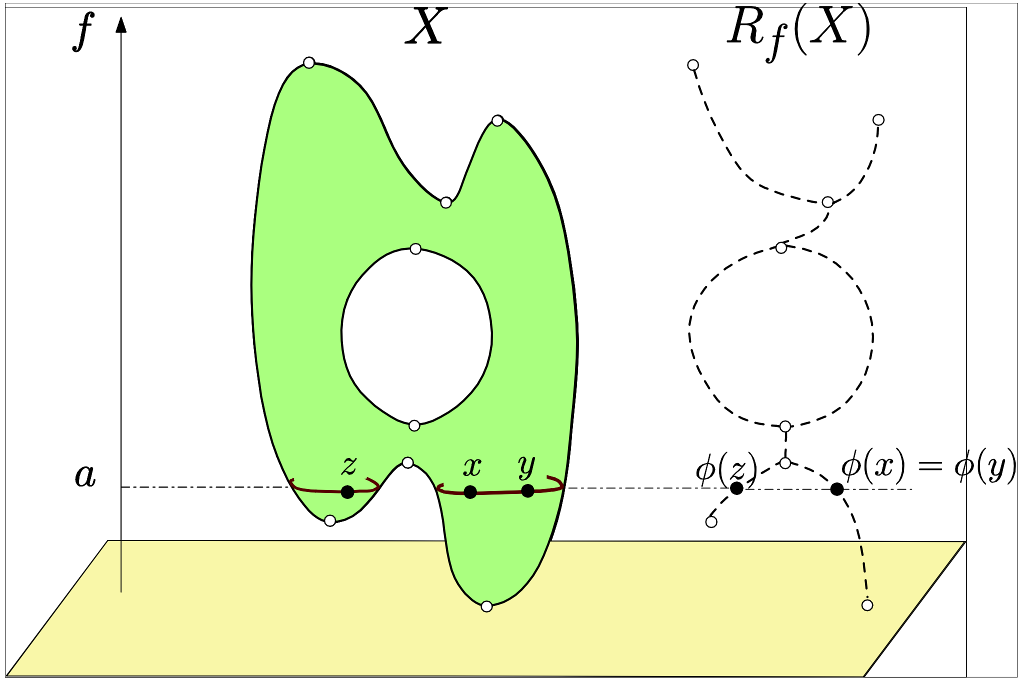
X_1 has 1 contour

X_2 has 2 contours

X_3 has 1 contours

Reeb Graph

- *Reeb graph* $R_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \to R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour



$R_f = X/\sim$, where $x \sim y$ iff

(1) $f(x) = f(y) = a$

(2) x & y connected in $f^{-1}(\{a\})$

Reeb Graph and similar constructions

Reeb graph

$R_f = X/\sim$, for $x \sim y$ iff

(1) $f(x) = f(y) = a$

(2) x & y connected in $f^{-1}(\{a\})$

Join/Merge tree

$T_J = X/\sim$, for $x \sim y$ iff

(1) $f(x) = f(y) = a$

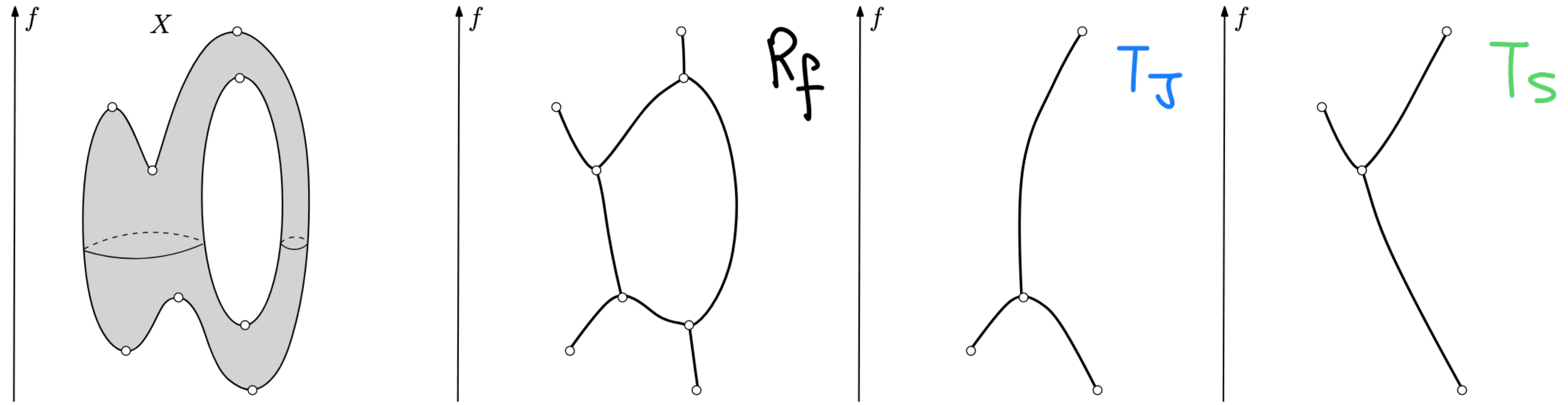
(2) x & y connected in $f^{-1}((-\infty, a])$

Split tree

$T_S = X/\sim$, for $x \sim y$ iff

(1) $f(x) = f(y) = a$

(2) x & y connected in $f^{-1}([a, \infty))$



(a) Input scalar field (b) Reeb graph (c) Merge tree Split tree

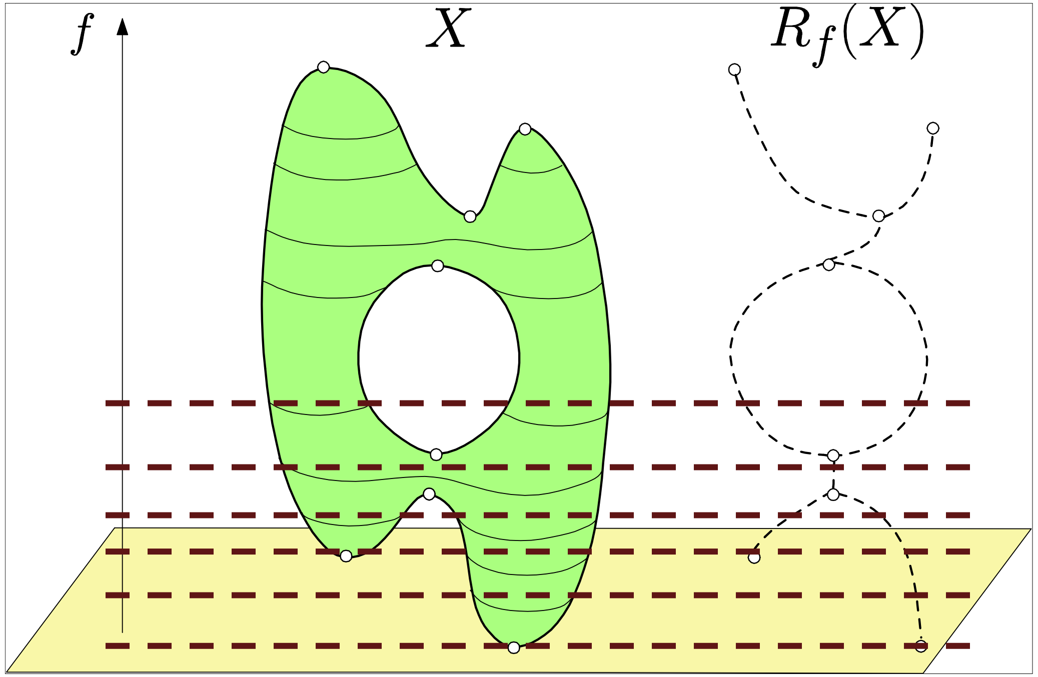
More on Reeb Graph

- Reeb graph $R_f(X)$ is an abstract graph
- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour

Lemma

$$\beta_0(R_f(X)) = \beta_0(X)$$

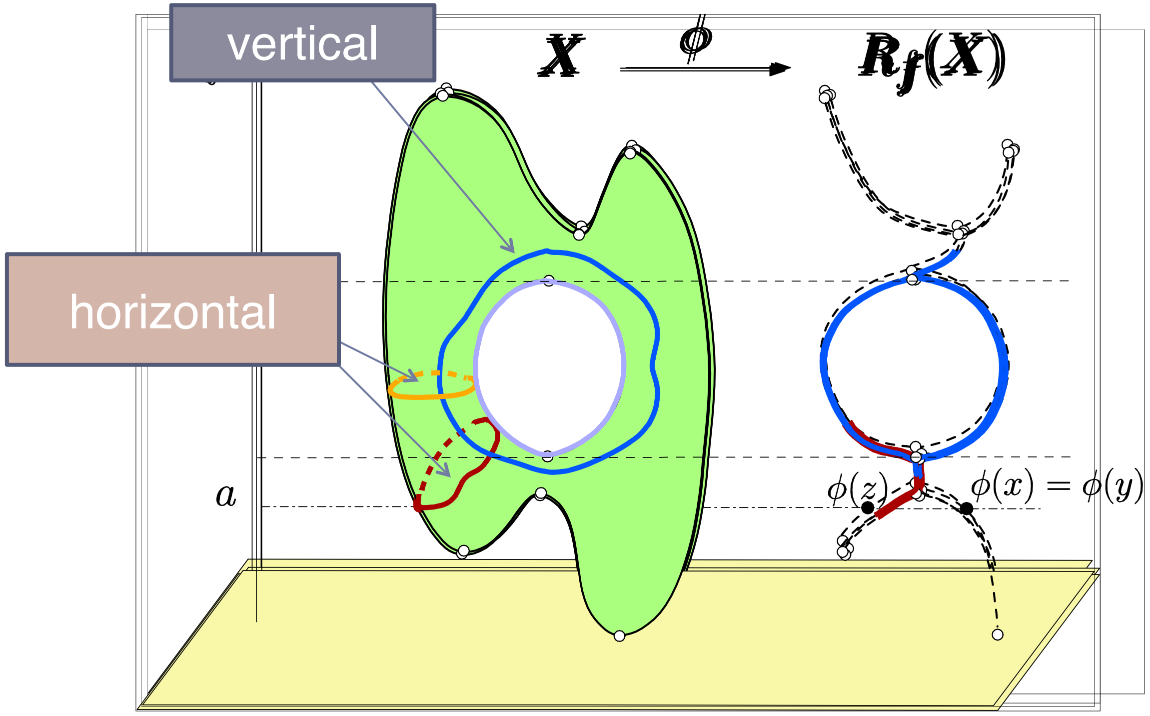
$$\beta_1(R_f(X)) \leq \beta_1(X)$$



More on Reeb Graph

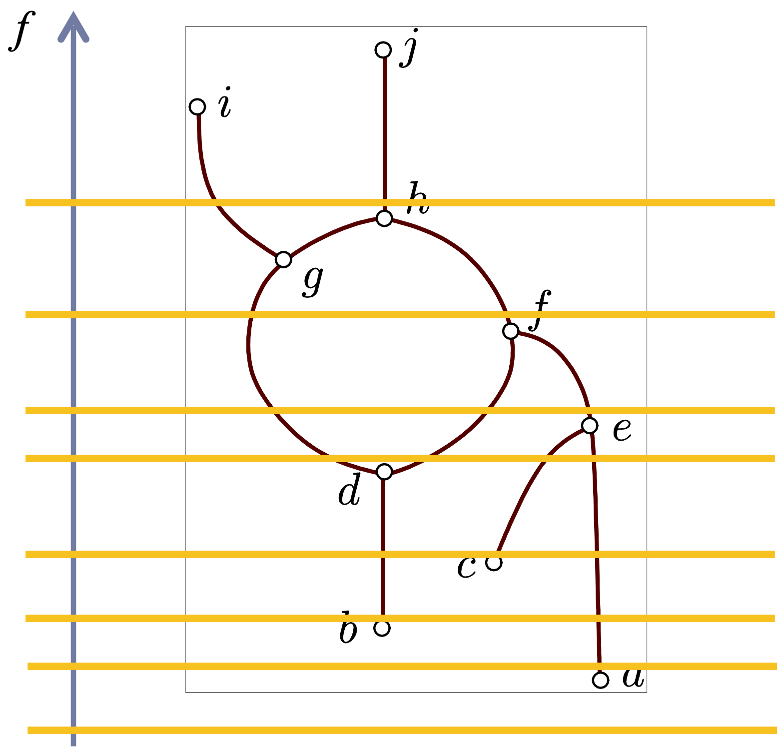
In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

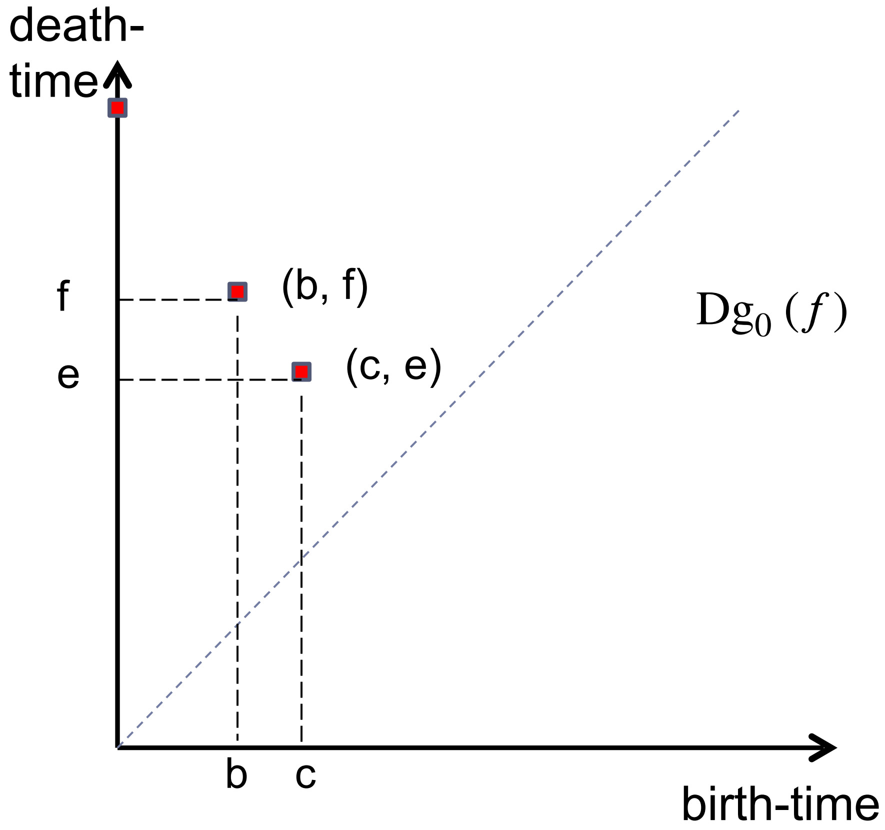
- Vertical holes are holes that survive after being mapped to the Reeb graph
- See Chapter 7 of textbook for more details



*[Dey and **W**, DCG2012]*

Branching Features in Reeb Graphs





- This uses dim-o information. How about higher dimensions?
- Level sets do not orm a filtration. To computing higher dimensions, we need new tools.

Remark: Extended Persistence

- Let $a_1 < \cdots < a_m$ be the homological critical values of $f: X \to R$, i.e. where the homology of the level set changes.
- Let $X_{\leq a_i} := f^{-1}((-\infty, a_i])$ be the sublevel set and $X^{\leq a_i} := f^{-1}([a_i, \infty))$be the superlevel set.
- **Extended persistence sequence:**
  - $\emptyset = H_p\left(X_{\leq a_1}\right) \to \cdots \to H_p\left(X_{\leq a_m}\right) = H_p(X) = H_p(X; X^{\geq a_m})$

$\to H_p\left(X; X^{\geq a_{m-1}}\right) \to \cdots \to H_p\left(X; X^{\geq a_2}\right) \to H_p\left(X; X^{\geq a_1}\right) = \emptyset$

Remark: Extended Persistence

Theorem 4.16. *Let $\mathcal{K}$ and $\mathcal{E}$ denote the simplicial levelset zigzag filtration and the extended filtration of a PL-function $f : |K| \to \mathbb{R}$. Then, for every $p \geq 0$,*

1. $[a_i, a_j)$ *is a bar for* $\mathrm{Dgm}_p(\mathcal{K})$ *iff it is a bar for* $\mathrm{Dgm}_p(\mathcal{E})$,
2. $(a_i, a_j]$ *is a bar for* $\mathrm{Dgm}_p(\mathcal{K})$ *iff* $[a_{n+j}, a_{n+i})$ *is a bar for* $\mathrm{Dgm}_{p+1}(\mathcal{E})$,
3. $[a_i, a_j]$ *is a bar for* $\mathrm{Dgm}_p(\mathcal{K})$ *iff* $[a_i, a_{n+j})$ *is a bar for* $\mathrm{Dgm}_p(\mathcal{E})$,
4. (a_i, a_j) *is a bar for* $\mathrm{Dgm}_p(\mathcal{K})$ *iff* $[a_j, a_{n+i})$ *is a bar for* $\mathrm{Dgm}_{p+1}(\mathcal{E})$.

Remark: Extended Persistence

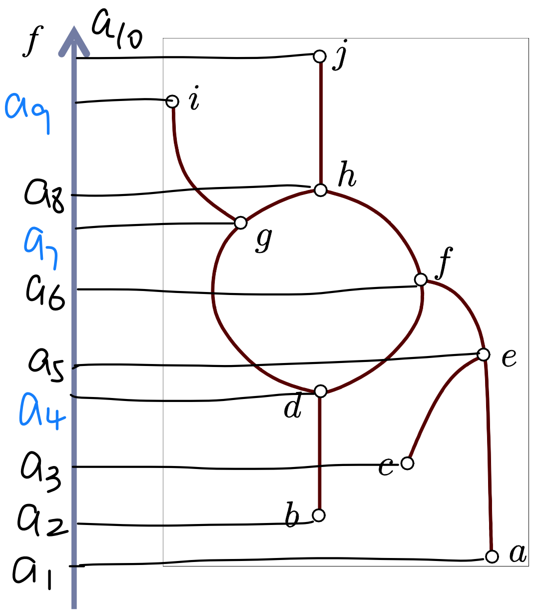
- Given a function $f: X \to R$ together with
  - an index set $I = \{ a_1 \le a_2 \ldots a_m \}$ with $a_1 < f_{min}, \quad a_m > f_{max}$
- Extended persistence sequence
  - $\emptyset = H_p\left(X_{\le a_1}\right) \to \cdots \to H_p\left(X_{\le a_m}\right) = H_p(X) = H_p(X; X^{\ge a_m})$

$\to H_p\left(X; X^{\ge a_{m-1}}\right) \to \cdots \to H_p\left(X; X^{\ge a_2}\right) \to H_p\left(X; X^{\ge a_1}\right) = \emptyset$

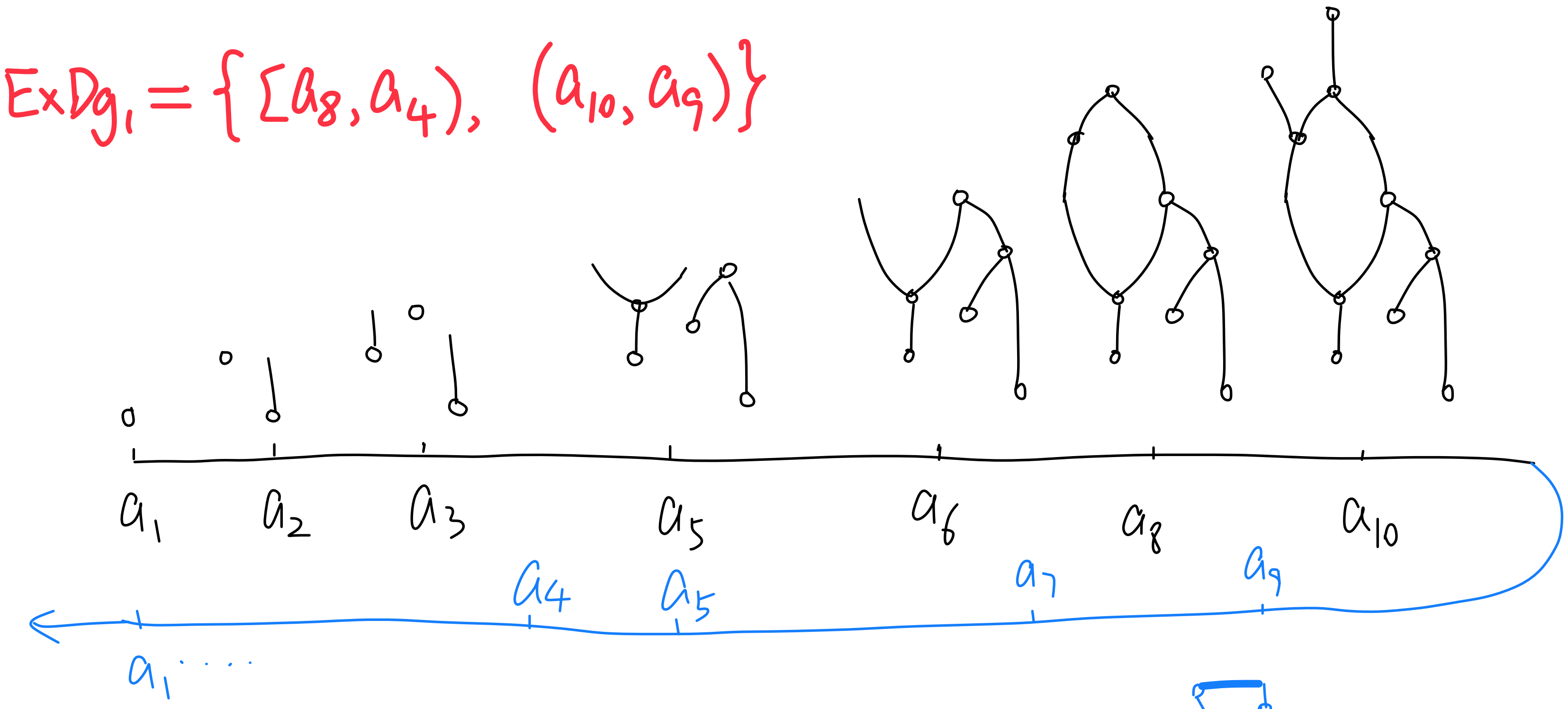
$H_p(X;Y)$ is the relative homology of X w.r.t. $Y \subset X$.

[You may think of $H_p(X;Y)$ as $H_p(X/Y)$ for $p \ge 1$]

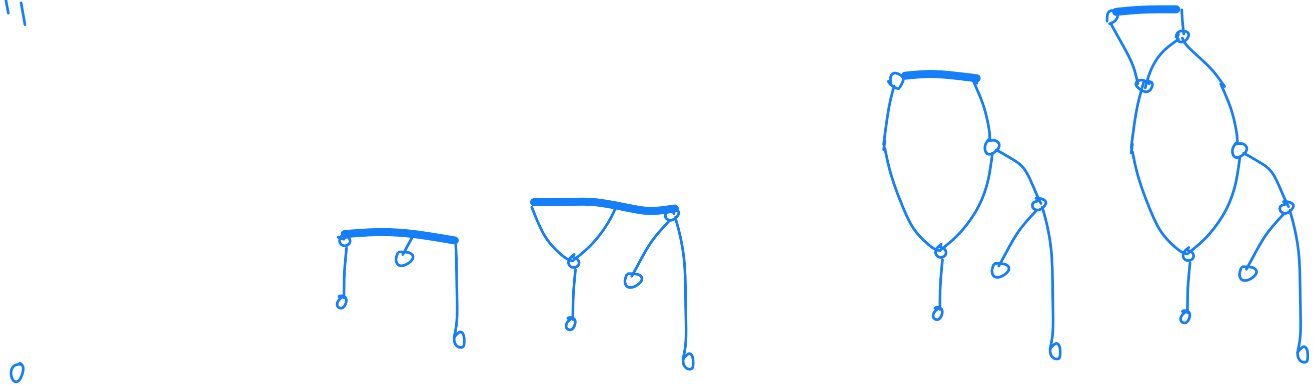
Remark: Extended Persistence



$$\mathrm{ExDg}_1 = \{ [a_8, a_4),\ (a_{10}, a_9) \}$$

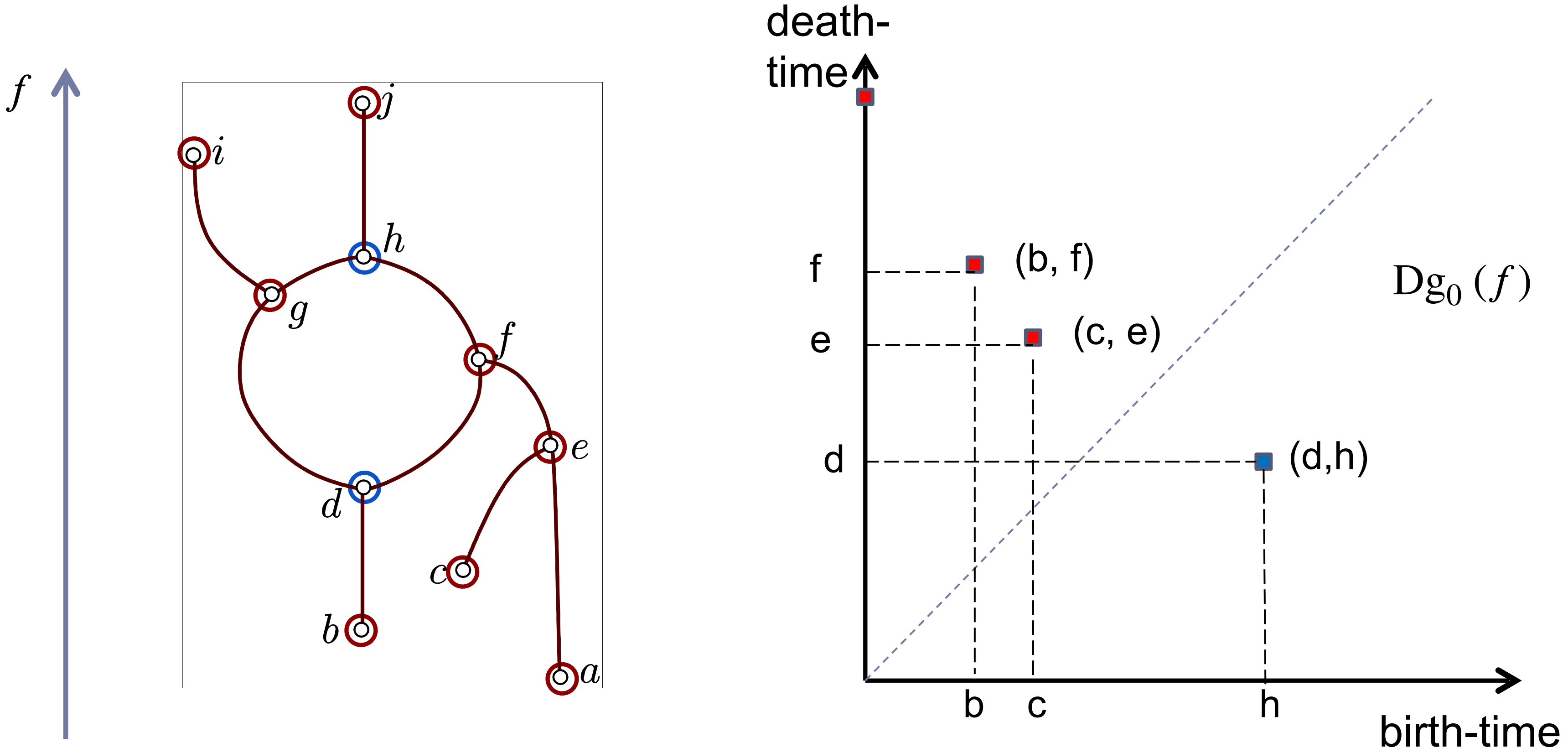


$X / f^{-1}([a,\infty))$:
identify $f^{-1}([a,\infty))$
to a point



Features in Reeb Graphs

- Branching features: Captured by persistence pairing induced by sub- / super-level set filtration
- Loop features: Captured by persistence pairing induced by level-set filtration



Distance between Reeb graphs

Definition 7.8 (Functional distortion distance)**.** Given two Reeb graphs $(\mathbb{F}, f)$ and $(\mathbb{G}, g)$, and a pair of continuous maps $\Phi : \mathbb{F} \to \mathbb{G}$ and $\Psi : \mathbb{G} \to \mathbb{F}$, set

$$C(\Phi, \Psi) = \{(x, y) \in \mathbb{F} \times \mathbb{G} \mid \Phi(x) = y, \text{or } x = \Psi(y)\}$$

and

$$D(\Phi, \Psi) = \sup_{(x,y),(x',y') \in C(\Phi,\Psi)} \frac{1}{2} \left| \mathsf{d}_f(x, x') - \mathsf{d}_g(y, y') \right| .$$

(Note that $C(\Phi, \Psi)$ defines a correspondence between $\mathbb{F}$ & $\mathbb{G}$).

The *functional distortion distance* between $(\mathbb{F}, f)$ and $(\mathbb{G}, g)$ is defined as:

$$\mathsf{d}_{FD}(\mathbb{F}, \mathbb{G}) = \inf_{\Phi,\Psi} \max\{ D(\Phi, \Psi), \|f - g \circ \Phi\|_\infty, \|g - f \circ \Psi\|_\infty \}. \tag{7.2}$$

Stability / Informative

- Stability Theorem: Under *some mild conditions* on $f,\ g\colon X \to R$,
  $D_{FD}\left(R_f,\ R_g \right) \leq \left\| f - g \right\|_\infty$

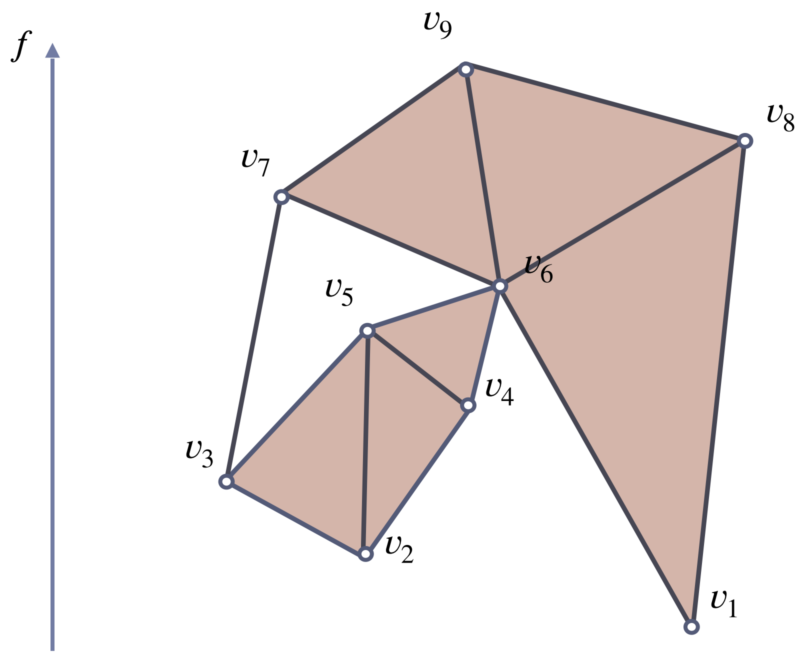
- Discrimination Theorem:
  - $d_B\left(\mathrm{Dg}_0\left(f\right),\ \mathrm{Dg}_0\left(g\right)\right) \leq D_{FD}(R_f,\ R_g)$
  - $d_B\left(exD_1\left(f\right),\ exD_1\left(g\right)\right) \leq 3D_{FD}\left(R_f,\ R_g\right)$

- See Chapter 7 of the textbook for more results, as well as other choices of distance between Reeb graphs.

Function-induced persistence on simplicial complexes

Functions on Simplicial Complexes

- K: a simplicial complex, $|K|$ its underlying space (e.g. a triangulation of a manifold)
- Piecewise linear (PL) function $f: |K| \to R$
  - f defined at vertices (0-simplices) V of K and linearly interpolated within each simplex $\sigma \in K$



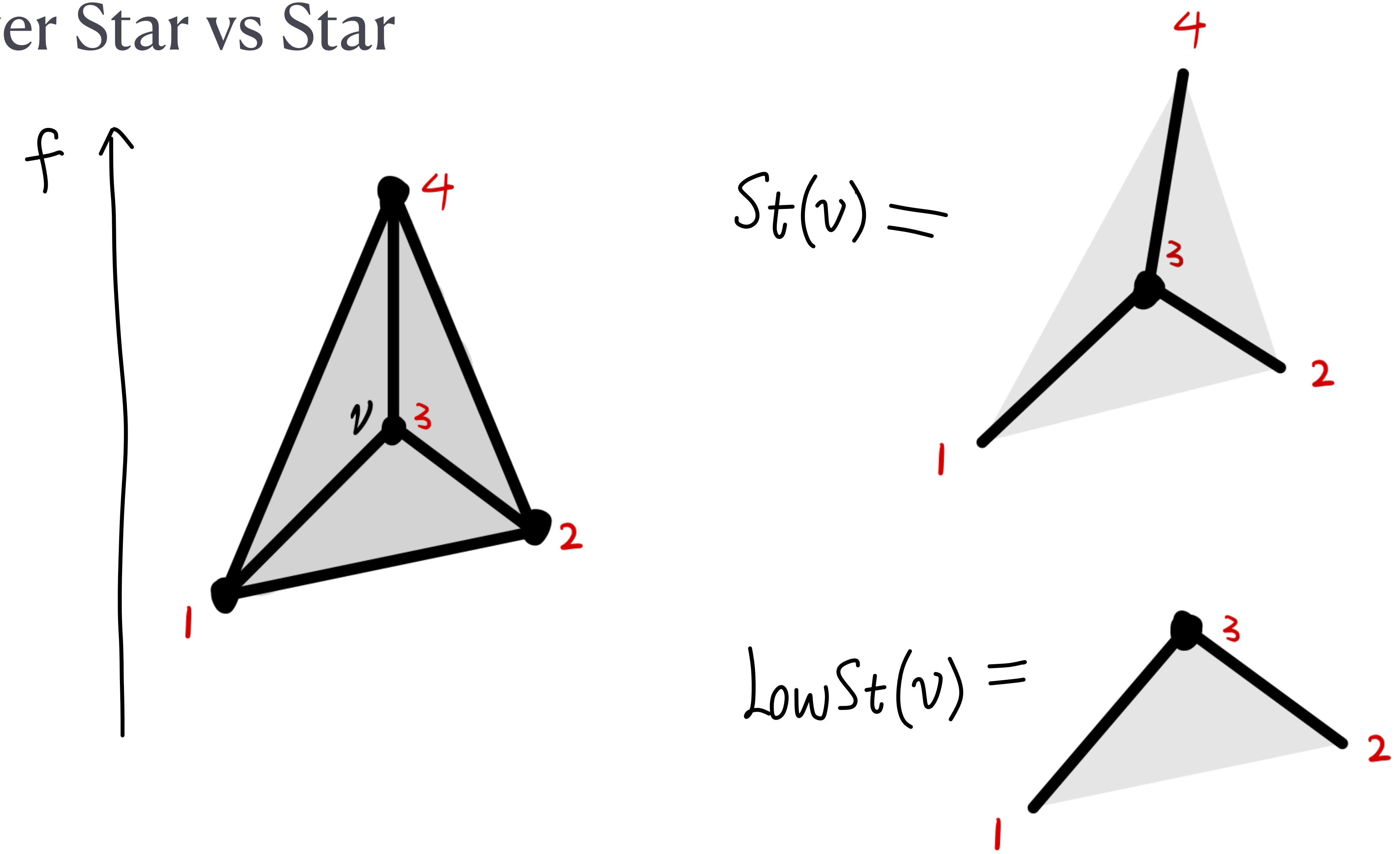
Computation – PL Function

- Given PL-function $f : \left| K \right| \to R$, consider the persistence module induced by its sub-level set filtration
  - $\mathscr{P}_f = \left\{ H_*\left(|K|^{\leq a} \right) \to H_*\left(|K|^{\leq b} \right) \right\}_{a \leq b}$
- $\{ \, |K|^{\leq a} \subset |K|^{\leq b} \, \}$ is still a filtration of topological spaces
- To compute persistence pairings for $\mathscr{P}_f$, we want to simulate sub-level set filtration by a **filtered simplicial complex**

Lower Star filtration

- Assume vertices $\{v_1, \ldots, v_n\}$ sorted in non-decreasing order with respect to function value f
- Consider discrete values $a_1 \leq \cdots \leq a_n$ with $a_i = f\left(v_i\right)$
  - $K_i := \{\sigma \in K \mid f(v) \leq a_i, \ \forall v \in \sigma\}$
  - Consider filtration $\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$
- This filtration is called the **lower star filtration**, because
  - $K_i = \bigcup_{j \leq i} LowSt(v_j)$
    - where $LowSt\ (v) := \{\sigma \in K \mid v \in \sigma \ and \ f(u) \leq f(v) \ for \ any \ u \in \sigma\}$

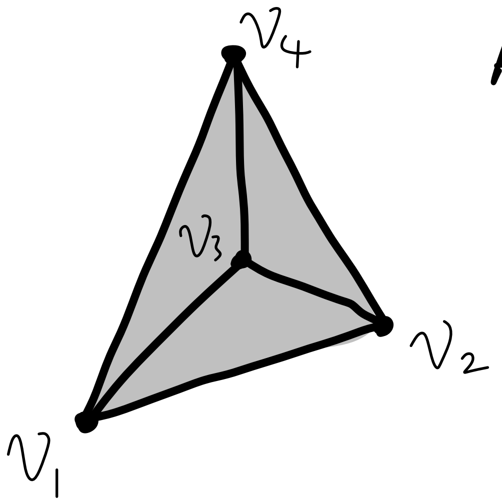
Lower Star vs Star

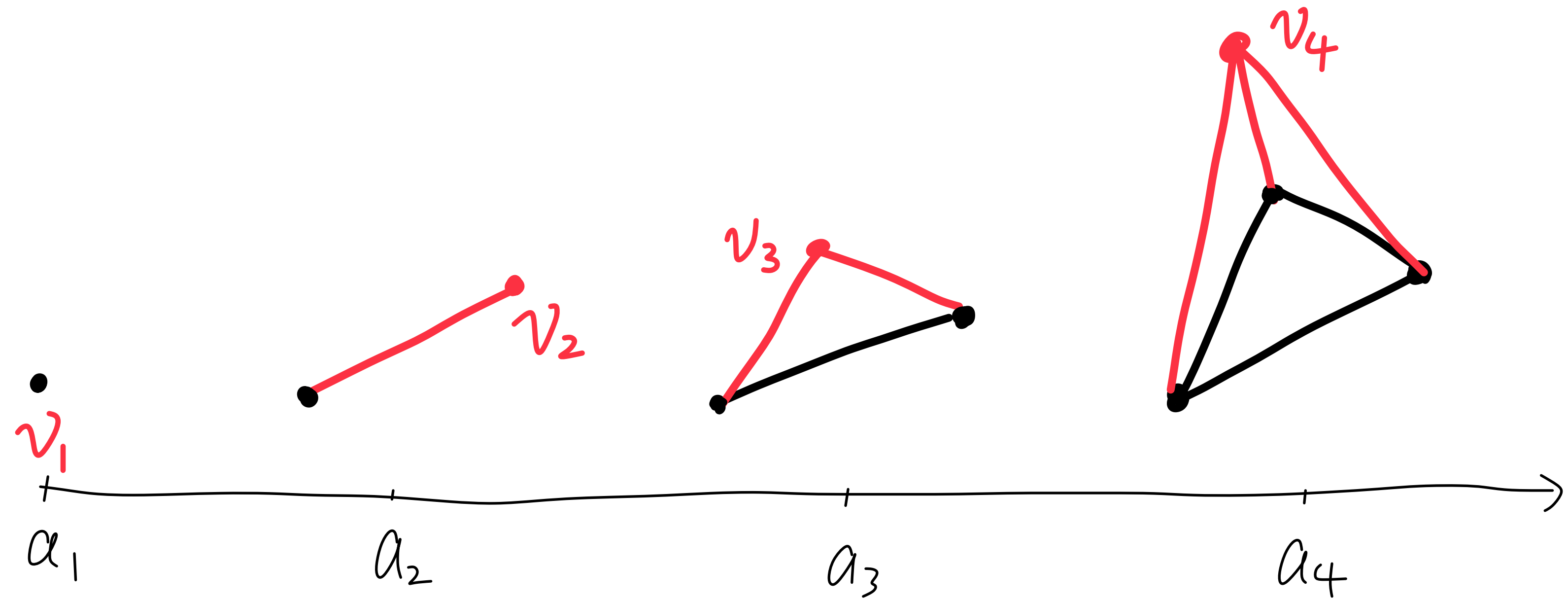


Lower Star filtration

- Consider discrete values $a_1 \leq \cdots \leq a_n$ with $a_i = f\left(v_i\right)$
  - $K_i := \{\sigma \in K \mid f(v) \leq a_i, \ \forall v \in \sigma\}$
  - Consider filtration $\varnothing = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$

Assume $a_i := f(v_i)$ & $a_1 \leq a_2 \leq a_3 \leq a_4$



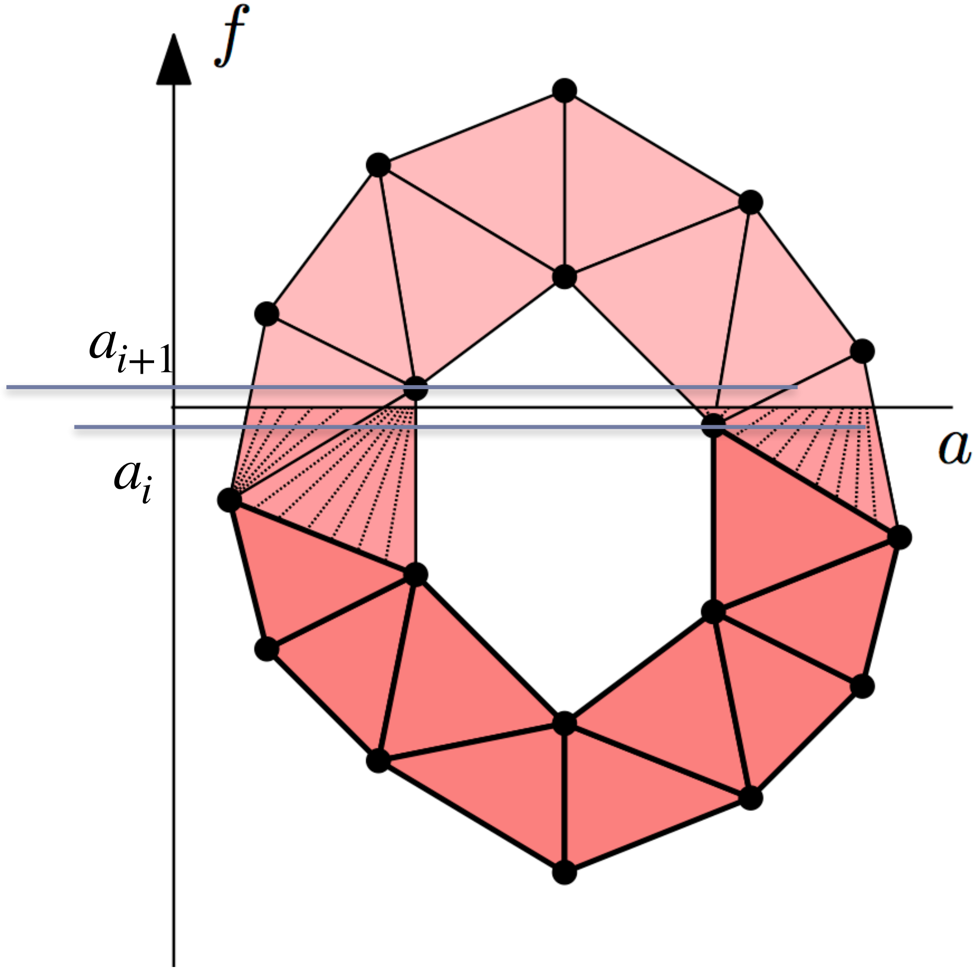


Compute sub-level set using lower star

- Goal: persistence pairings for the sub-level set filtration
  - $\mathcal{P}_f = \left\{ H_*\left(|K|^{\leq a}\right) \to H_*\left(|K|^{\leq b}\right) \right\}_{a \leq b}$
- Simulate sub-level set filtration by ***lower star filtration***
  - Assume vertices $\{v_1, \ldots, v_n\}$ sorted in non-decreasing order by function value f. Consider discrete values $a_1 \leq \cdots \leq a_n$ with $a_i = f\left(v_i\right)$
  - $\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$
    - where $K_i := \left\{\sigma \in K \mid f(v) \leq a_i,\ \forall v \in \sigma\right\}$
  - $\Rightarrow H_*\left(K_0\right) \to H_*\left(K_1\right) \to H_*\left(K_2\right) \to \cdots \to H_*\left(K_n\right)$

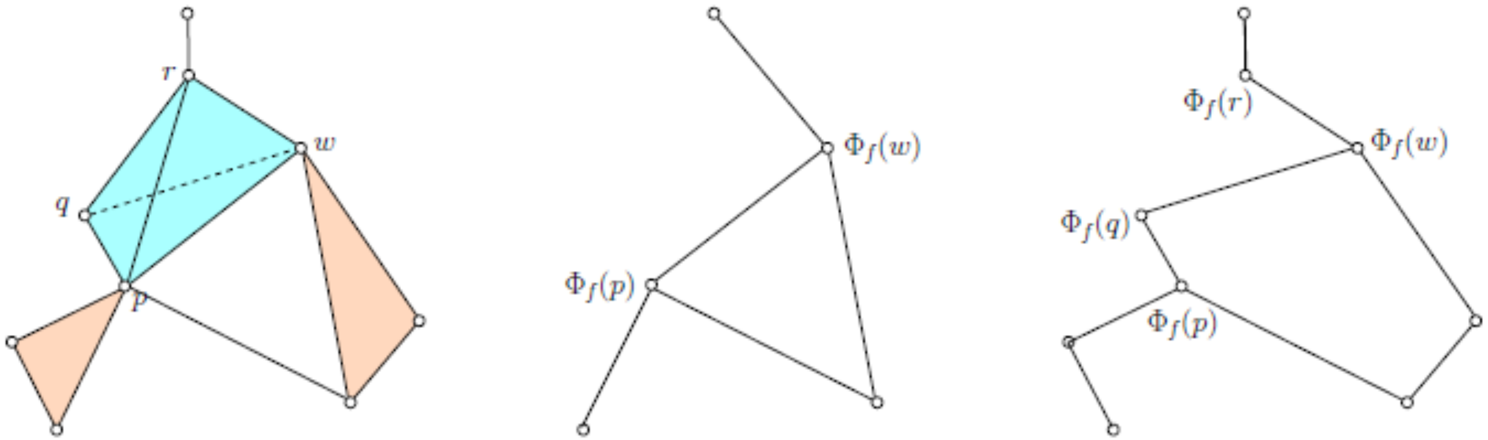
Sub-level set vs Lower Star filtrations

- If $a_i \leq a < a_{i+1}$, then $|K|^{\leq a} \simeq K_i$
- Lower star filtration
  - $\emptyset \subset K_1 \subset \cdots \subset K_n$
- Sub-level set filtration
  $\emptyset \subset |K|^{\leq a_1} \subset \cdots \subset |K|^{\leq a_n}$
- They induce **isomorphic** persistence homology



Reeb graph of PL functions on simplicial complexes

- PL function f defined on simplicial complex K
  - f is decided by function values on the vertices V of K
  - only 2-skelenton (V,E,T) of K matters
  - Reeb graph $R_f(X)$ can be computed in $O(m \log n)$ time
    - m: number of vertices, edges, and triangles of X,
    - n: number of vertices



Application of Reeb graphs

Geometric Graphs Reconstruction

- Geometric graphs
  - River / road networks, root systems, blood vessels, particle trajectories …
- Problem statement:
  - Given a set of points *P* sampled on / around a hidden geometric graph, reconstruct the graph from *P*
  - *[Ge, Safa, Belkin, Wang, NIPS 2011]*

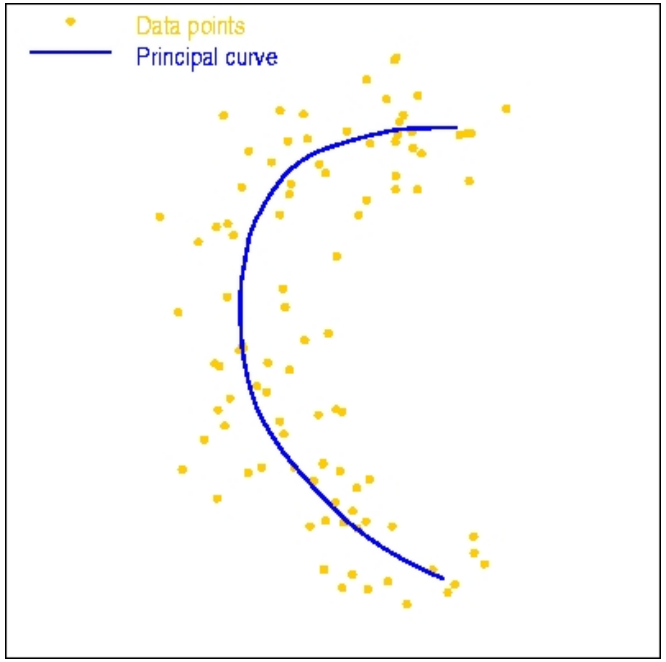
Data Skeletonization via Reeb Graphs

Xiaoyin Ge **Issam Safa** **Mikhail Belkin** **Yusu Wang**

Computer Science and Engineering Department
The Ohio State University
gex,safa,mbelkin,yusu@cse.ohio-state.edu

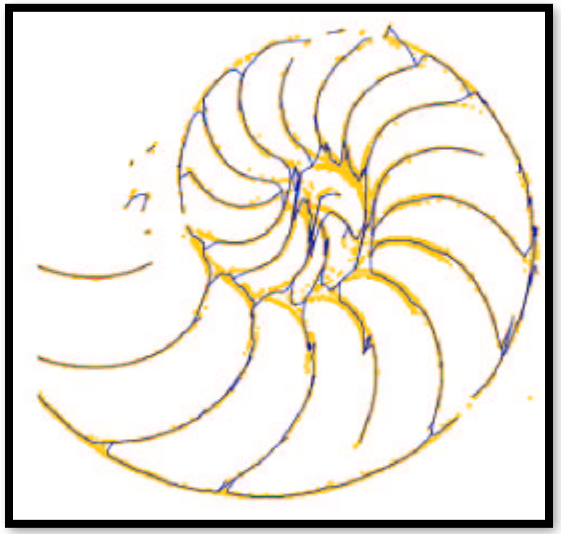
Related Work

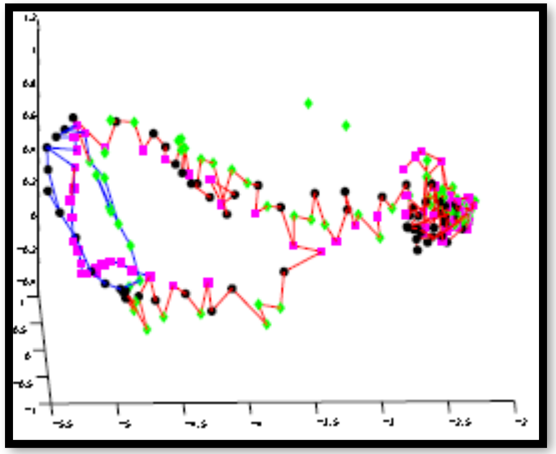
- Principal curves
  - *[Hastie and Stuetzle 84, 89],* and many followups
  - Self-consistent curves passing through the middle of a data cloud
- Principle graph
  - *[Kegl et al., 2002]*
  - 2D images
- Principle graphs / surfaces
  - *[Ozertem and Erdogmus 2011]*
  - High-dimensional, but does not guarantee output is a graph
- Metric graph learning
  - *[Aanjaneya et al., 2011]*
  - With certain theoretical gurantee! But sensitive to parameters

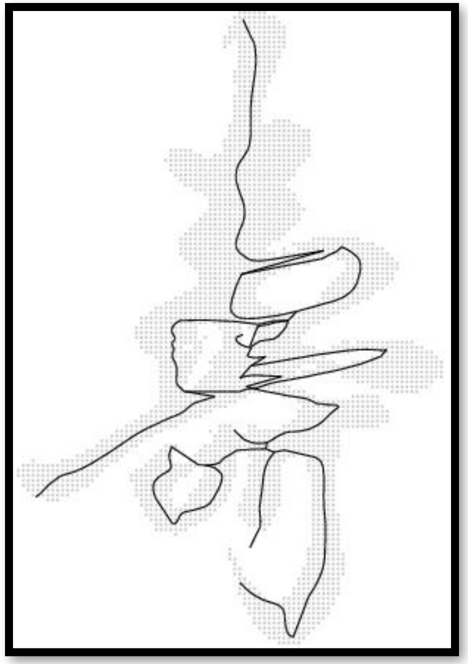


New Approach

- Use the Reeb graph
  - Robust, always recover a graph structure
  - Natural simplification algorithm
  - Simple and efficient
  - Certain theoretical guarantee

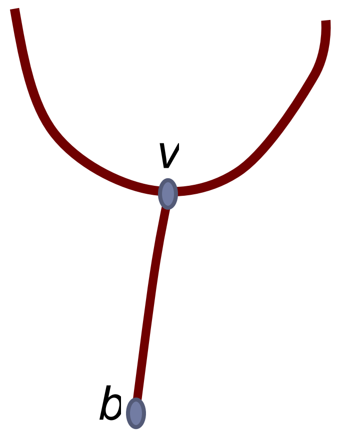




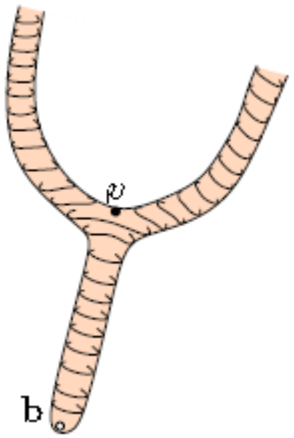


Intuition

- Given a graph G, fix any base point b, consider the shortest path distance function f (to b)
  - Reeb graph $R_f(G)$ same as G

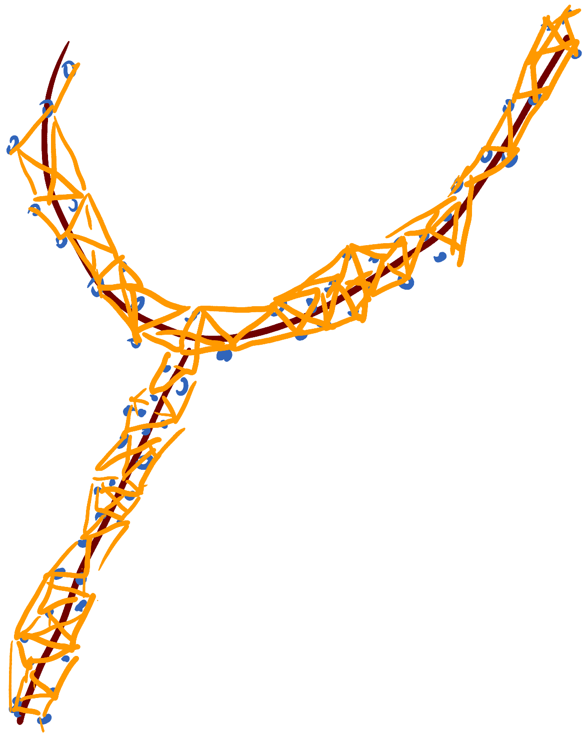


- Given a graph-like structure X, fix any base point b, consider the shortest distance function f (to b)
  - Reeb graph $R_f(X)$ captures underlying graph



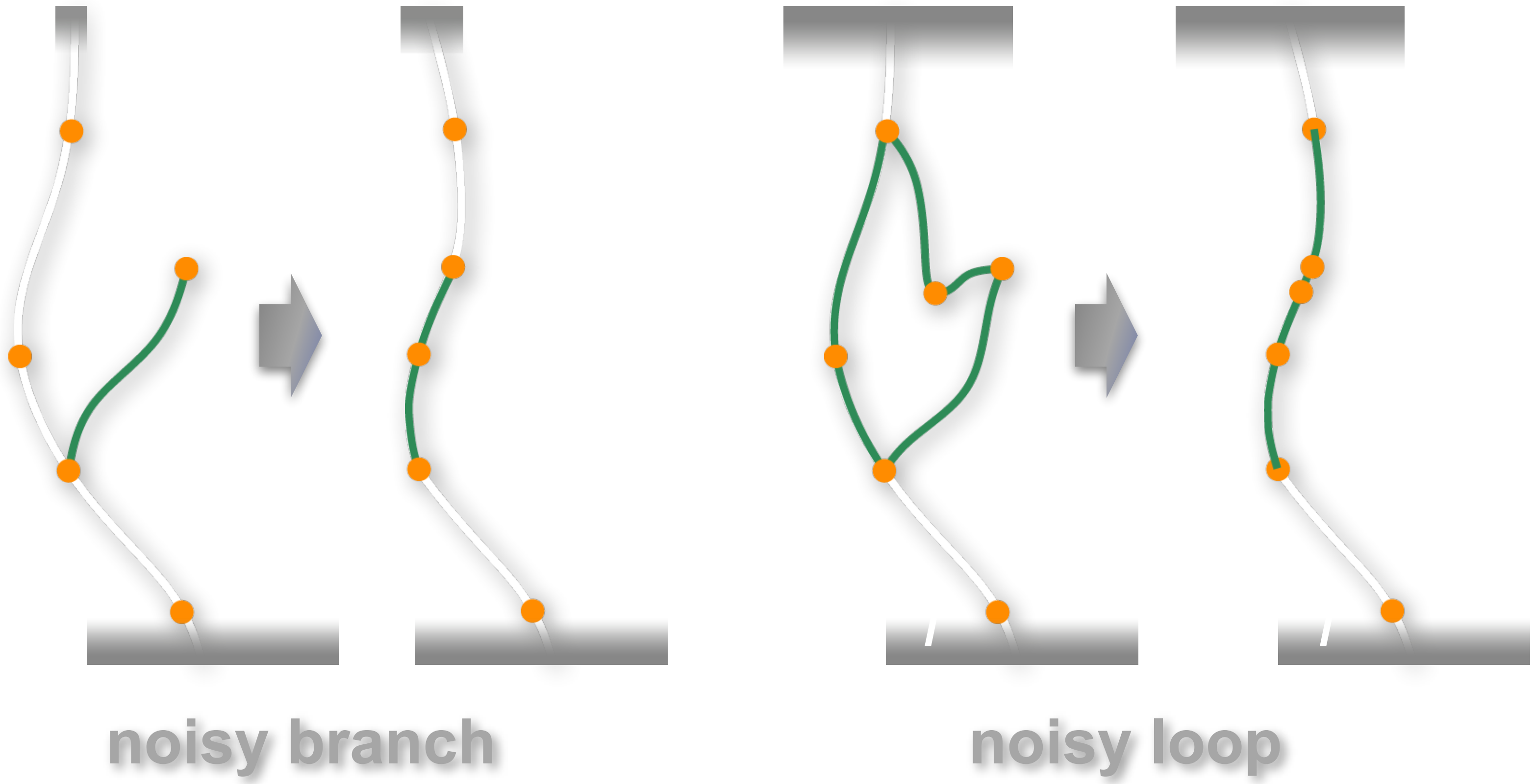
Algorithm Overview

- Input: Given a set of input points P (Presumably sampled around a hidden geometric graph-like structure).
- Construct Rips Complex $K=R^r(P)$ for appropriate parameter r
- Choose a base point b, compute shortest distance to all other vertices in K as a PL function f
- Compute Reeb graph w.r.t. f
  - Relation between the Reeb graph from Rips complex with that of the hidden structure: see [Dey and Wang, SoCG 2011]
- Simplify Reeb graph (to remove noise) if necessary



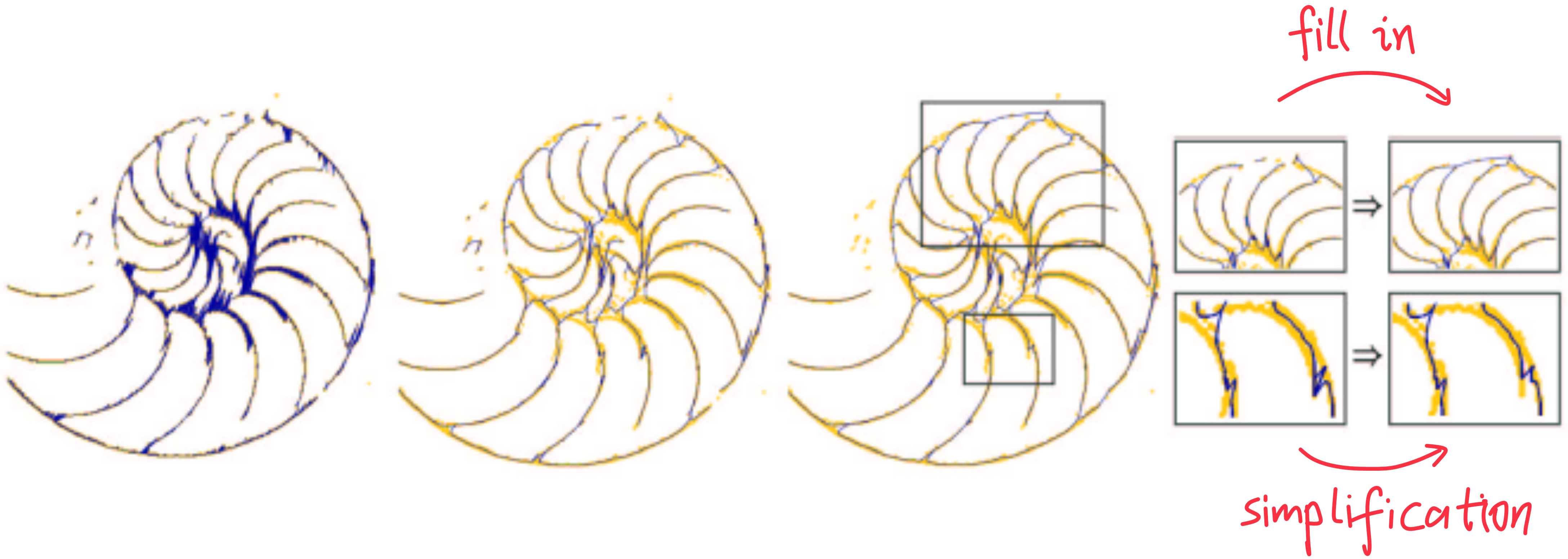
Post-processing: Simplification

- Branches and Loops simplificaiton

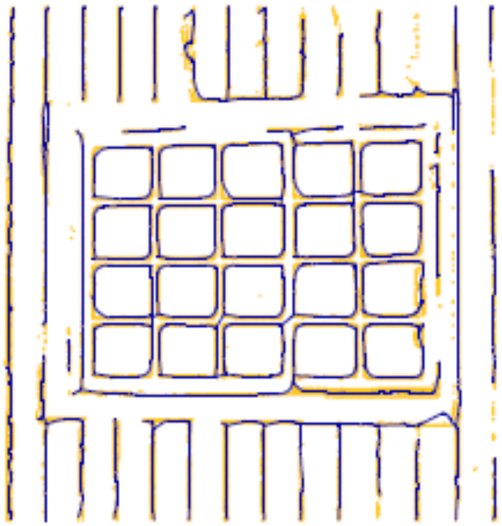


Post-processing

- May fill in some missing links before simplification



Examples



(a)

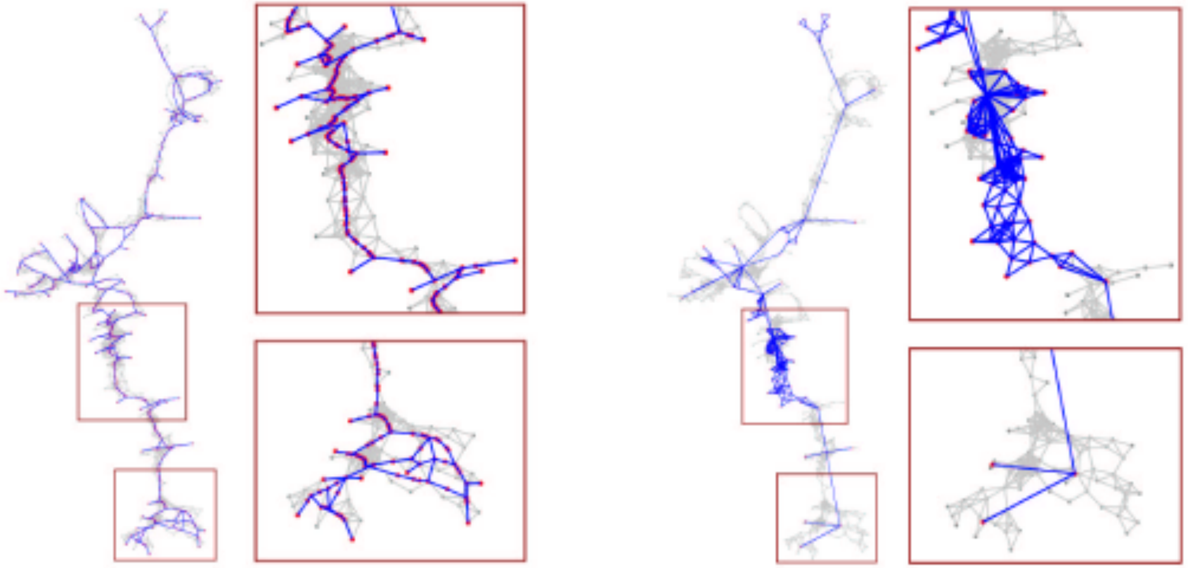
(b)

(c)

More Examples

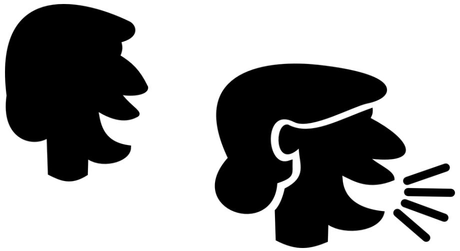
(a) Our output (b) PGA [19] (c) LDPC [23]

More Examples

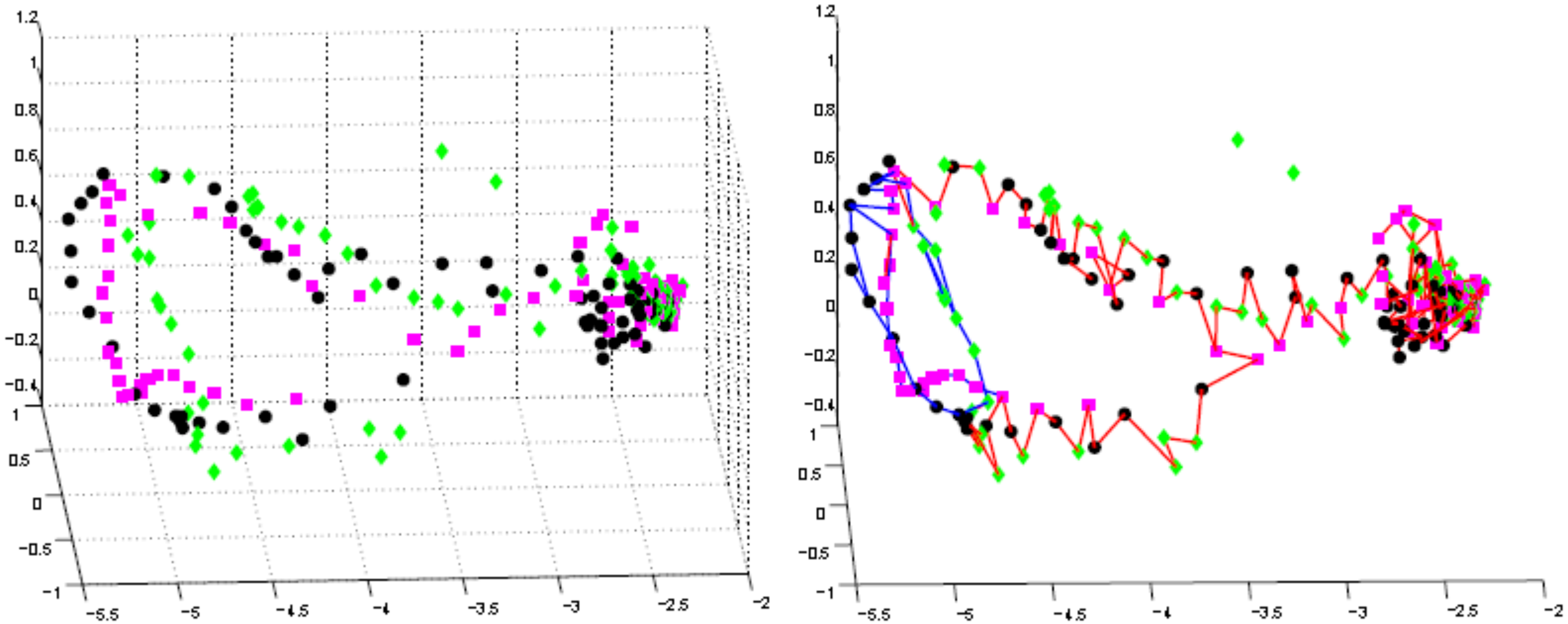


(a) Our output (b) MGR [2]

More Examples – Speech Data



- Projected in 3D for visualization

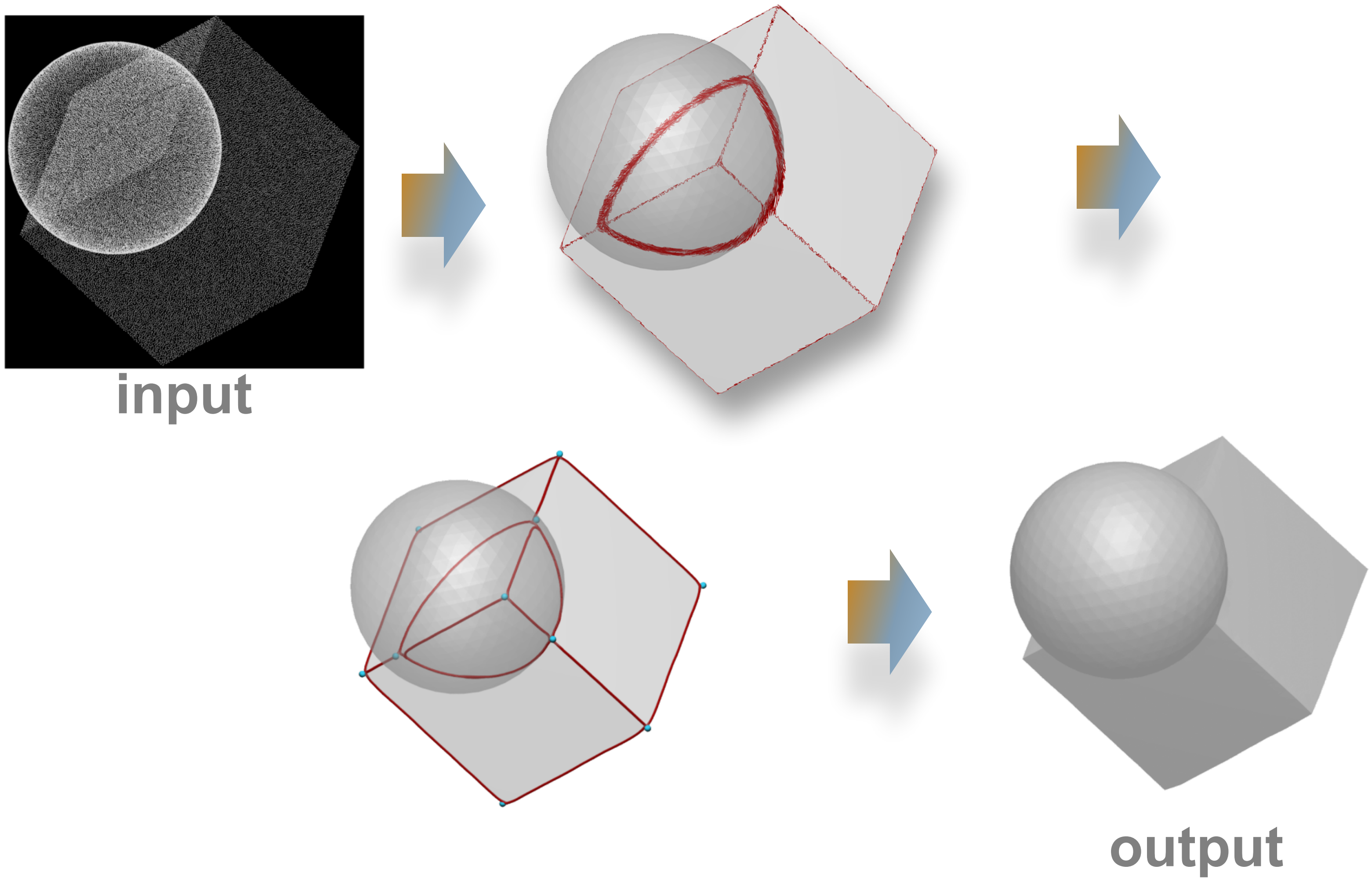


3D singular surface reconstruction

- *[Dey, Ge,Que,Safa,Wang,Wang, SGP2012]*

Feature-Preserving Reconstruction of Singular Surfaces

T. K. Dey, X. Ge, Q. Que, I. Safa, L. Wang, Y. Wang



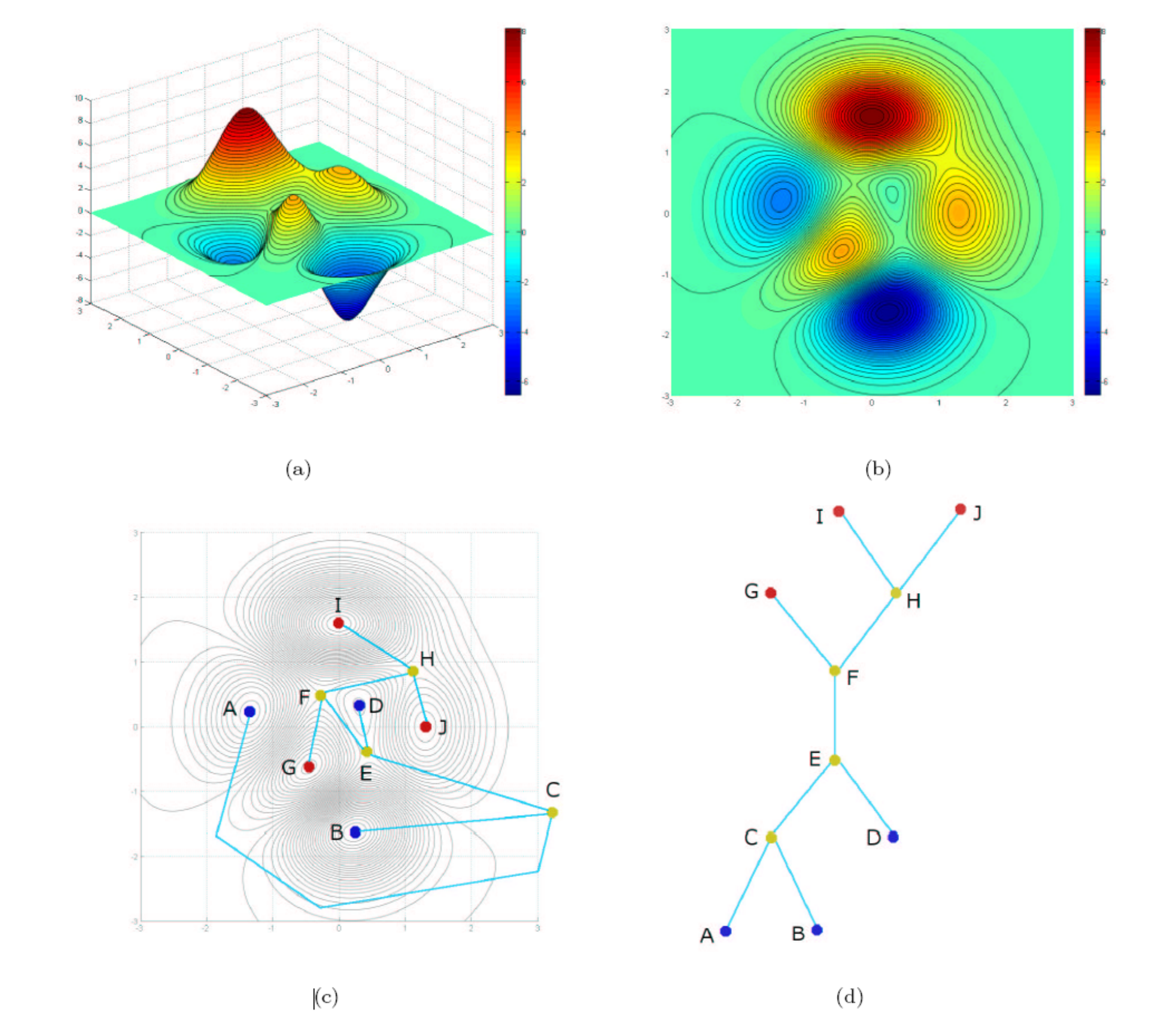
Contour trees

Contour Trees

If domain X is simply-connected,
then $R_f(X)$ is loop-free (i.e, is a tree)
which is called a *contour tree*.
This is true independent of the function f.

The inverse is not true.

Contour Trees



(a) (b) (c) (d)

Join/Merge, Split and Contour Trees

- Contour tree tracks connected components in level sets
- Join/Merge tree tracks connected components in sublevel sets
- Split tree tracks connected components in suplevel set

Contour tree

$T_C = X/\sim$, for $x \sim y$ iff

(1) $f(x) = f(y) = a$

(2) x & y connected in $f^{-1}(\{a\})$

Join/Merge tree

$T_J = X/\sim$, for $x \sim y$ iff

(1) $f(x) = f(y) = a$

(2) x & y connected in $f^{-1}((-\infty, a])$

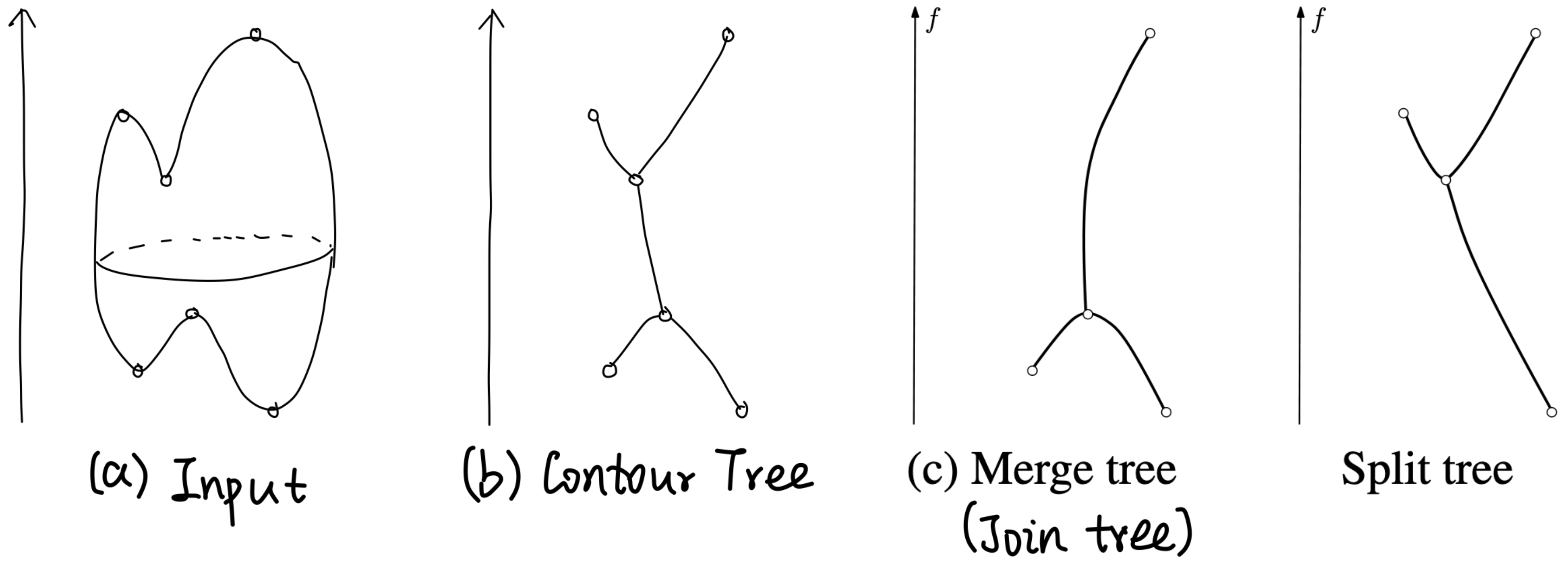
Split tree

$T_S = X/\sim$, for $x \sim y$ iff

(1) $f(x) = f(y) = a$

(2) x & y connected in $f^{-1}([a, \infty))$

Join/Merge, Split and Contour Trees



- T_C if exists, is the unique tree such that
  - $T_J = JoinTree(T_C)$
  - $T_S = SplitTree(T_C)$

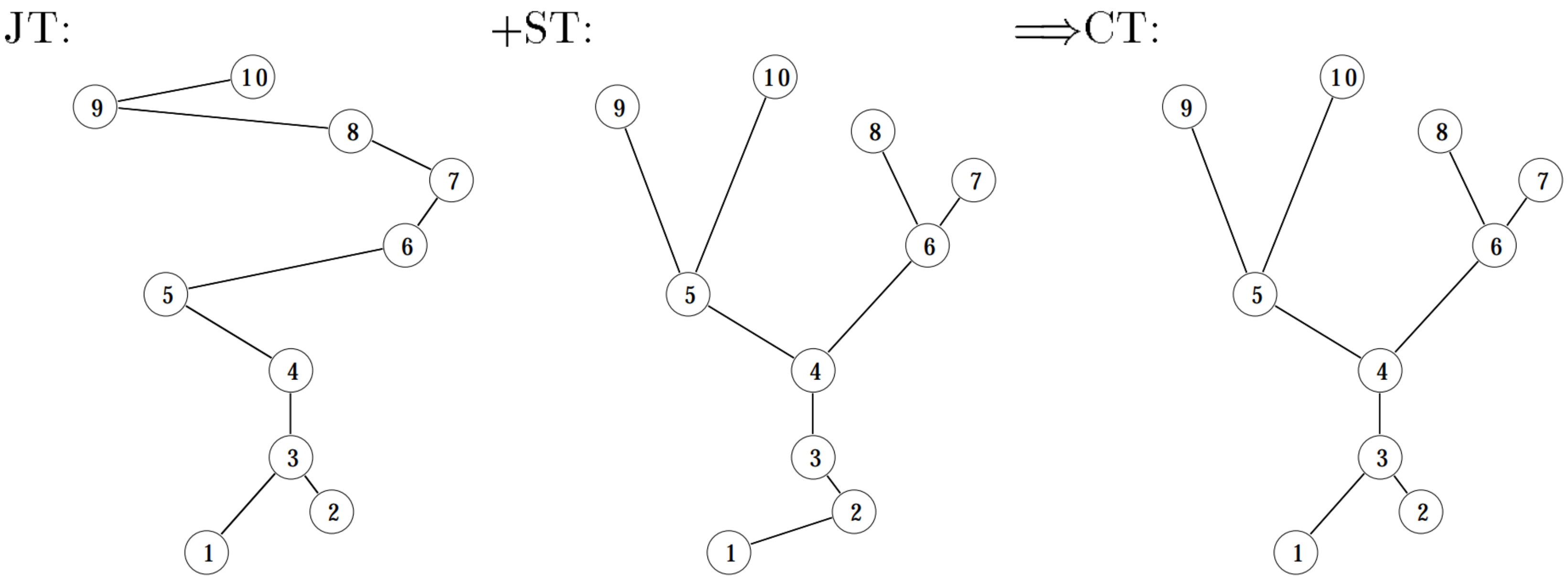
Contour tree contains full information of join and split trees!

Computation

Computing Contour Trees in All Dimensions*

Hamish Carr † Jack Snoeyink‡ Ulrike Axen§

- An elegant near-linear time algorithm to compute contour tree based on *merging* join and split trees
  - *[Carr, Snoeyink, Axen, 2003]*

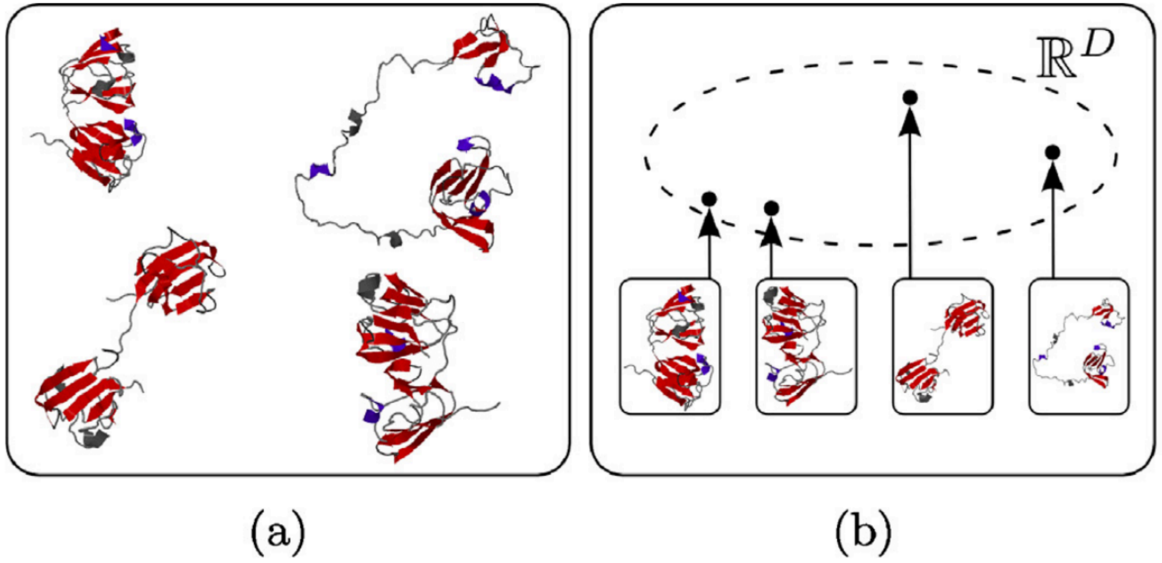


Application

A collaborative visual analytics suite for protein folding research

William Harvey[a,*], In-Hee Park[b], Oliver Rübel[c], Valerio Pascucci[d], Peer-Timo Bremer[e], Chenglong Li[f], Yusu Wang[g,**]

2014



- Each molecular conformation consists of 3D coordinates of atoms.

$$\text{Conformation } i = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_N, y_N, z_N)\}$$

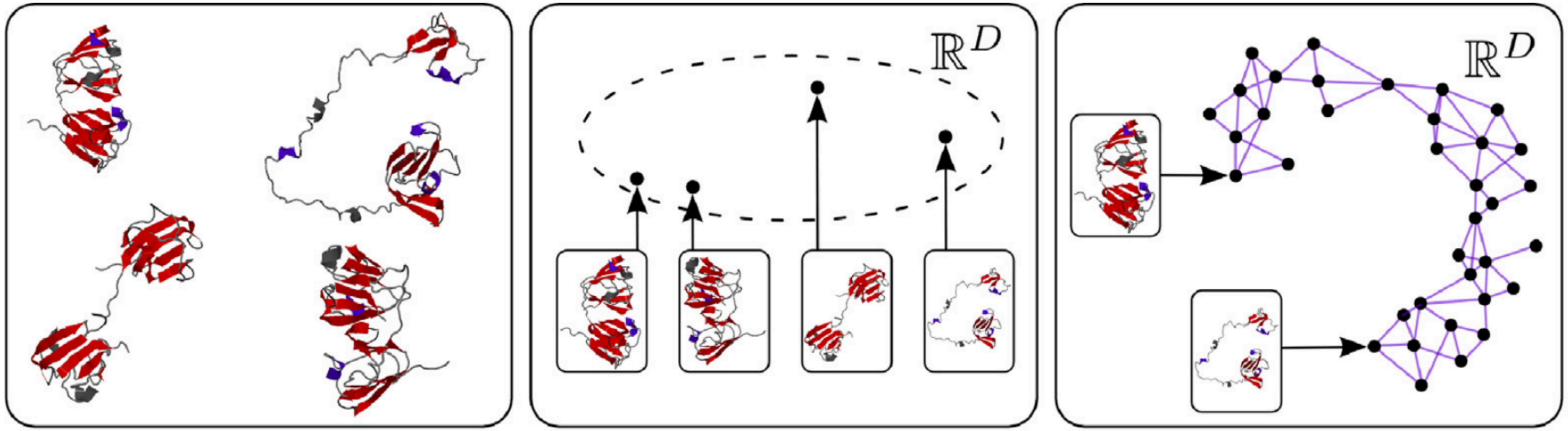
- Embed a conformation to $\mathbb{R}^D$
  - Option 1: flatten each conformation to a 3N vector, so D = 3N. (sensitive to rotation)
  - Option 2: use pairwise distances between atoms. (sensitive to reflections)

Application

A collaborative visual analytics suite for protein folding research

William Harvey[a,*], In-Hee Park[b], Oliver Rübel[c], Valerio Pascucci[d], Peer-Timo Bremer[e], Chenglong Li[f], Yusu Wang[g,**]

2014



(a) (b) (c)

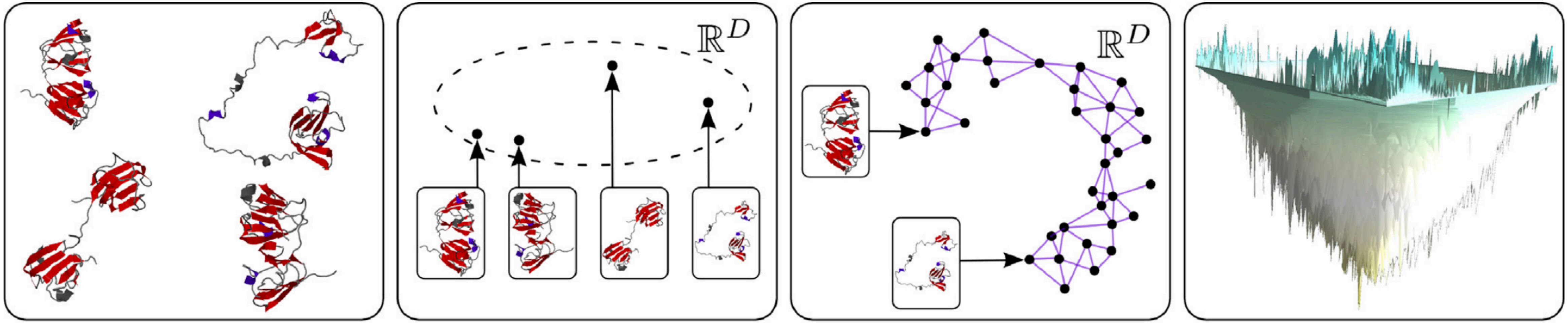
- For each point p in (b), connect p with all k-nearest neighbors of p to form a graph G.
- Let K be the clique complex of G.
- Define f to be the energy function, sending each point p to the energy of the corresponding conformation.

Application

A collaborative visual analytics suite for protein folding research

William Harvey[a,*], In-Hee Park[b], Oliver Rübel[c], Valerio Pascucci[d], Peer-Timo Bremer[e], Chenglong Li[f], Yusu Wang[g,**]

2014



(a) (b) (c) (d)

- Compute the contour tree (T, f) of (K, f) using [Carr, Snoeyink, Axen, 2003]
  - Only the 1-skeleton is needed in the algorithm
  - Nodes of the contour tree represents minima, maxima and saddle of f
- Visualization: make a 3D terrain such that its contour tree is also (T, f)

Visualization/Interpretation for Contour Trees

- Input:
  - A contour tree T of high dimensional landscape $E: X \to R$
- Output:
  - A 2D terrain $F: [0,1] \times [0,1] \to R$ such that F shares the same contour tree as T

Topological Landscape Ensembles for Visualization of Scalar-Valued Functions

William Harvey, Yusu Wang

First published: 12 August 2010 | https://doi.org/10.1111/j.1467-8659.2009.01706.x | Citations: 33