“Persistent Homology” Summer School - Rabat

From a Point Cloud To a Filtered Simplicial Complex

Ulderico Fugacci

Kaiserslautern University of Technology

Department of Computer Science

Technische Universität Kaiserslautern

July 3, 2017

Outline

Describing a Shape through Persistence Pairs

Outline

Describing a Shape through Persistence Pairs

From a Point Cloud to a Filtered Simplicial Complex

Persistent Homology

In a Nutshell:

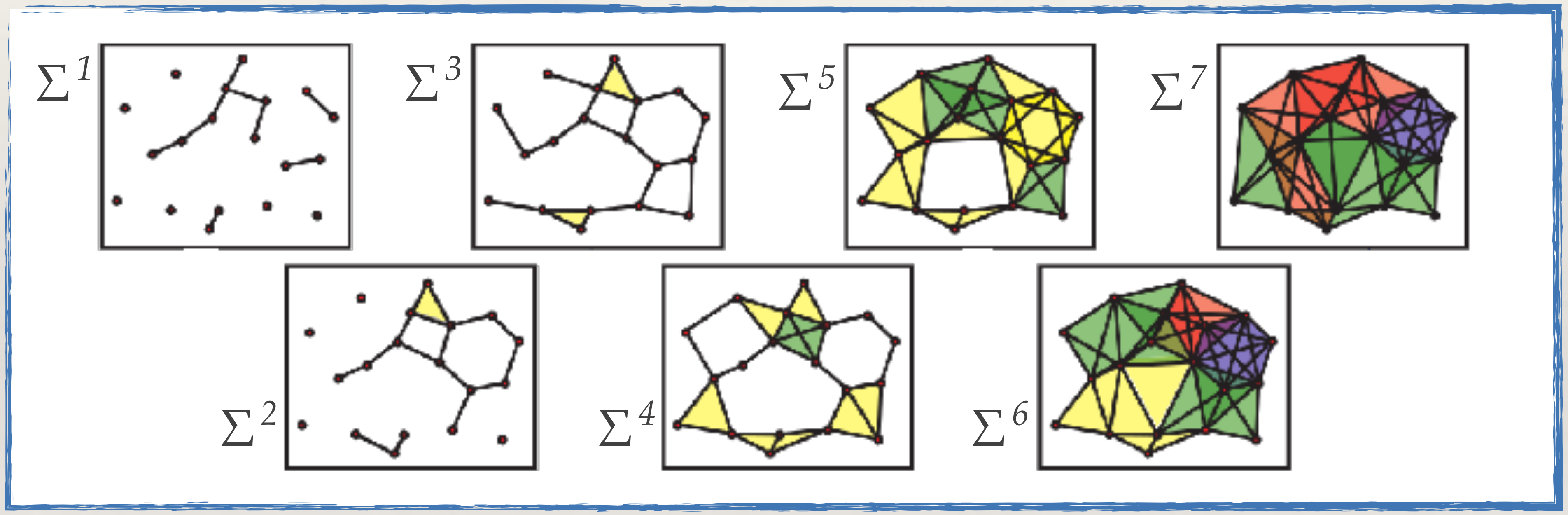


Image from [Ghrist 2008]

Persistent homology allows for
describing the changes in the shape of an evolving object

Persistent Homology

An Evolving Notion:

1990

Frosini

Size Functions:

- ***Estimation of natural pseudo-distance*** between shapes endowed with a function f
- Tracking of the ***connected components*** of a shape along its evolution induced by f

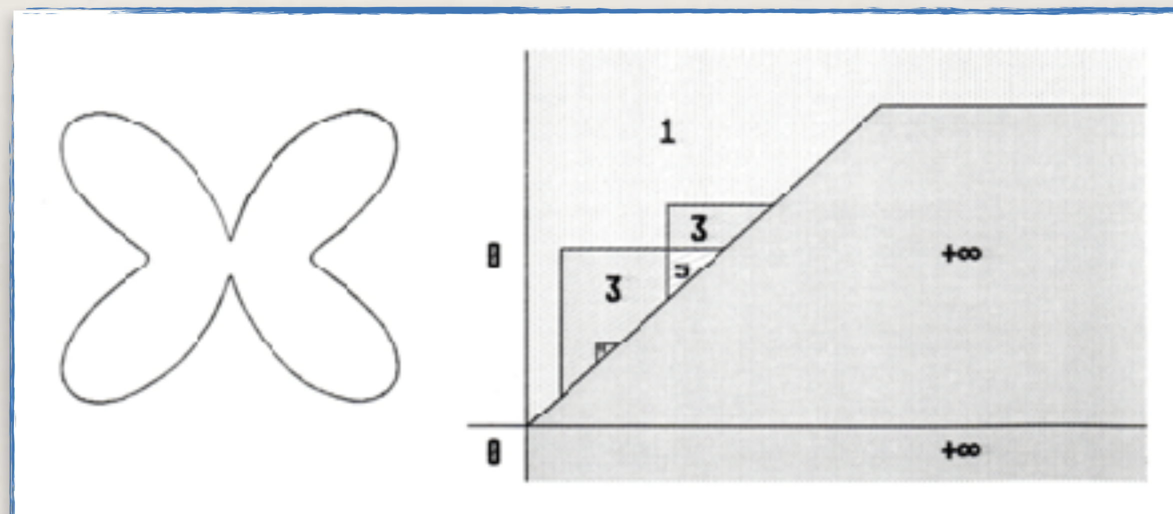


Image from [Frosini 1992]

Actually, this coincides with ***persistent homology in degree 0***

Persistent Homology

An Evolving Notion:

1990 — Frosini

1994 — Delfinado, Edelsbrunner

Incremental Algorithm for Betti Numbers:

- Introduction of the notion of *filtration*
- De facto computation of ***persistence pairs***

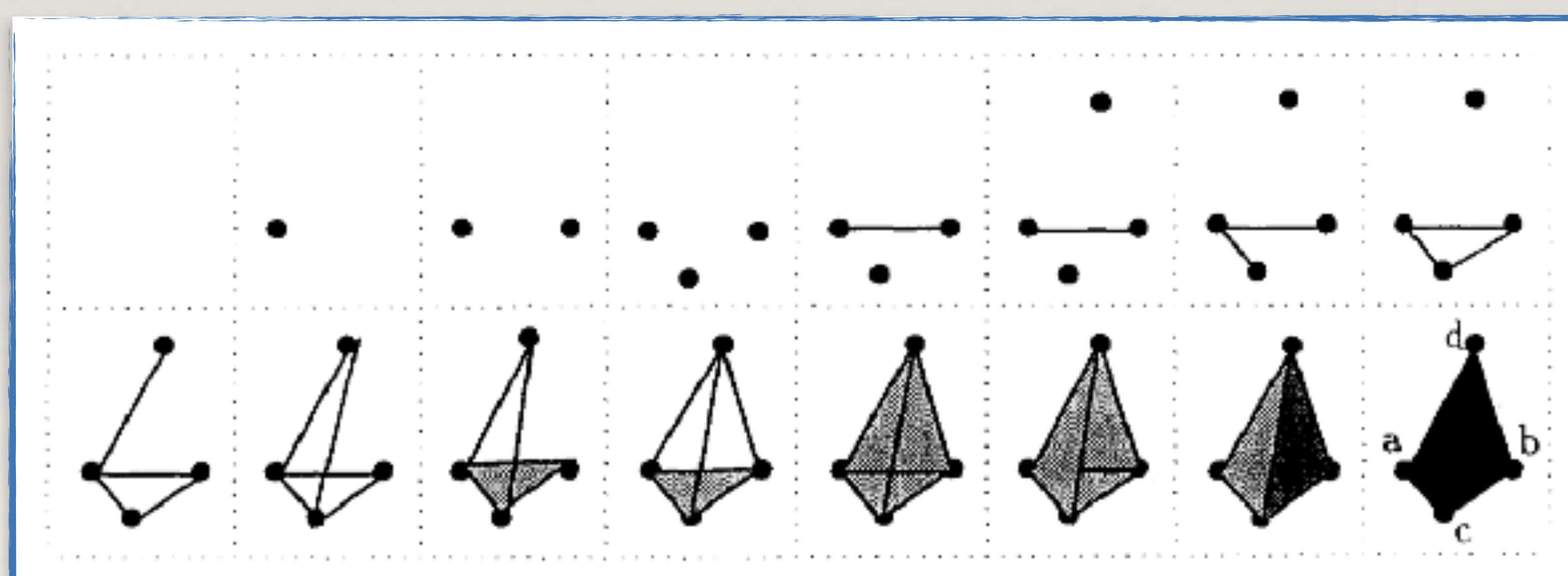


Image from [Delfinado, Edelsbrunner 1995]

Persistent Homology

An Evolving Notion:

1990 — Frosini

1994 — Delfinado, Edelsbrunner

1999 — Robins

Homology from Finite Approximations:

- *Extrapolation of the homology* of a metric space from a *finite point-set approximation*
- Introduction of *persistent Betti numbers*

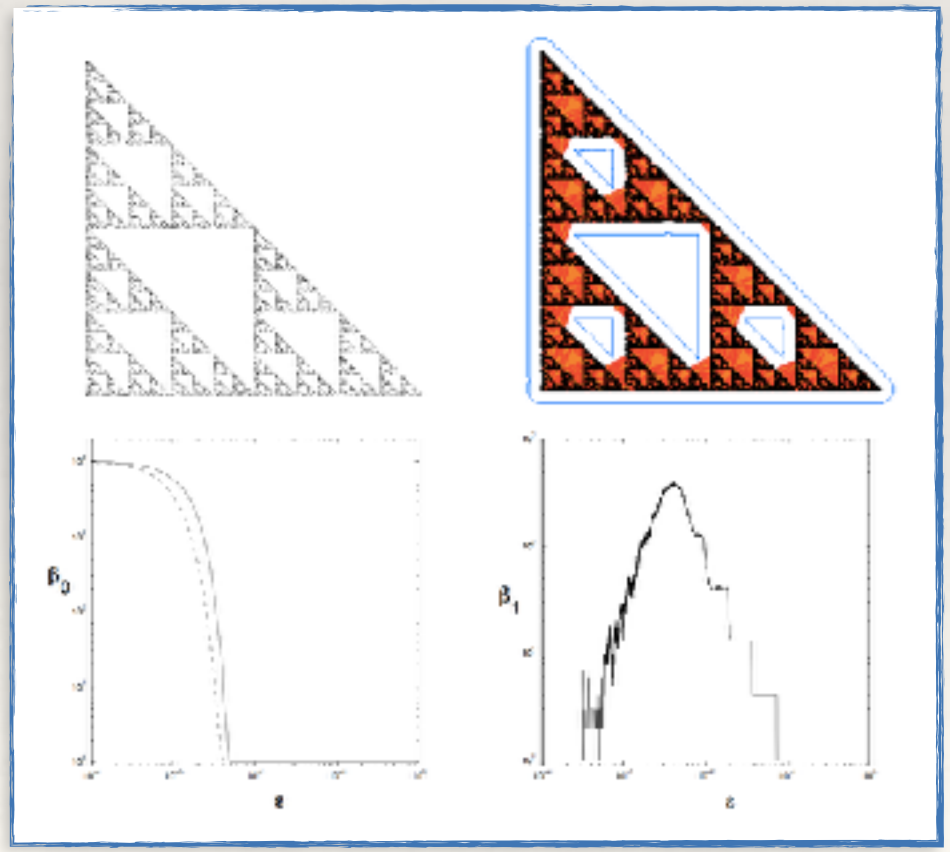


Image from [Robins 1999]

Persistent Homology

An Evolving Notion:

1990 — Frosini

1994 — Delfinado, Edelsbrunner

1999 — Robins

2002 — Edelsbrunner, Letscher, Zomorodian

Topological Persistence:

- Introduction and algebraic formulation of the notion of *persistent homology*
- ***Description of an algorithm*** for computing persistent homology

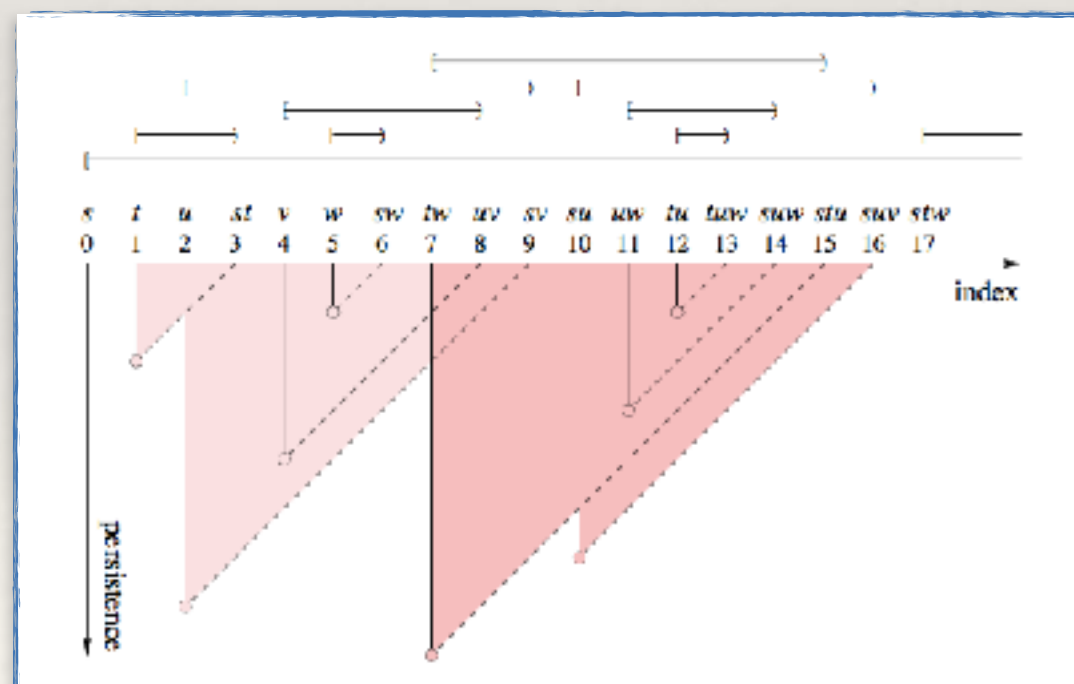


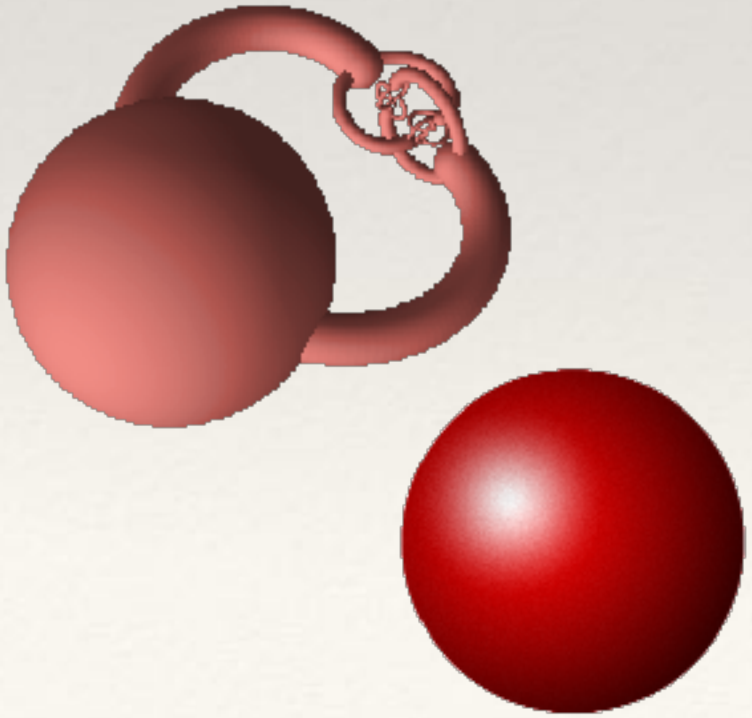
Image from [Edelsbrunner et al. 2002]

Persistent Homology

A Twofold Purpose:

Shape Description

- *Which is the shape of a given data?*



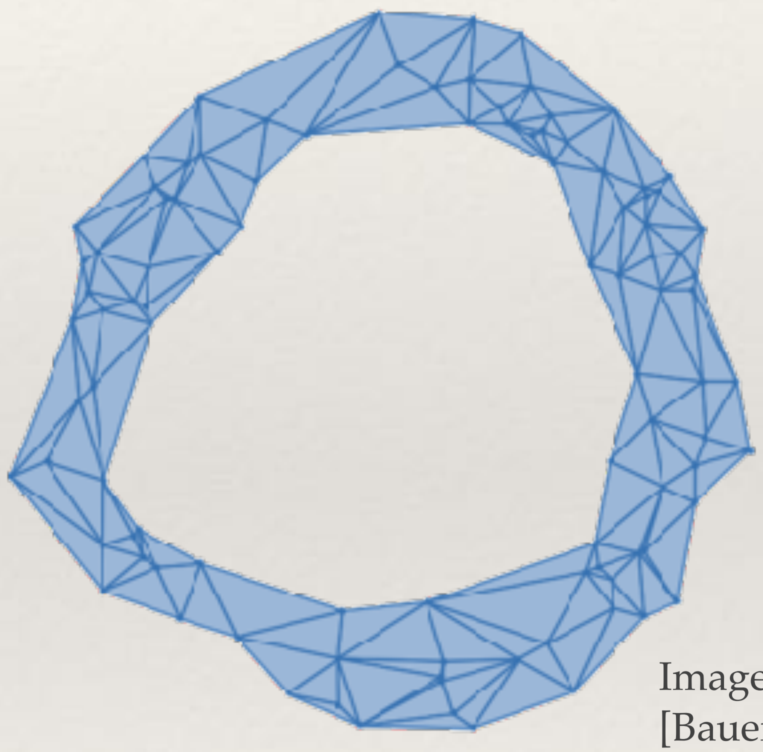
Shape Comparison

- *Given two data, do they have the same shape?*

Shape Description

- *Which is the shape of a given data?*

Persistent homology allows for the retrieval of the *"actual"* homological information of a data



Images from [Bauer 2015]

Point Cloud Dataset → **Topological Nature of the "Underlying" Shape**

Shape Description

- *Which is the shape of a given data?*

Persistent homology allows for the retrieval of the *"actual"* homological information of a data

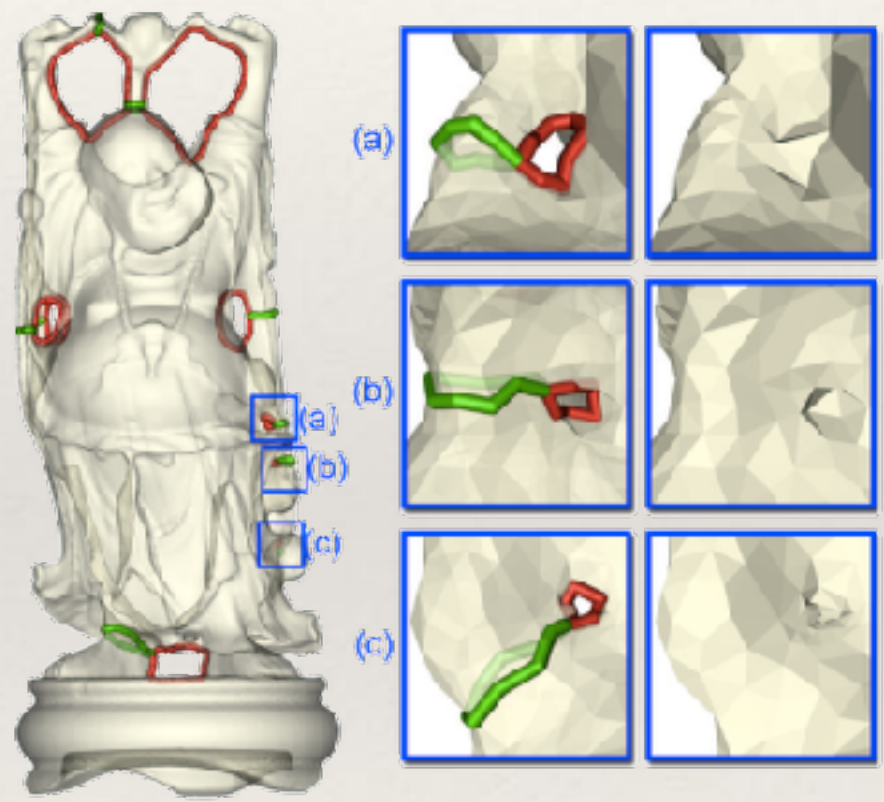


Image from [Dey et al. 2008]

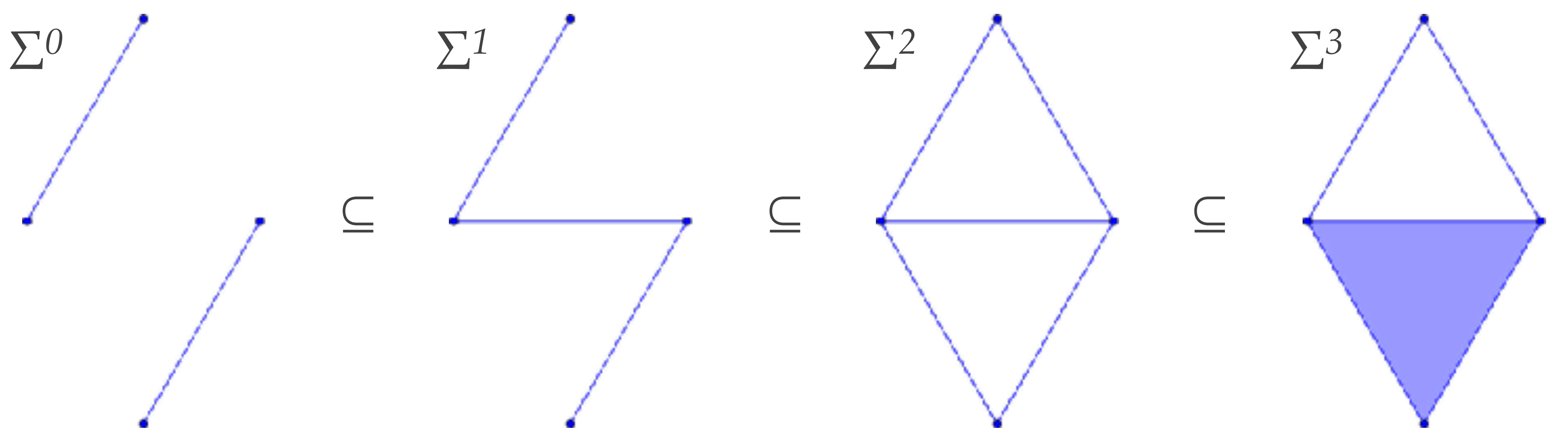
Noisy Dataset → **Relevant Homological Information**

Shape Description

The *core information* of persistent homology is given by the ***persistence pairs***

Persistence Pairs:

Given a filtration $\Sigma^0 \subseteq \Sigma^1 \subseteq \ldots \subseteq \Sigma^m$,



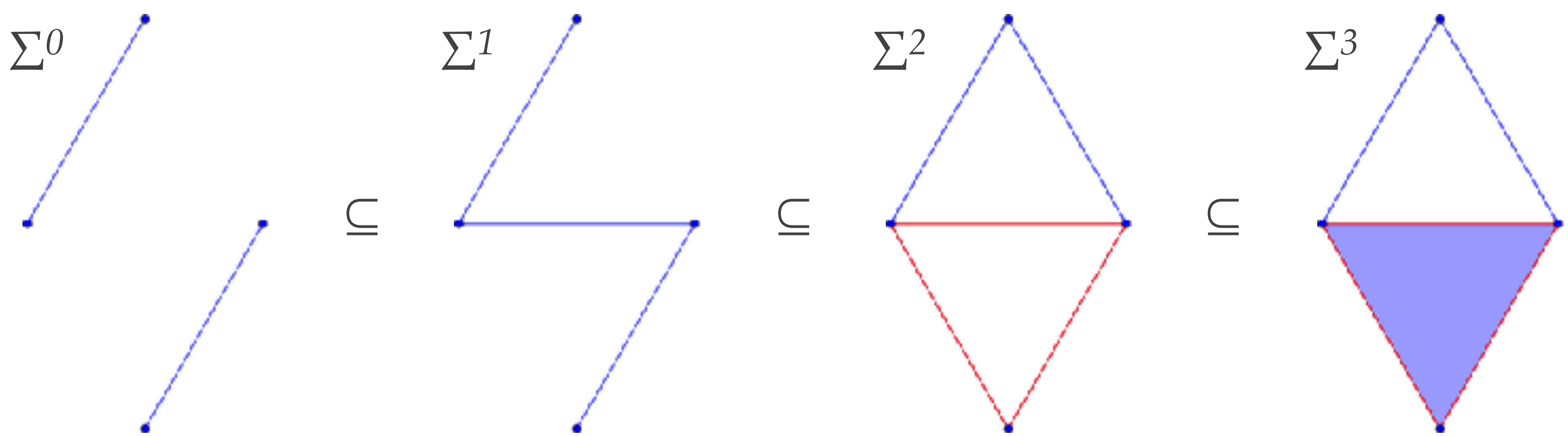
A **persistence pair** (p, q) is an element in $\{0, \ldots, m\} \times (\{0, \ldots, m\} \cup \{\infty\})$ such that $p < q$ representing a **homological class** that is **born at step** p and **dies at step** q

Shape Description

The *core information* of persistent homology is given by the ***persistence pairs***

Persistence Pairs:

Given a filtration $\Sigma^0 \subseteq \Sigma^1 \subseteq \ldots \subseteq \Sigma^m$,



(2, 3)

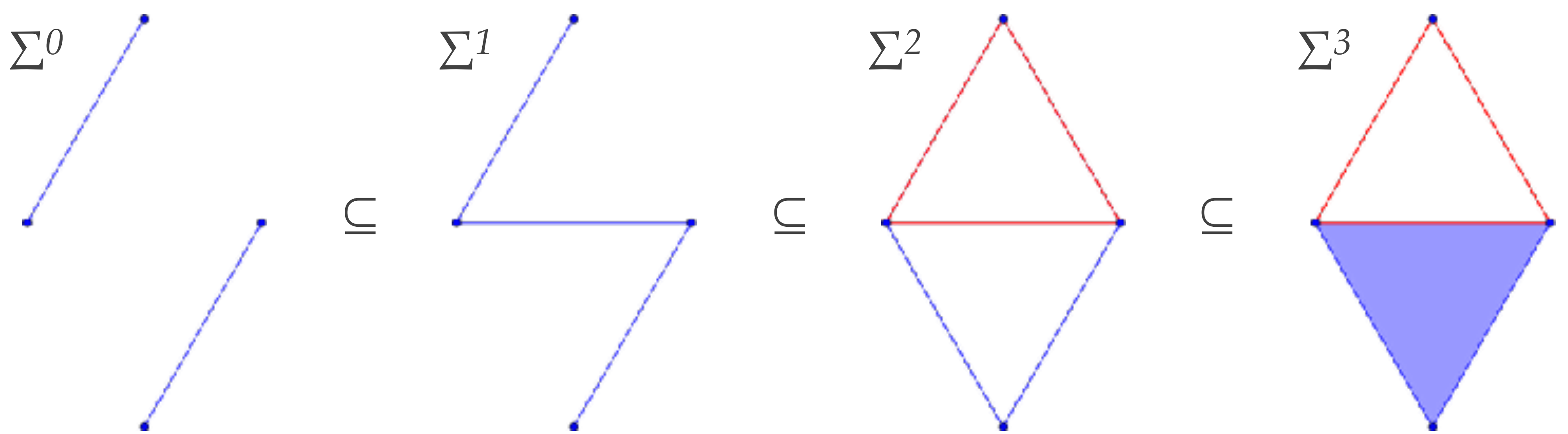
A **persistence pair** (p, q) is an element in $\{0, \ldots, m\} \times (\{0, \ldots, m\} \cup \{\infty\})$ such that $p < q$ representing a **homological class** that is **born at step** p and **dies at step** q

Shape Description

The *core information* of persistent homology is given by the *persistence pairs*

Persistence Pairs:

Given a filtration $\Sigma^0 \subseteq \Sigma^1 \subseteq \ldots \subseteq \Sigma^m$,

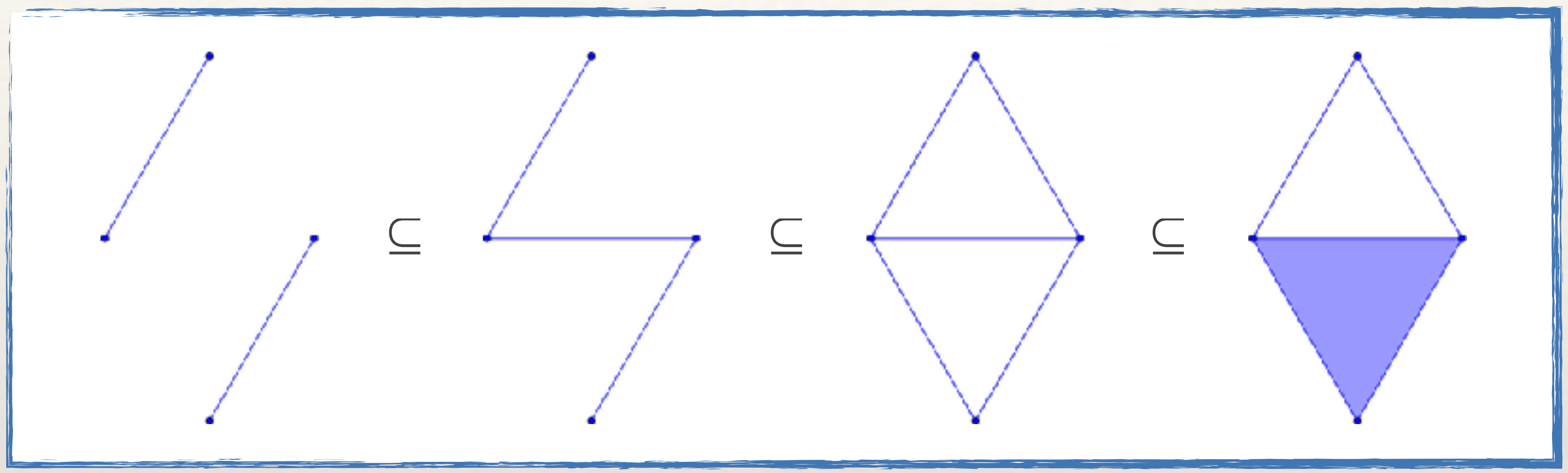


$(2, \infty)$ *essential pair*

A **persistence pair** (p, q) is an element in $\{0, \ldots, m\} \times (\{0, \ldots, m\} \cup \{\infty\})$ such that $p < q$ representing a **homological class** that is **born at step** p and **dies at step** q

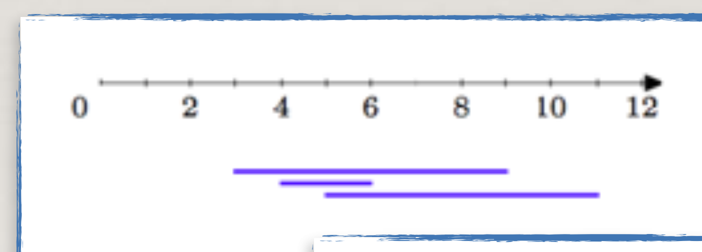
Shape Description

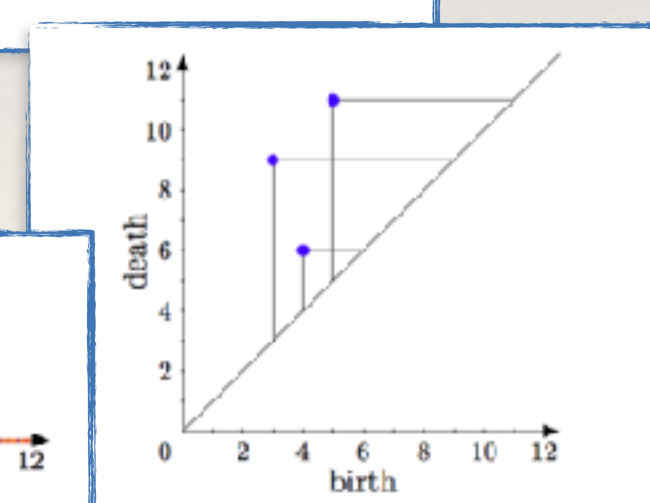
Given a filtered simplicial complex Σ,

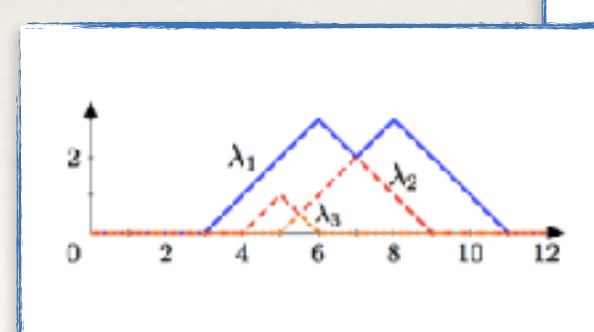


Persistent pairs of Σ can be visualized through:

- ***Barcodes*** [Carlsson et al. 2005; Ghrist 2008]
- ***Persistence diagrams*** [Edelsbrunner, Harer 2008]
- ***Persistence landscapes*** [Bubenik 2015]
- ***Corner points and lines*** [Frosini, Landi 2001]
- ***Half-open intervals*** [Edelsbrunner et al. 2002]
- ***k-triangles*** [Edelsbrunner et al. 2002]

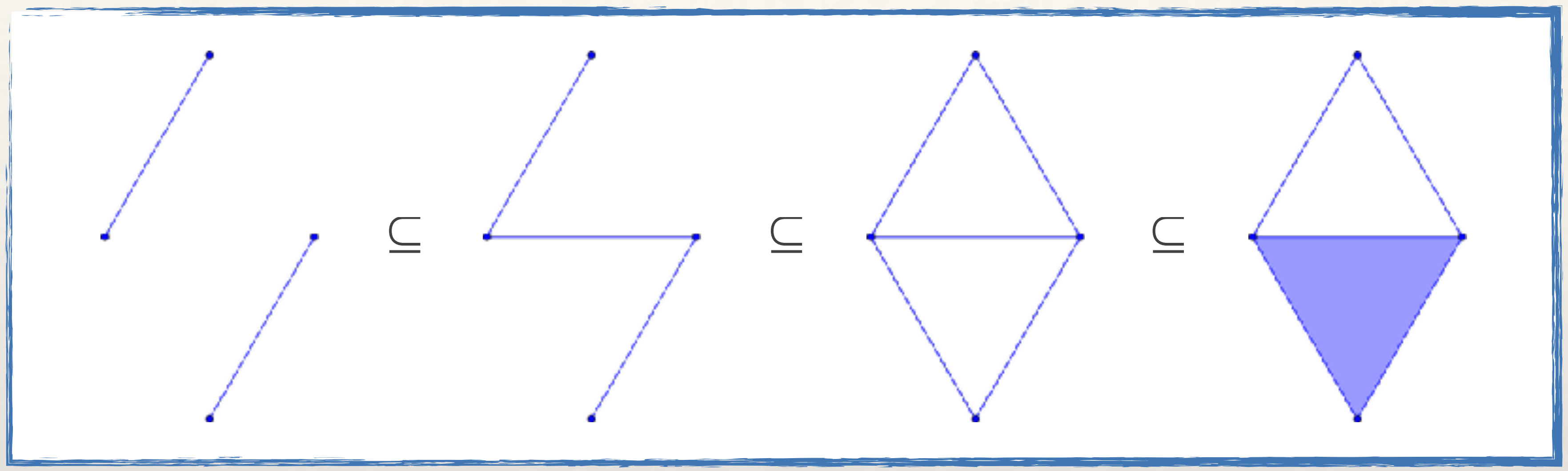






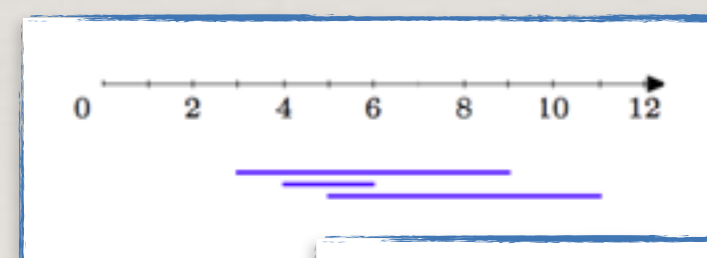
Shape Description

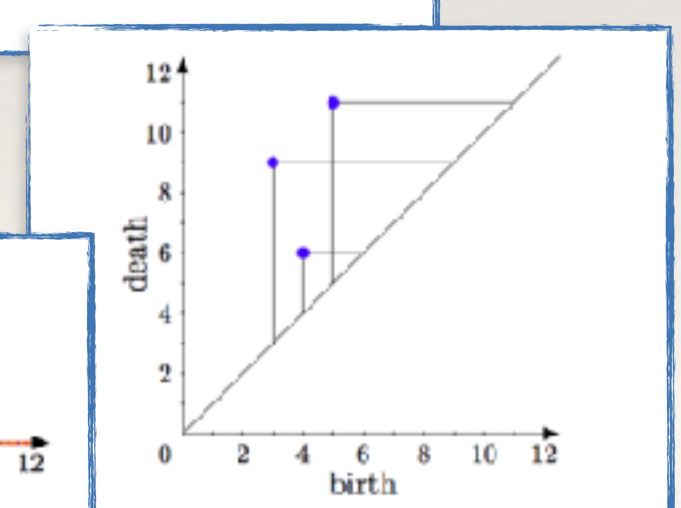
Given a filtered simplicial complex Σ,

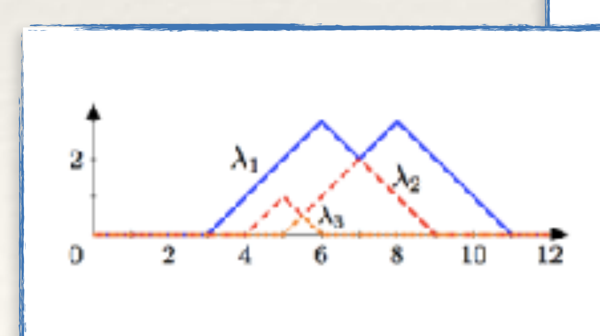


Persistent pairs of Σ can be visualized through:

- ***Barcodes*** [Carlsson et al. 2005; Ghrist 2008]
- ***Persistence diagrams*** [Edelsbrunner, Harer 2008]
- *Persistence landscapes* [Bubenik 2015]
- *Corner points and lines* [Frosini, Landi 2001]
- *Half-open intervals* [Edelsbrunner et al. 2002]
- *k-triangles* [Edelsbrunner et al. 2002]



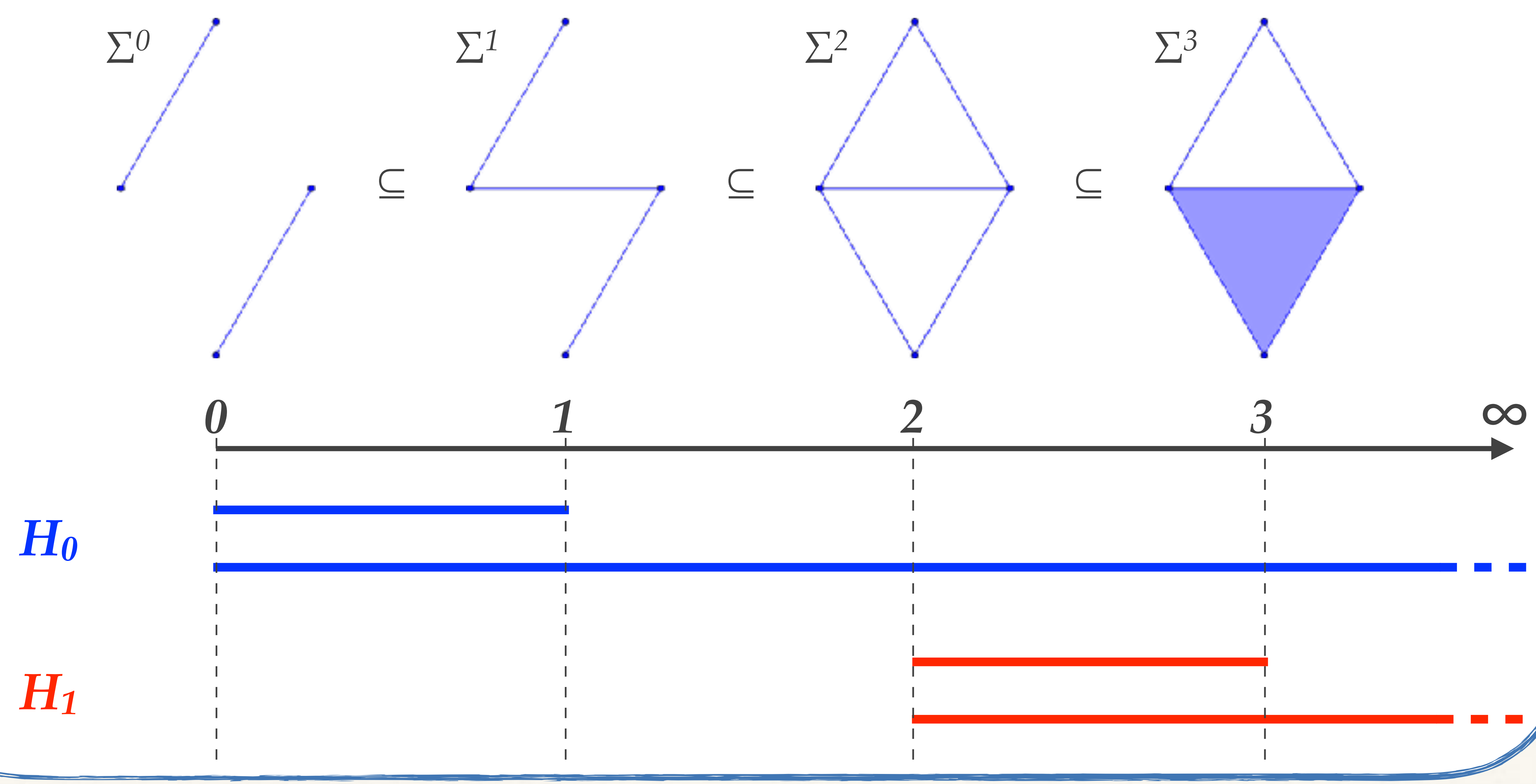




Shape Description

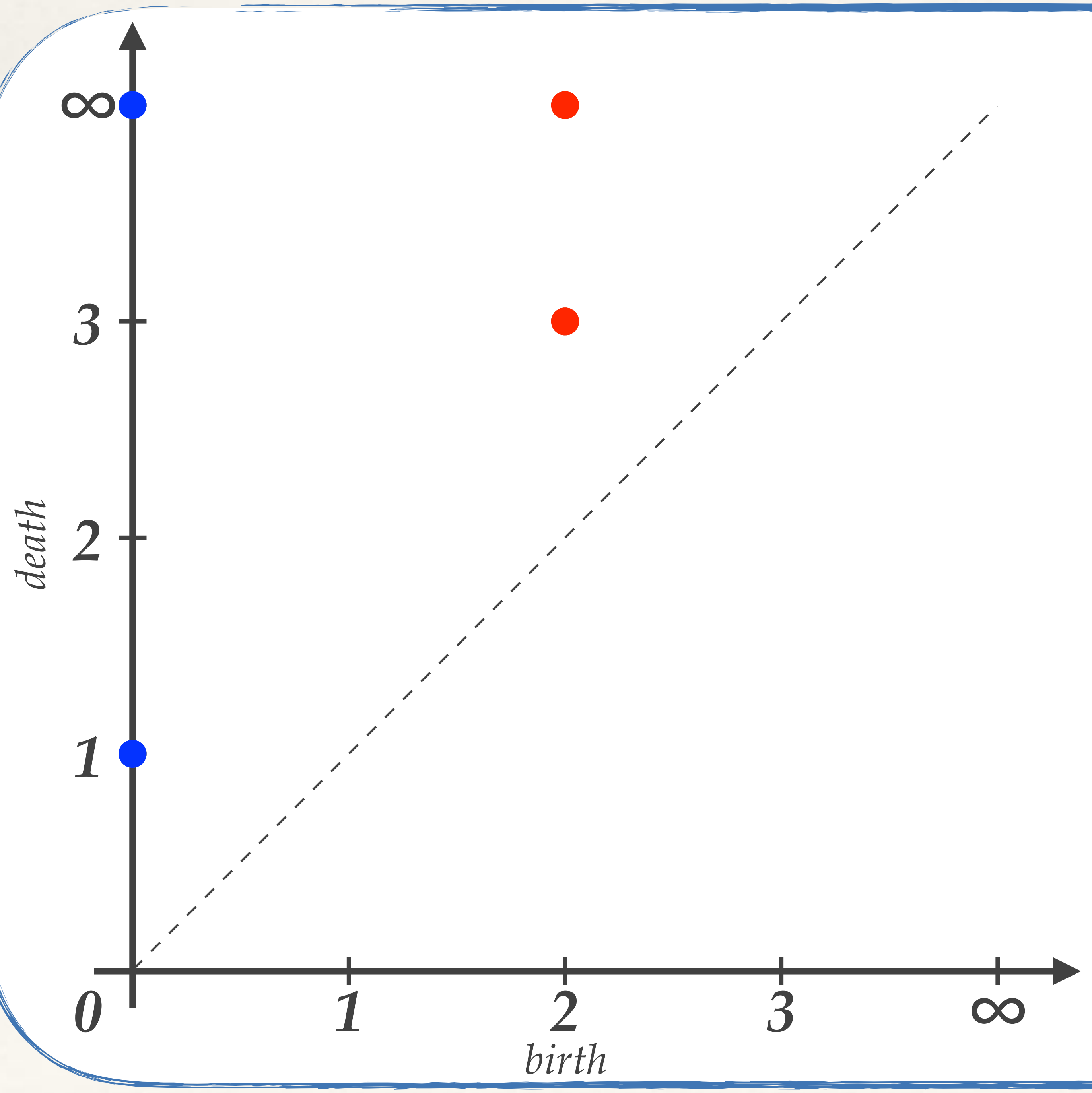
Barcodes:

Persistence pairs are represented as ***intervals in R***



Shape Description

Persistence Diagrams:



*Persistence pairs are represented as **points in R^2***

H_0 $(0, 1)$ $(0, \infty)$

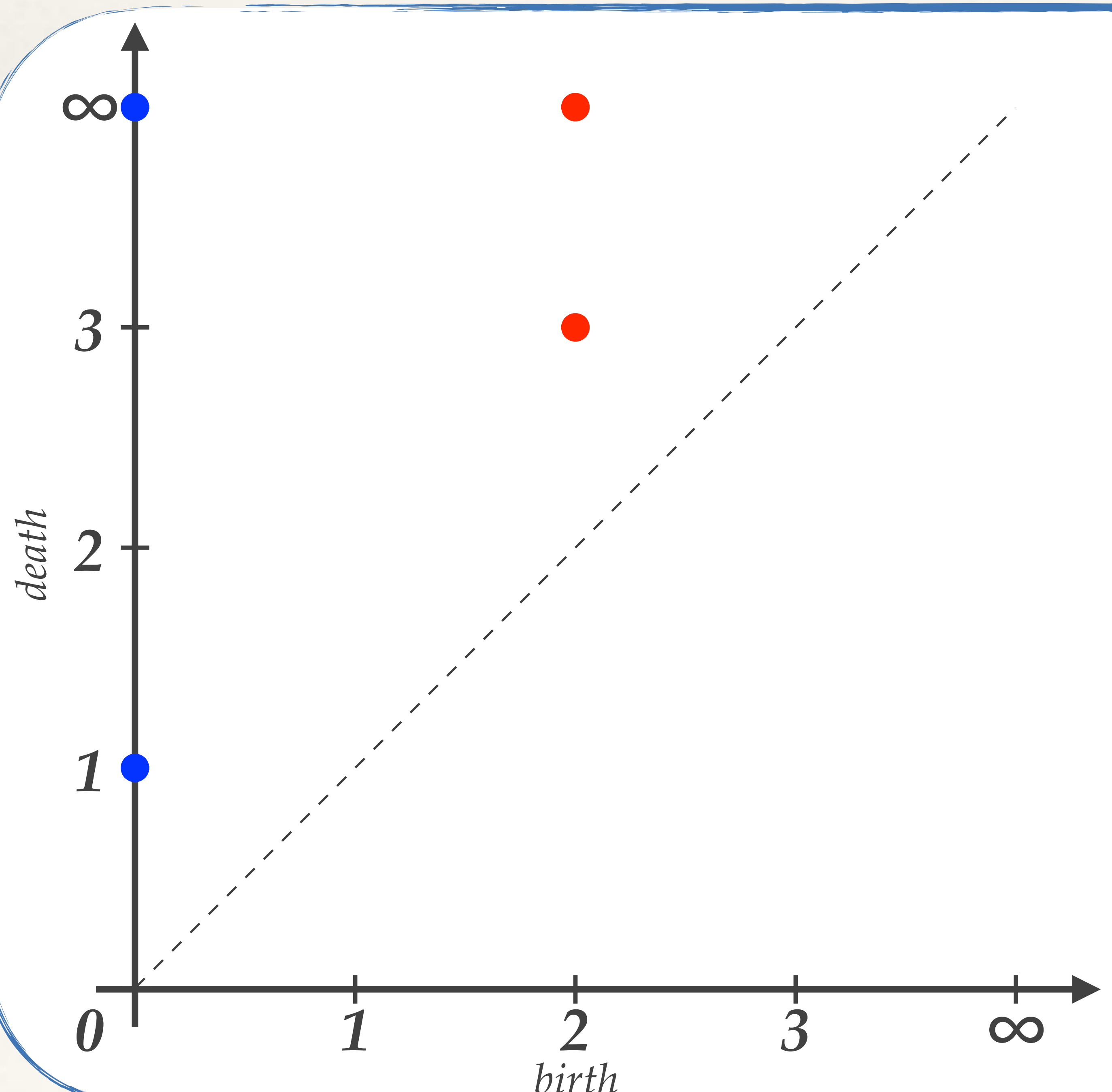
H_1 $(2, 3)$ $(2, \infty)$

Formally, a persistence diagram is a ***multiset***

- Points are endowed with **multiplicity**

Shape Description

Persistence Diagrams:



*Persistence pairs are represented as **points in** $\mathbb{R} \times (\mathbb{R} \cup \{\infty\})$*

H_0 $(0, 1)$ $(0, \infty)$

H_1 $(2, 3)$ $(2, \infty)$

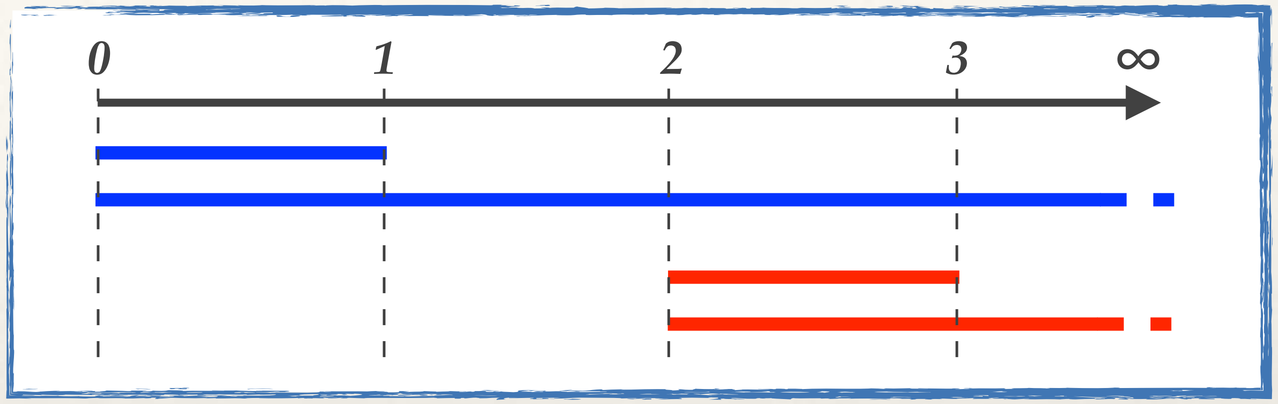
Formally, a persistence diagram is a ***multiset***

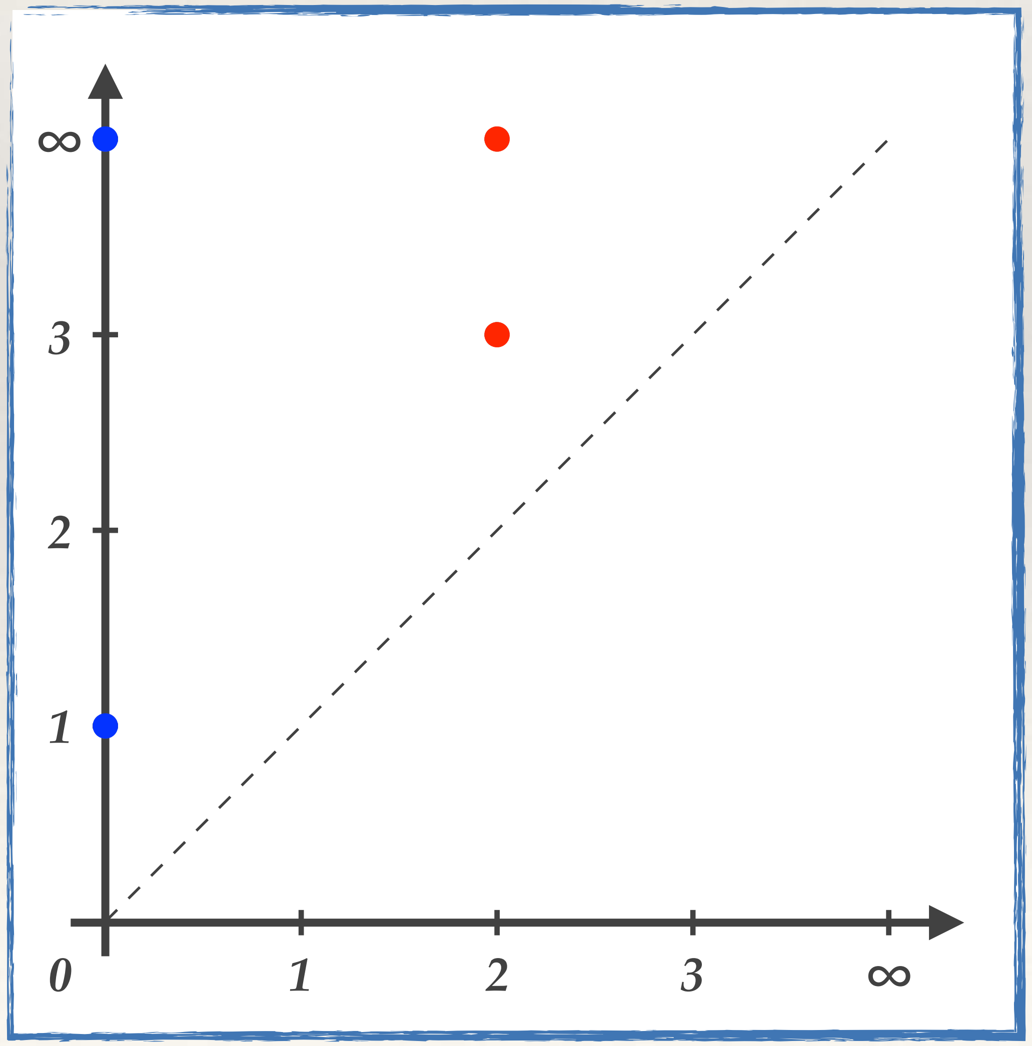
- Points are endowed with **multiplicity**

Shape Description

Both tools ***visually represent*** the ***lifespan*** of the homology classes:

- **Barcode:** *length of the intervals*
- **Persistence Diagram:** *distance from the diagonal*

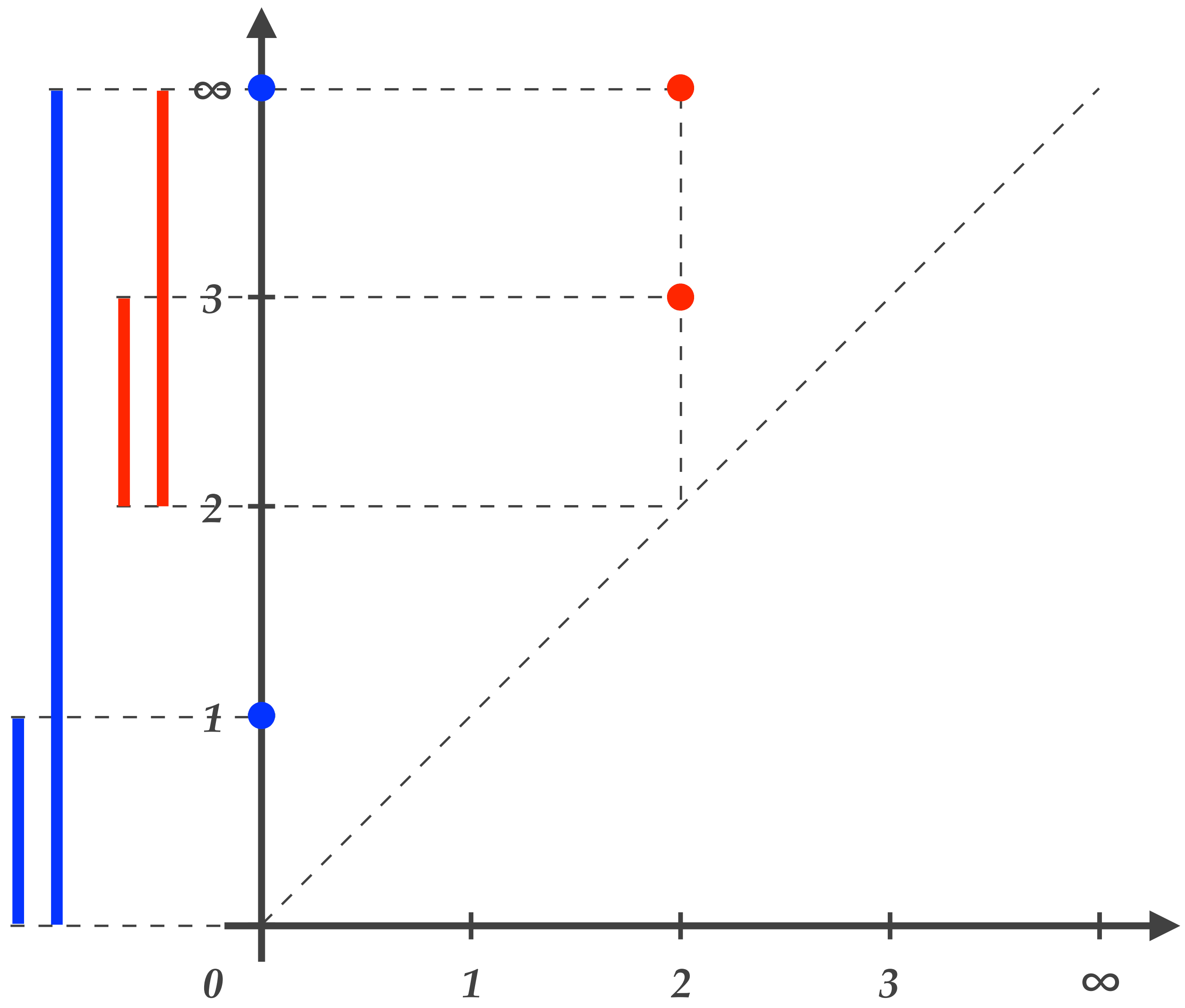




Barcodes and Persistence Diagrams **encode** ***equivalent*** **information**

Shape Description

Barcodes and Persistence Diagrams **encode *equivalent* information**



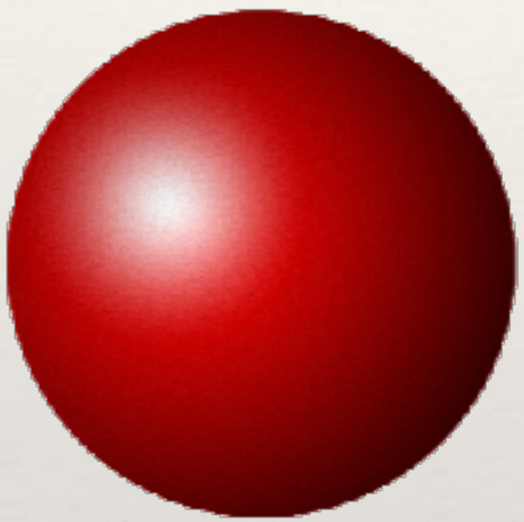
A visualization can be easily ***"translated"*** in the other one:

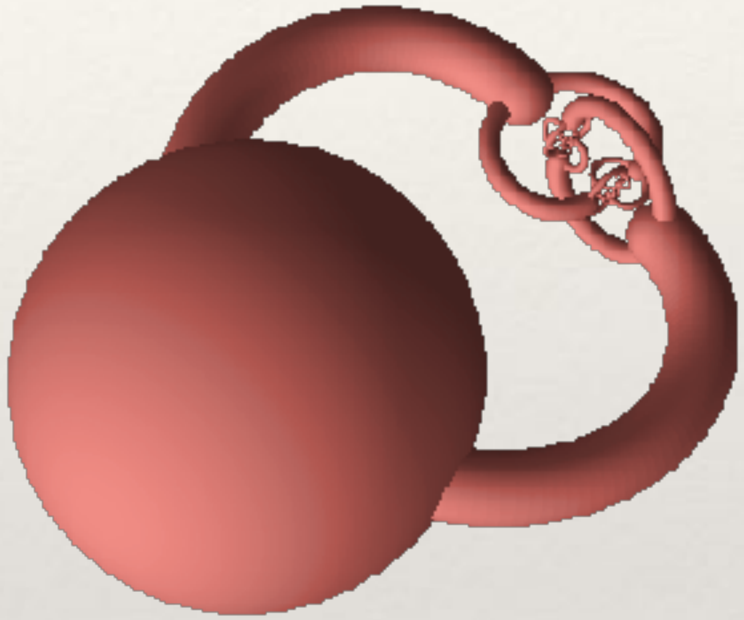
[p, q] ⟷ *(p, q)*

[p, ∞) ⟷ *(p, ∞)*

Shape Comparison

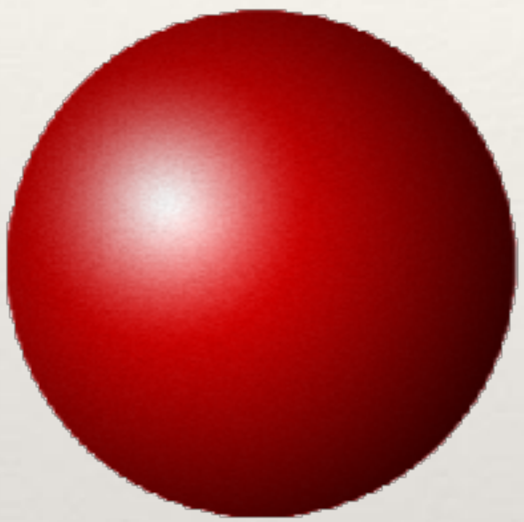
- *Do they have the same shape?*

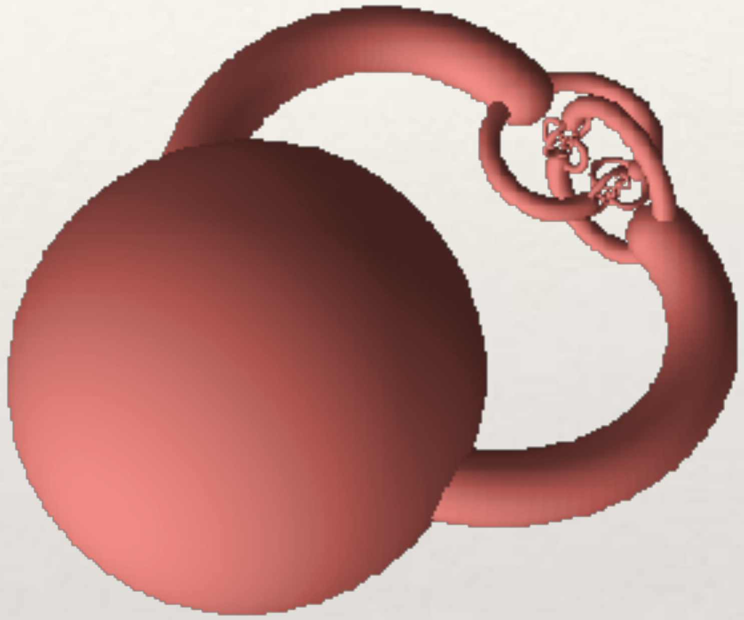




Shape Comparison

- *Do they have the same shape?*

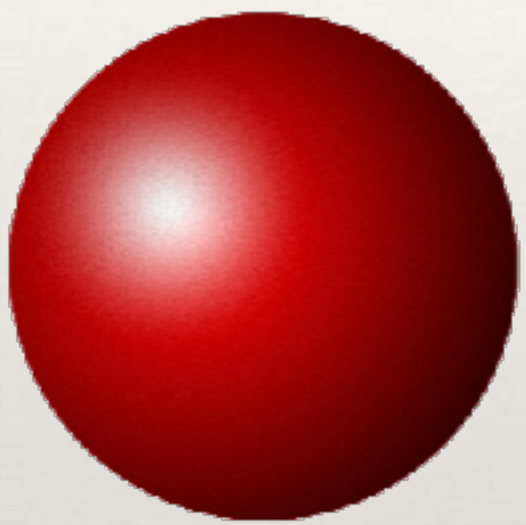


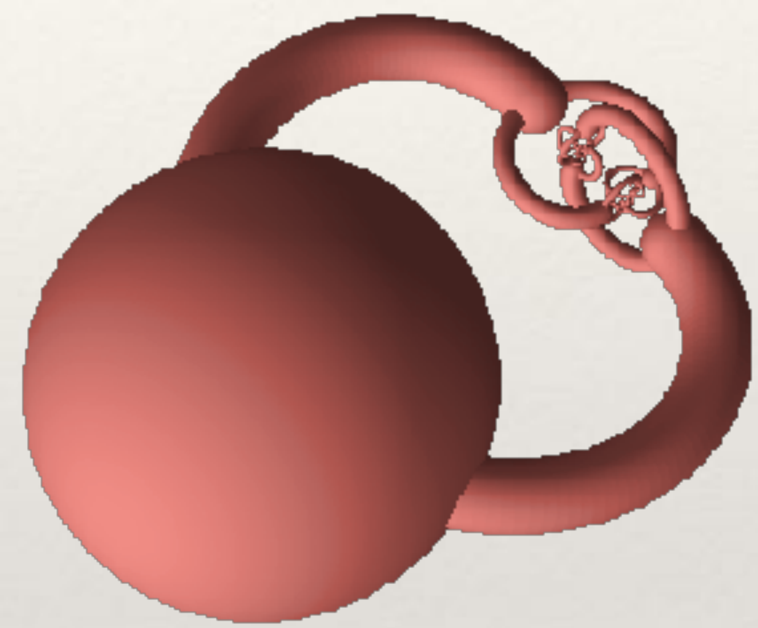


In Practice?	**In Theory?**

Shape Comparison

- *Do they have the same shape?*





In Practice?	In Theory?
✗	✓
	*They are **homeomorphic***

Shape Comparison

- *Do they have the same shape?*

Shape Comparison

- *Do they have the same shape?*

In Practice?	**In Theory?**

Shape Comparison

- *Do they have the same shape?*

In Practice?	**In Theory?**
✓	✗
	*They are **not homeomorphic***

Shape Comparison

It is possible to ***compare two shapes*** by comparing their ***homology groups***

Shape Comparison

It is possible to ***compare two shapes*** by comparing their ***homology groups***

Differently from homology, persistent homology provides
a notion of "shape" closer to our everyday perception

Shape Comparison

It is possible to ***compare two shapes*** by comparing their ***hom***… s

PERSISTENCE PAIRS

Differently from homology, persistent homology provides
a notion of "shape" closer to our everyday perception

Need for a notion of ***distance*** between sets of persistence pairs

Shape Comparison

Distances between Persistence Diagrams: [Cohen-Steiner et al. 2007]

Let X, Y be two persistence diagrams (points of the main diagonal are included with infinite multiplicity)

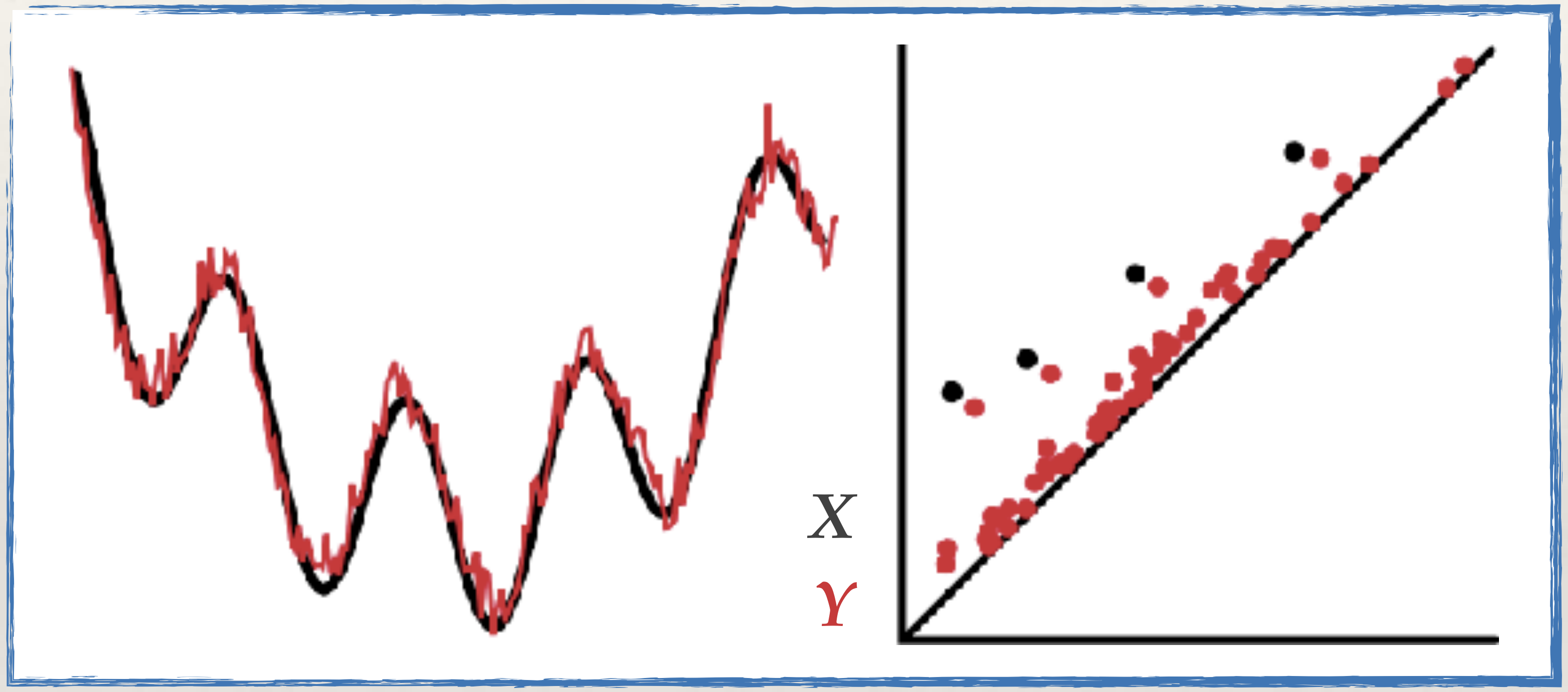


Image from [Rieck 2016]

Shape Comparison

Distances between Persistence Diagrams: [Cohen-Steiner et al. 2007]

Let X, Y be two persistence diagrams (points of the main diagonal are included with infinite multiplicity)

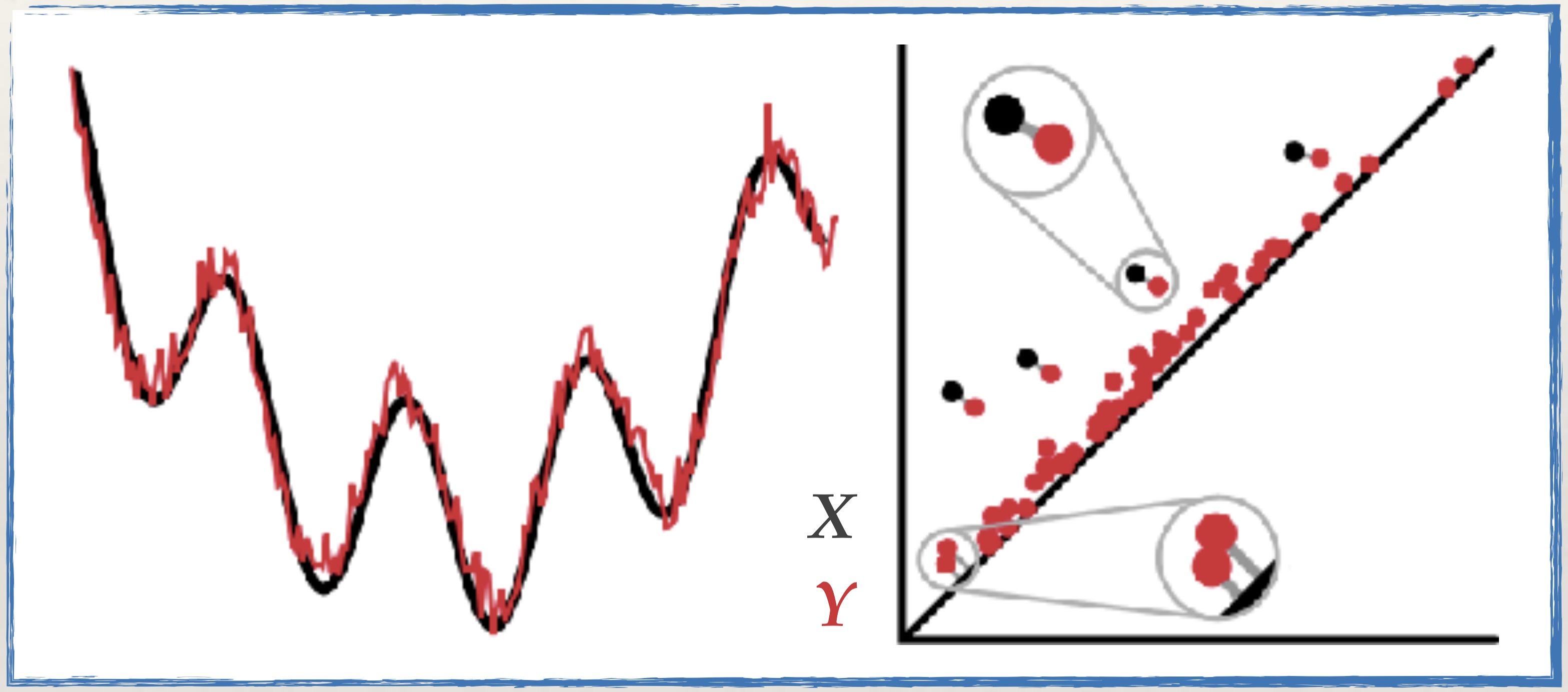


Image from [Rieck 2016]

- ***Bottleneck distance***

$$d_B(X,Y) = \inf_{\gamma} \sup_{x} \|x - \gamma(x)\|_{\infty}$$

Shape Comparison

Distances between Persistence Diagrams: [Cohen-Steiner et al. 2007]

Let X, Y be two persistence diagrams (points of the main diagonal are included with infinite multiplicity)

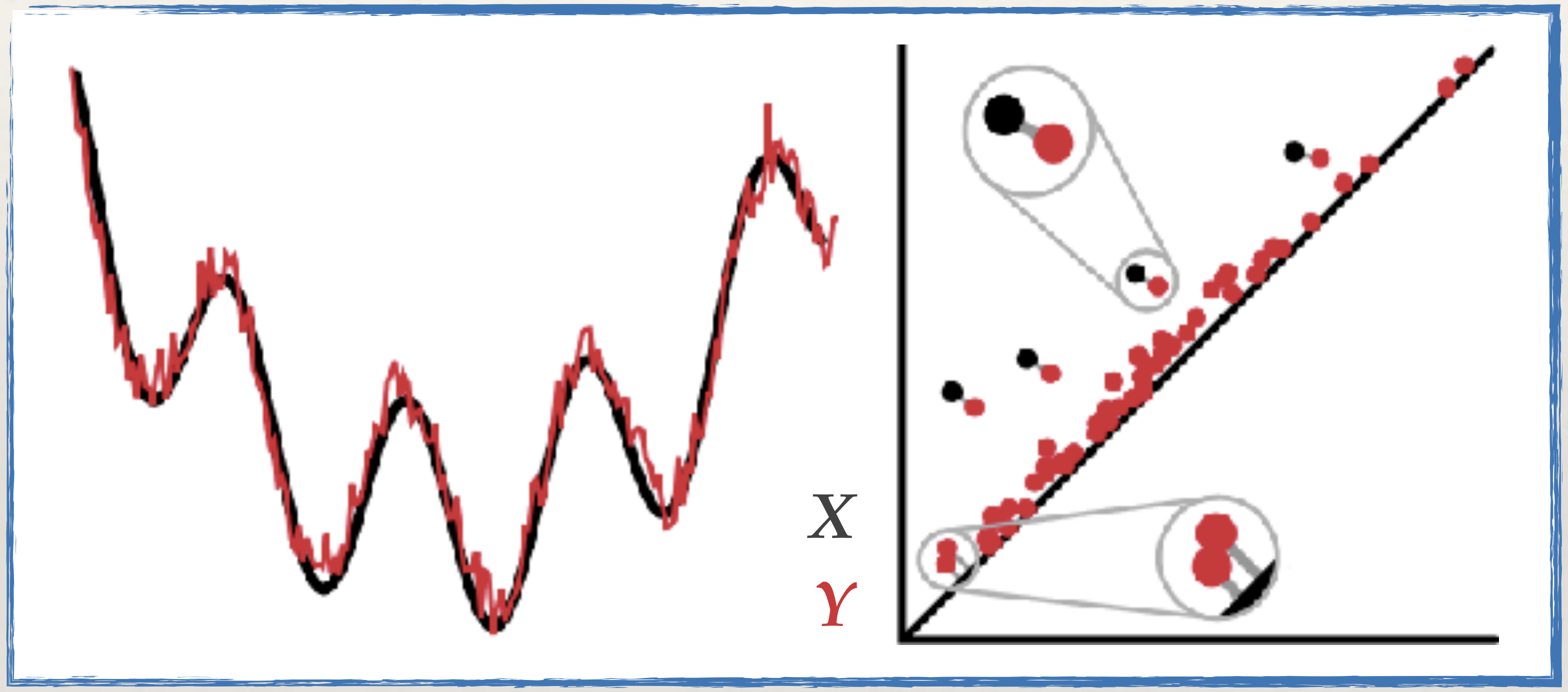


Image from [Rieck 2016]

- *Bottleneck distance*
- ***Wasserstein distance***

$$d_W^q(X,Y) = \left(\inf_\gamma \sum_x \|x-\gamma(x)\|_\infty^q\right)^{1/q}$$

$$d_W^\infty = d_B$$

Shape Comparison

Distances between Persistence Diagrams: [Cohen-Steiner et al. 2007]

Let X, Y be two persistence diagrams (points of the main diagonal are included with infinite multiplicity)

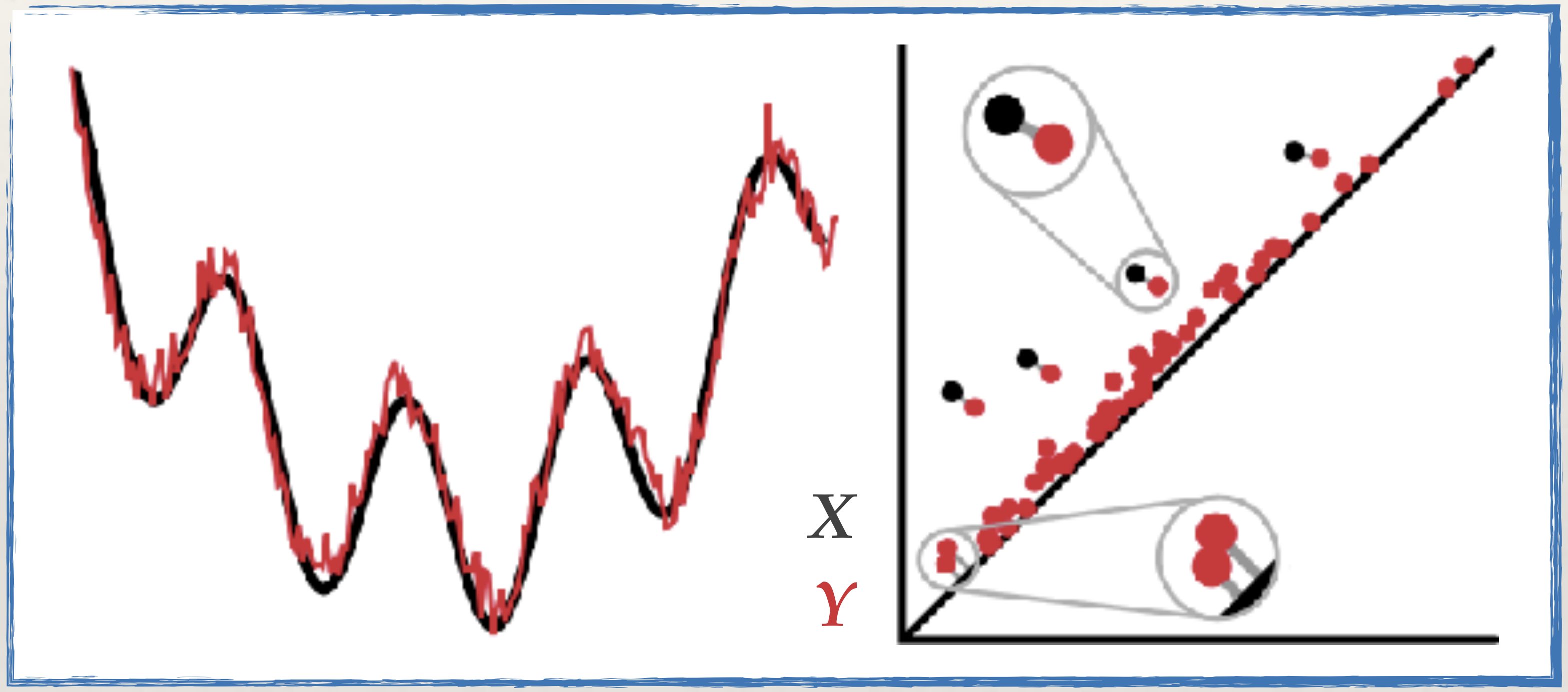


Image from [Rieck 2016]

- *Bottleneck distance*
- *Wasserstein distance*
- ***Hausdorff distance***

$$d_H(X,Y) = \max\left\{ \sup_x \inf_y \|x-y\|_\infty, \sup_y \inf_x \|y-x\|_\infty \right\}$$

$$d_H \leq d_B$$

Shape Comparison

Distances between Persistence Diagrams: [Cohen-Steiner et al. 2007]

Let X, Y be two persistence diagrams (points of the main diagonal are included with infinite multiplicity)

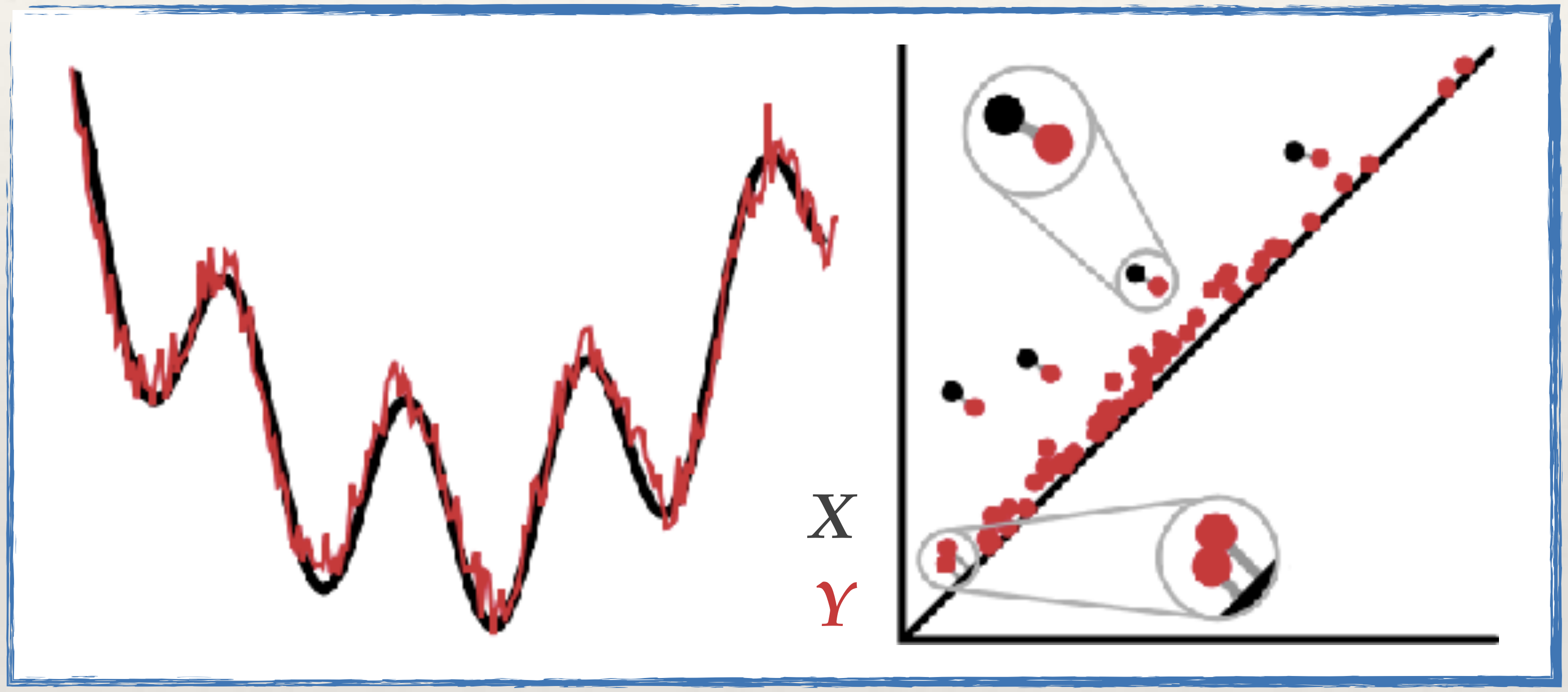


Image from [Rieck 2016]

- *Bottleneck distance*
- *Wasserstein distance*
- *Hausdorff distance*

Stability:

Similar shapes have ***similar persistence diagrams***?

Outline

Describing a Shape through Persistence Pairs

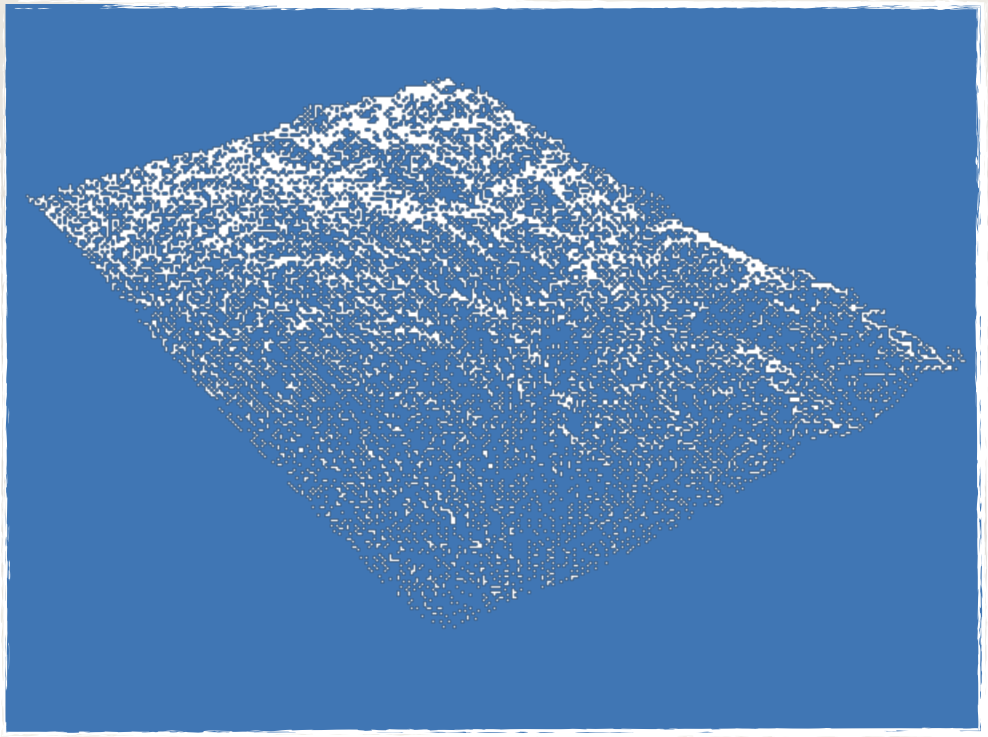
From a Point Cloud to a Filtered Simplicial Complex

From a Point Cloud To a Complex

Point Cloud Datasets:

More and more, data consist of **point clouds**:

- ***finite set of points V in R^d*** *(more generally, embedded in a metric space)*



Coordinates

actual ***geometric position***

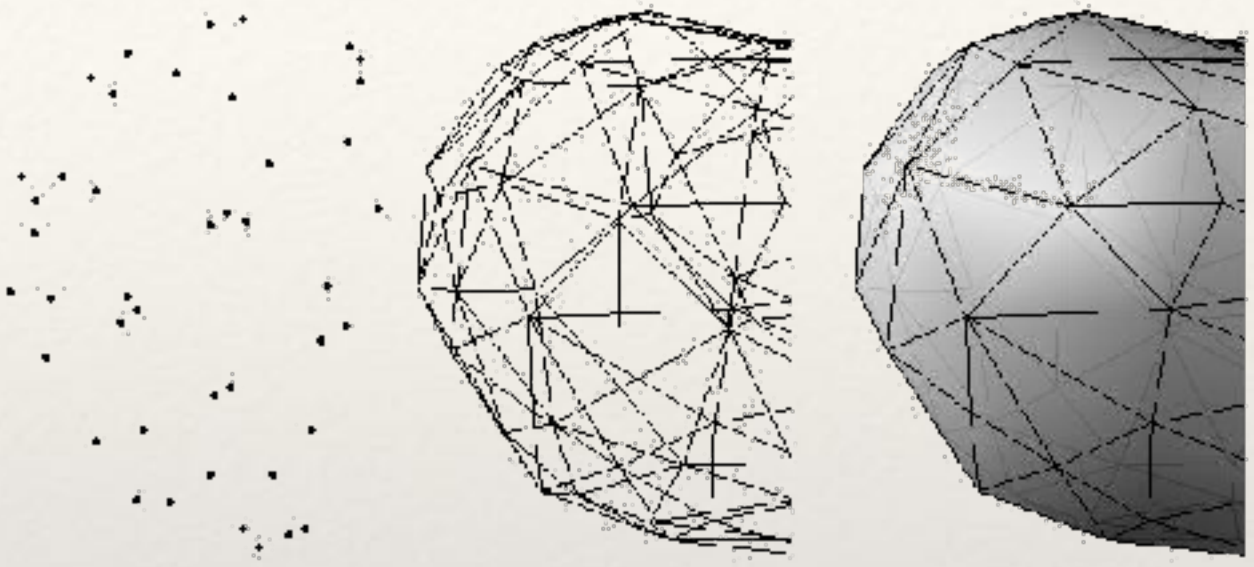
values of ***attributes*** *attached to each point*

We represent these ***unorganized, large-size*** and ***high-dimensional*** *data* through **simplicial complexes**

From a Point Cloud To a Complex

Various techniques can lead to

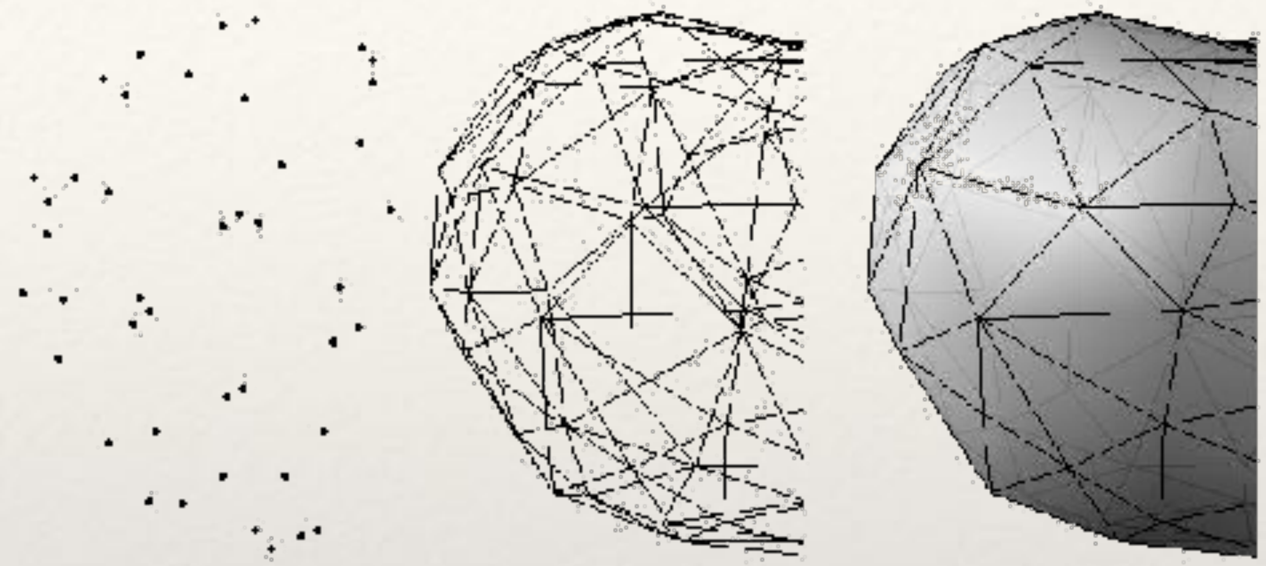
- *simplicial complex*
- *filtered simplicial complex*



From a Point Cloud To a Complex

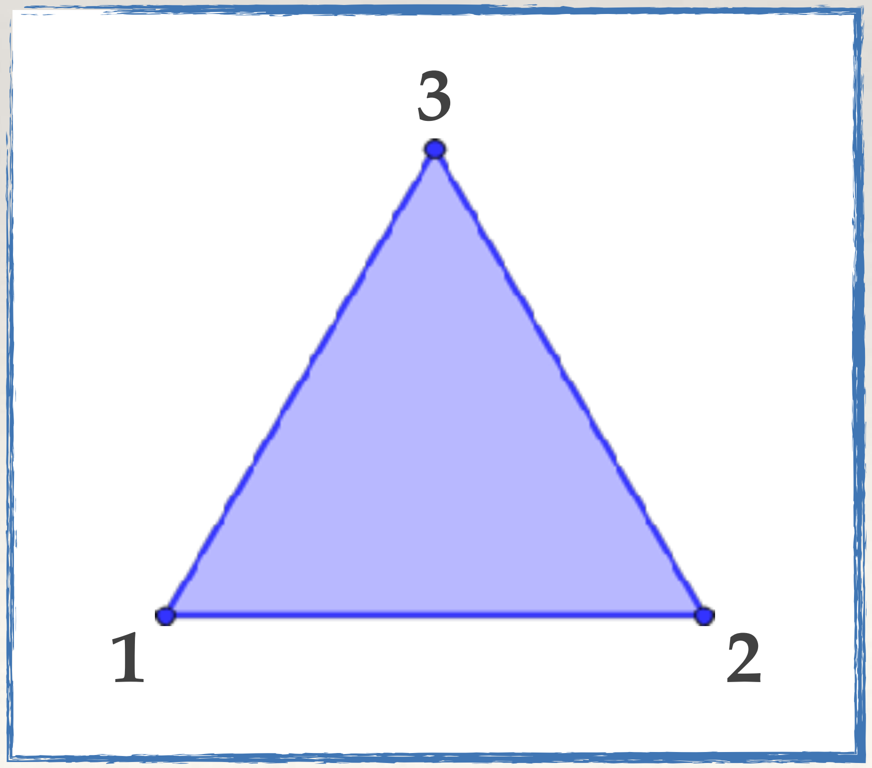
Various techniques can lead to

- ***simplicial complex***
- ***filtered simplicial complex***



Vertex-based Filtration:

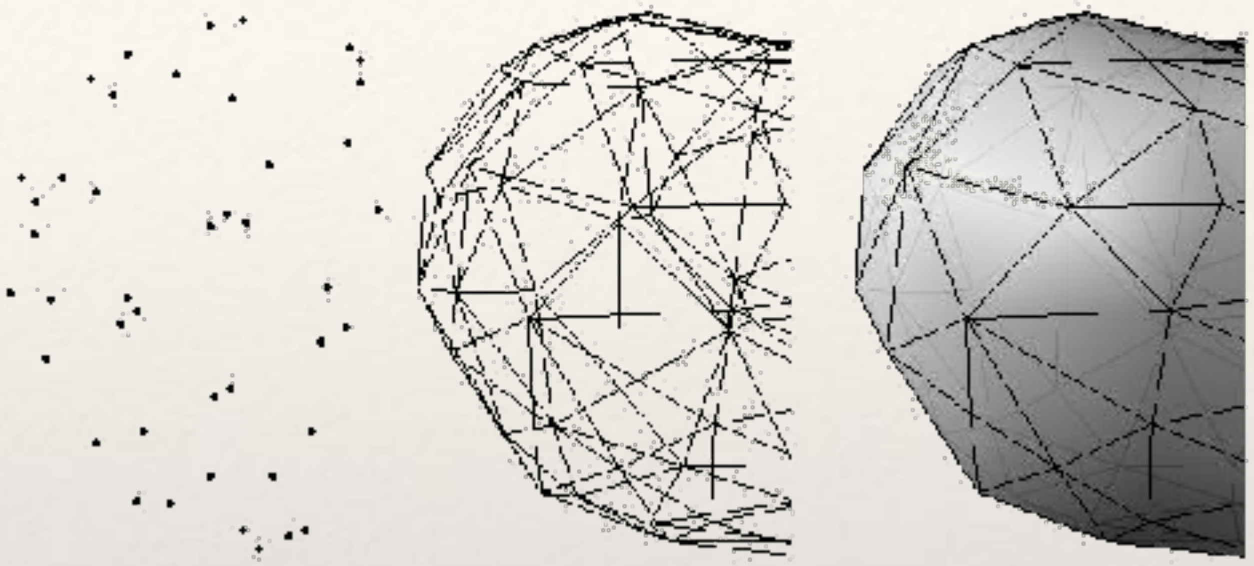
$F : V \to \mathbb{N}$ induces a filtration on Σ



From a Point Cloud To a Complex

Various techniques can lead to

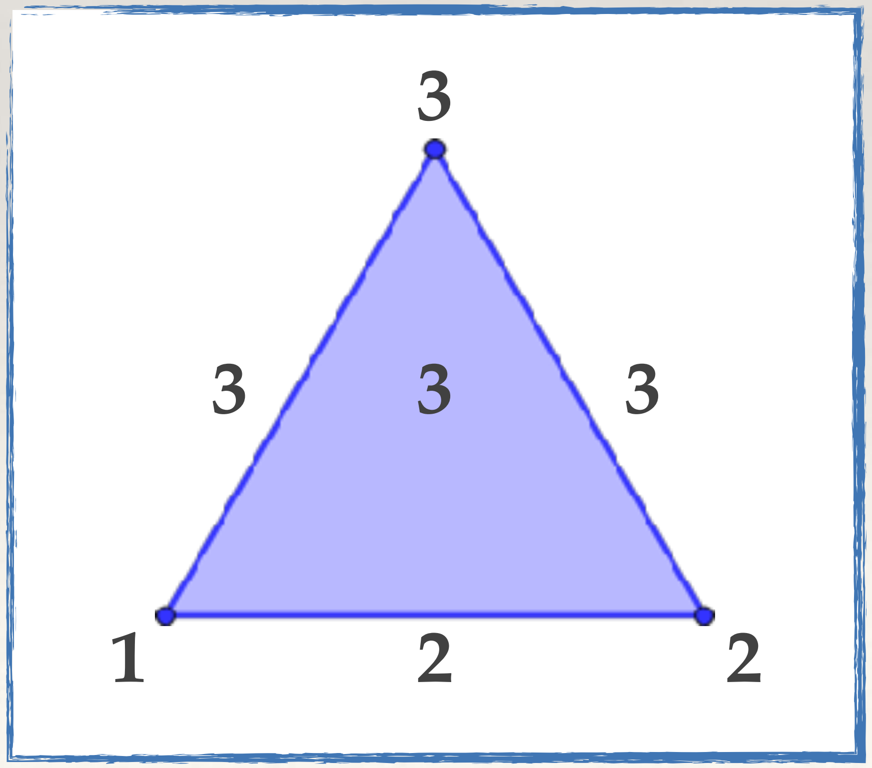
- ***simplicial complex***
- ***filtered simplicial complex***



Vertex-based Filtration:

$F : V \to \mathbb{N}$ induces a filtration on Σ

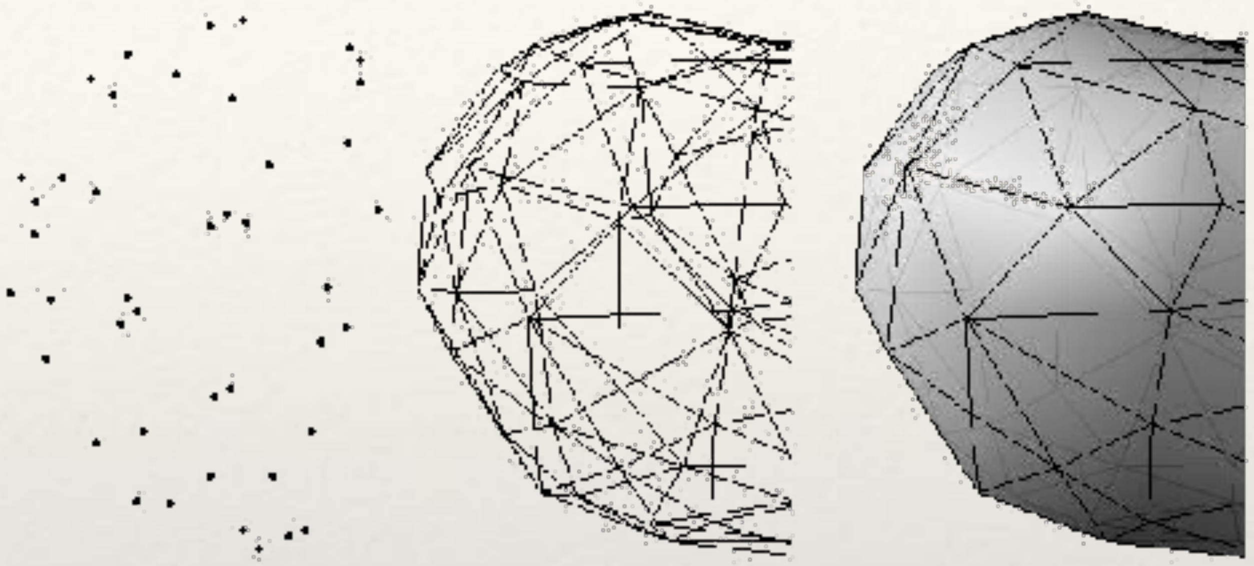
- $$F(\sigma) := \max_{v \in \sigma}\{F(v)\}$$
- $$\Sigma_p := \{\sigma \in \Sigma \,|\, F(\sigma) \leq p\}$$



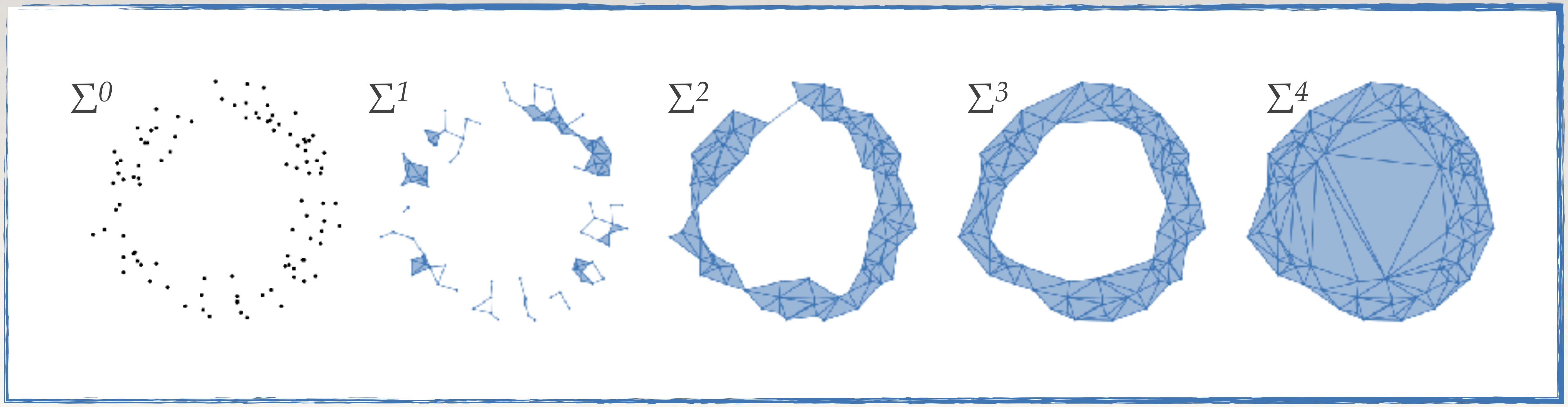
From a Point Cloud To a Complex

Various techniques can lead to

- *simplicial complex*
- ***filtered simplicial complex***



Multi-scale Representation:



From a Point Cloud To a Complex

Standard Constructions:

- *Delaunay* ***triangulations***
  - *Voronoi diagrams*
- *Čech* ***complexes***
- *Vietoris-Rips* ***complexes***
- *Alpha-shapes*
- *Witness* ***complexes***

References:

H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, 1987

H. Edelsbrunner, *Geometry and Topology for Mesh Generation*, 2001

From a Point Cloud To a Complex

Given a finite set of points V in R^d :

	Output	Dimension
Delaunay triangulation	**Simplicial Complex**	d
Čech complex	Filtered Simplicial Complex	Arbitrary (up to $\lvert V \rvert$ -1)
Vietoris-Rips complex	Filtered Simplicial Complex	Arbitrary (up to $\lvert V \rvert$ -1)
Alpha-shapes	Filtered Simplicial Complex	d
Witness complexes	Filtered Simplicial Complex	Arbitrary (up to $\lvert V \rvert$ -1)

From a Point Cloud To a Complex

Two Fundamental Notions:

Nerve Complex

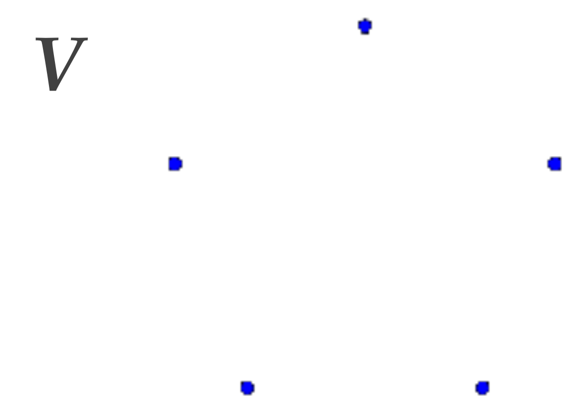
Abstract Simplicial Complex

From a Point Cloud To a Complex

Given a finite set V,

An **abstract simplicial complex** Σ on V is a ***collection of finite subsets of V*** such that:

- if $\tau \in \Sigma$, $\sigma \subseteq \tau$, then $\sigma \in \Sigma$

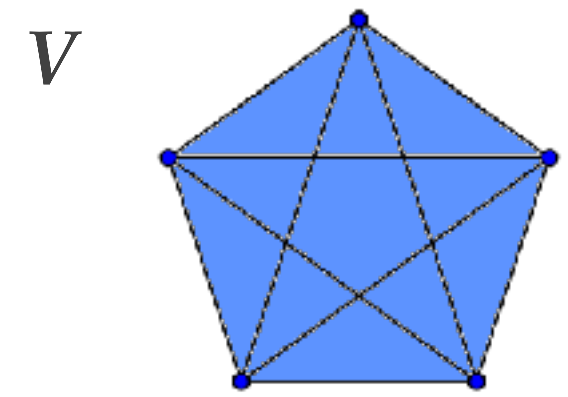


From a Point Cloud To a Complex

Given a finite set V,

An **abstract simplicial complex** Σ on V is a ***collection of finite subsets of V*** such that:

- if $\tau \in \Sigma$, $\sigma \subseteq \tau$, then $\sigma \in \Sigma$

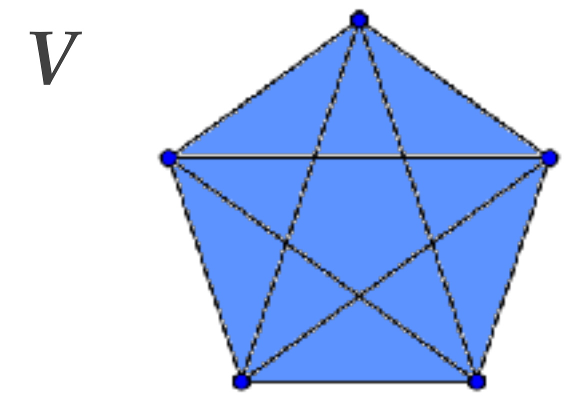


From a Point Cloud To a Complex

Given a finite set V,

An **abstract simplicial complex** Σ on V is a ***collection of finite subsets of V*** such that:

- if $\tau \in \Sigma$, $\sigma \subseteq \tau$, then $\sigma \in \Sigma$



Properties:

- Any simplicial complex is an ***abstract simplicial complex on the set of its vertices***
- Any abstract simplicial complex admits a ***geometrical realization in R^n***

From a Point Cloud To a Complex

Nerve Complex:

Given a finite collection S of closed sets in $\mathbf{R}^{\mathbf{d}}$,

the **nerve of S** is the ***abstract simplicial complex***
generated by the ***non-empty common intersections***

Formally,

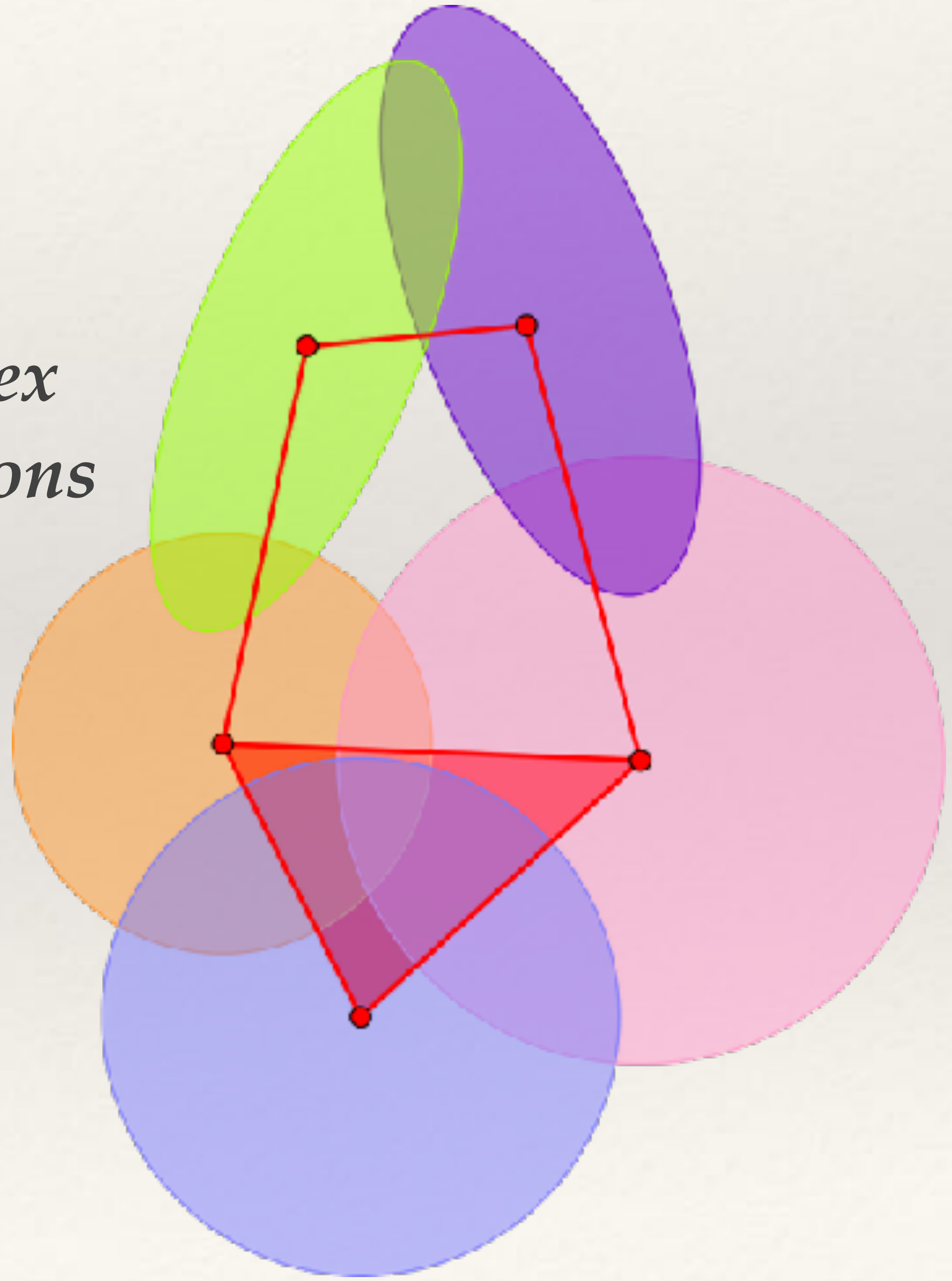
$$Nrv(S) := \{\sigma \subseteq S \mid \bigcap \sigma \neq \emptyset\}$$

From a Point Cloud To a Complex

Nerve Complex:

Given a finite collection S of closed sets in $\mathbf{R}^d$,

the **nerve of S** is the ***abstract simplicial complex*** generated by the ***non-empty common intersections***



Formally,

$$Nrv(S) := \{\sigma \subseteq S \mid \bigcap \sigma \neq \emptyset\}$$

From a Point Cloud To a Complex

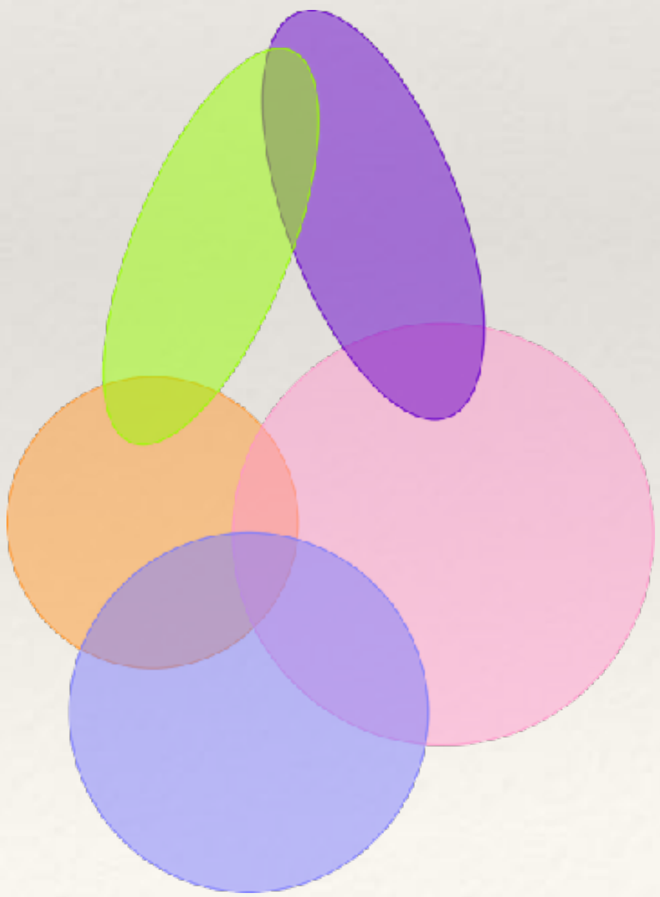
Nerve Theorem:

Let S be a finite collection of closed, **convex** sets in $\mathbf{R}^d$

Then, the nerve of S and the union of the sets in S have the same homotopy type

Same Homotopy Type

Isomorphic Homology

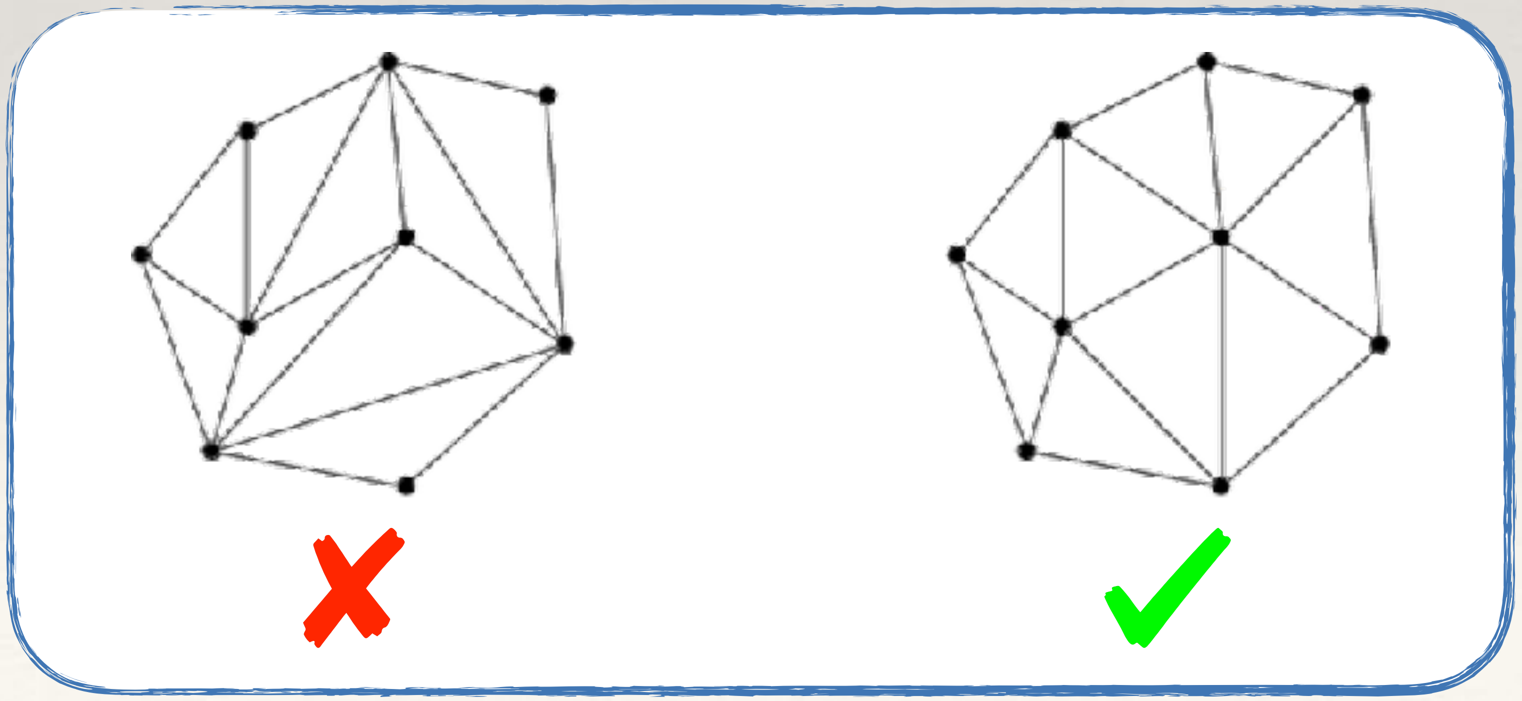


Delaunay Triangulation

Given a finite set of points V in $\mathbf{R}^2$,

Delaunay Triangulation is a classic notion in Computational Geometry:

- Producing a ***"nice" triangulation*** of V
  - free of long and skinny triangles
- Named after ***Boris Delaunay*** for his work on this topic from 1934
- Originally defined for sets of points in a ***plane***

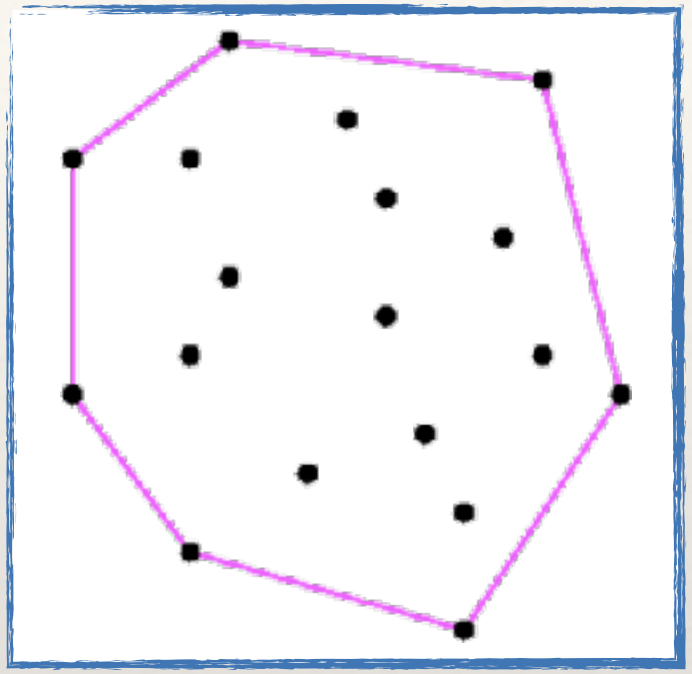


Delaunay Triangulation

Given a finite set of points *V* in $\mathbf{R}^2$,

Convex Hull of *V*:

The ***smallest convex*** subset ***CH(V)*** of $\mathbf{R}^2$ containing all the points of *V*



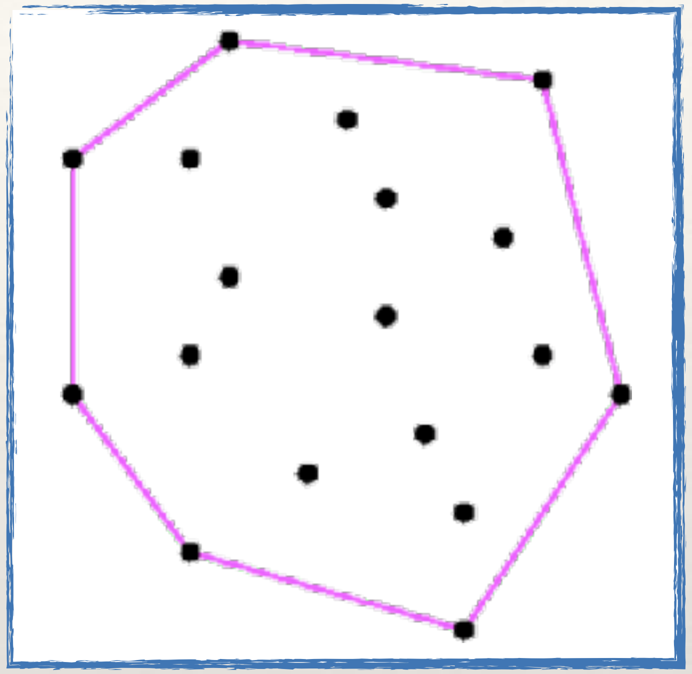
Images from [De Floriani 2003]

Delaunay Triangulation

Given a finite set of points V in $\mathbf{R}^2$,

Convex Hull of V:

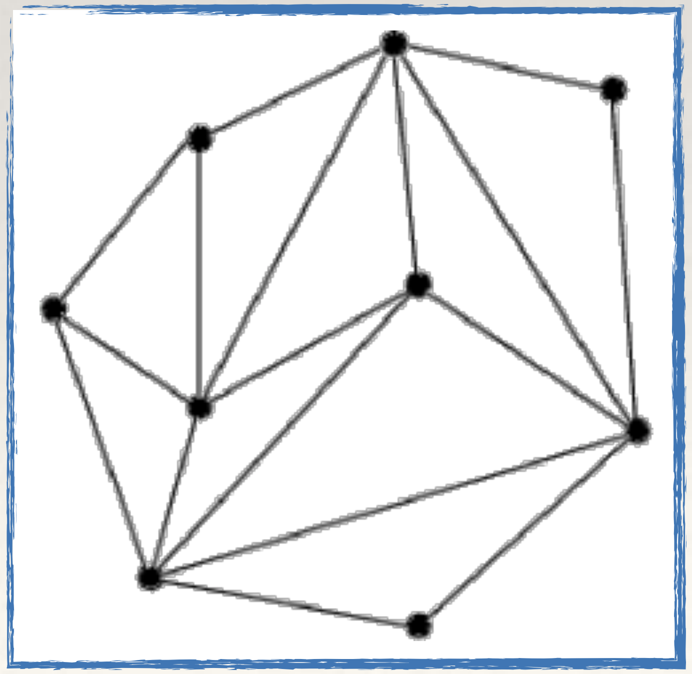
The ***smallest convex*** subset $CH(V)$ of $\mathbf{R}^2$ containing all the points of V



Triangulation of V:

A *2-dimensional simplicial complex* $\Sigma(V)$ such that:

- The domain of Σ is $CH(V)$
- The 0-simplices of Σ are the points in V

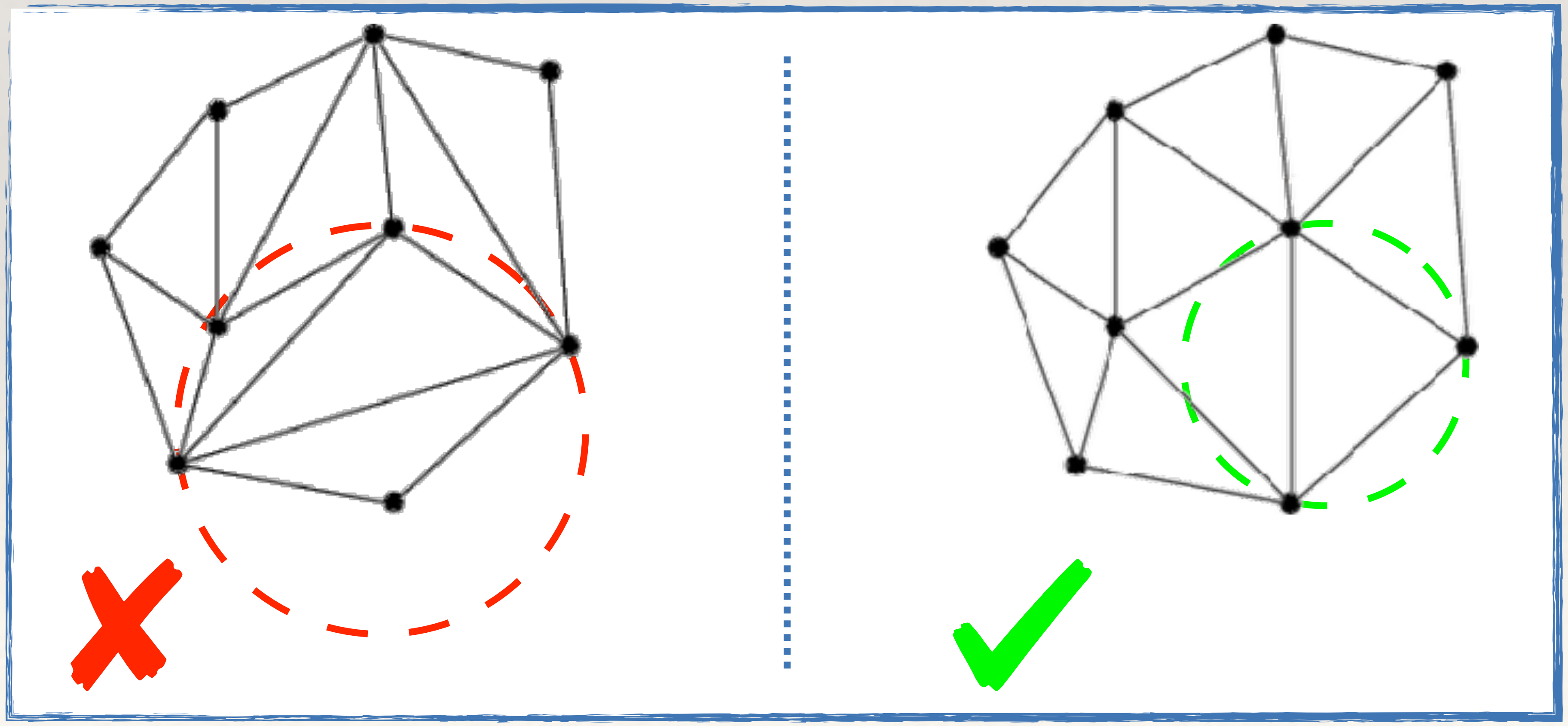


Images from [De Floriani 2003]

Delaunay Triangulation

Definition:

A **Delaunay triangulation** is a triangulation ***Del(V)*** of *V* such that:
the **circumcircle of any triangle** does **not contain any point** of *V* in its interior



Images from [De Floriani 2003]

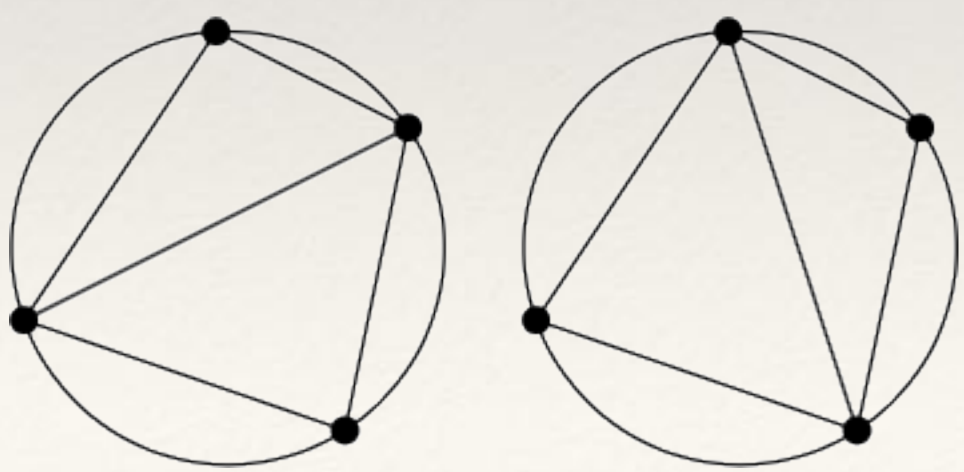
Delaunay Triangulation

A finite set of points V in $\mathbf{R}^d$ is **in general position** if *no d+2 of the points lie on a common (d − 1)-sphere*

For d=2,

V in general position ⟺ *no four or more points are co-circular*

Uniqueness: If V is in general position, then $Del(V)$ is **unique**



Delaunay Triangulation

Voronoi Region:

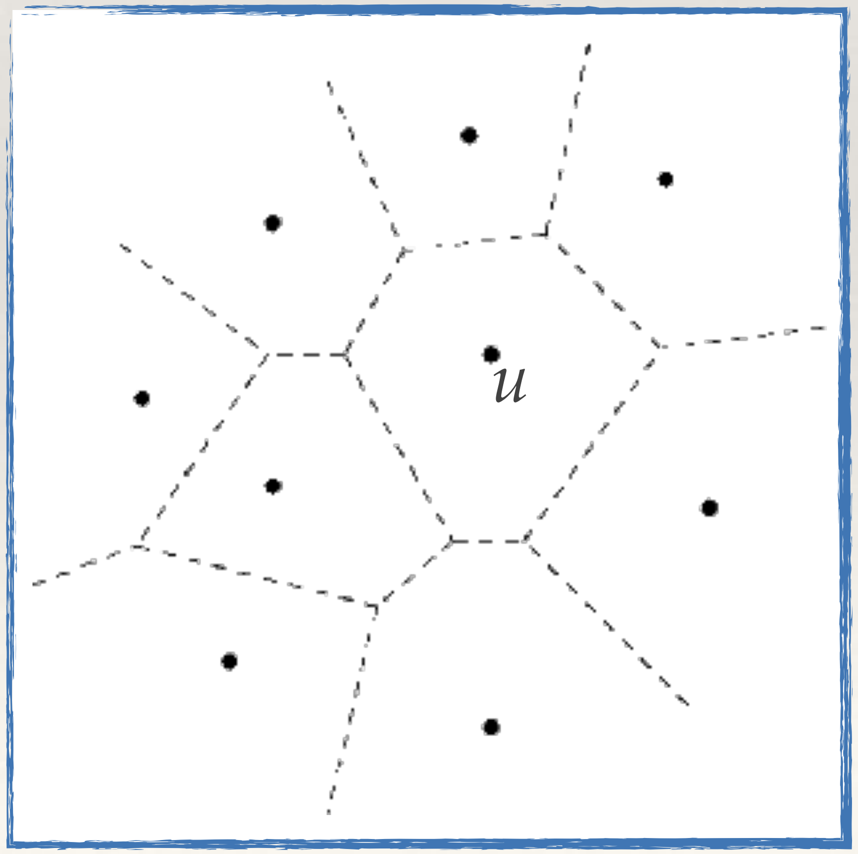
The ***Voronoi region*** of u in V is the set of points of **R²** for which u is the closest

$$R_V(u) = \{x \in \mathbb{R}^d \,|\, d(x,u) \leq d(x,v), v \in V\}$$

- Any Voronoi region is a ***convex*** closed subset of **R²**
- A Voronoi region is ***not necessarily bounded***

Voronoi Diagram:

The ***Voronoi diagram*** is the collection ***Vor(V)*** of the Voronoi regions of the points of V



Images from [De Floriani 2003]

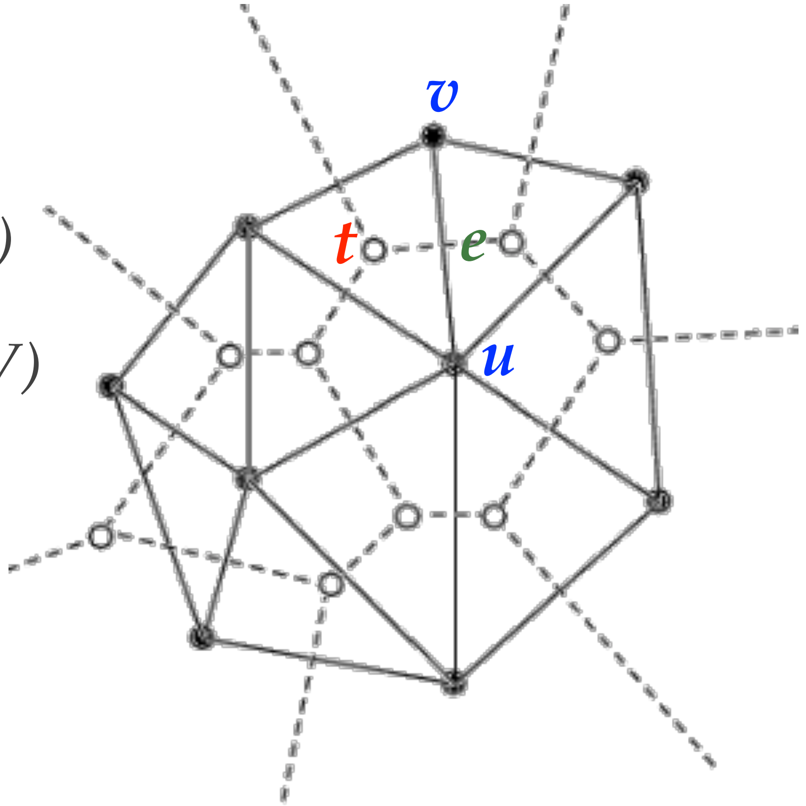
Delaunay Triangulation

Duality Property:

If V is in general position, then

the Delaunay triangulation coincides with the nerve of the Voronoi diagram

$$Del(V) = \{\sigma \subseteq V \mid \bigcap_{u \in \sigma} R_V(u) \neq \emptyset\}$$



- Every **point** ***u*** of V corresponds to a Voronoi region $R_V(u)$
- Every **triangle** ***t*** of $Del(V)$ correspond to a vertex in $Vor(V)$
- Every **edge** ***e=(u,v)*** in $Del(V)$ corresponds to an edge shared by the two Voronoi regions $R_V(u)$ and $R_V(v)$

Images from [De Floriani 2003]

Delaunay Triangulation

Algorithms:

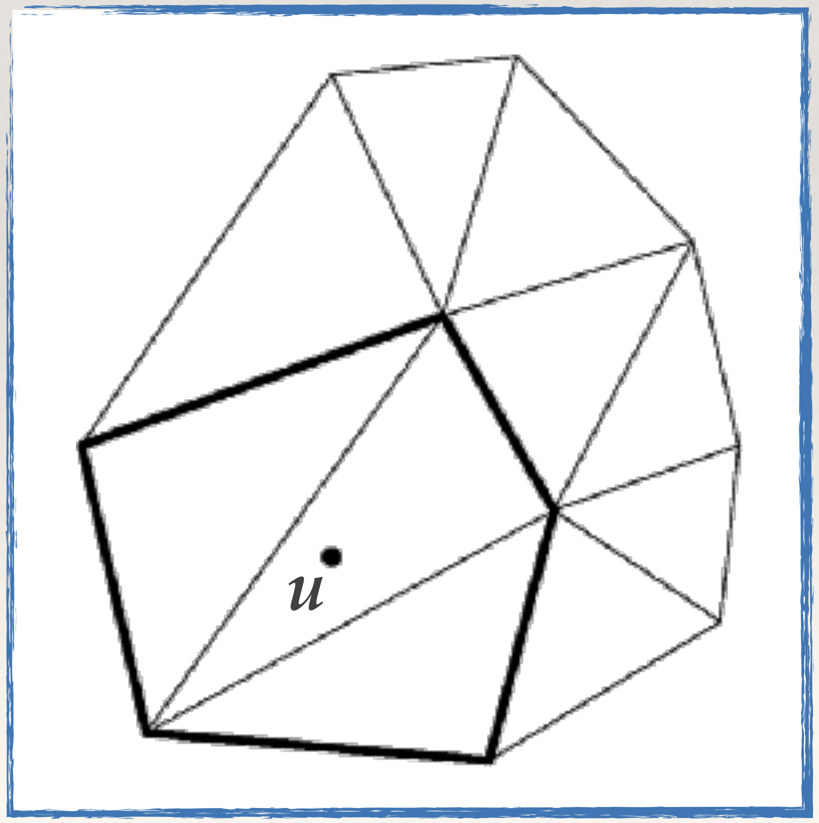
- ***Two-step algorithms***:
  - Computation of an arbitrary triangulation Σ'
  - Optimization of Σ' to produce a Delaunay triangulation
- ***Incremental algorithms*** [Guibas, Stolfi 1983; Watson 1981]:
  - Modification of an existing Delaunay triangulation while adding a new vertex at a time
- ***Divide-and-conquer algorithms*** [Shamos 1978; Lee, Schacter 1980]:
  - Recursive partition of the point set into two halves
  - Merging of the computed partial solutions
- ***Sweep-line algorithms*** [Fortune 1989]:
  - Step-wise construction of a Delaunay triangulation while moving a sweep-line in the plane

Delaunay Triangulation

Watson's Algorithm:

A Delaunay triangulation is computed by **incrementally adding a single point** to an existing Delaunay triangulation

Let V_i be a subset of V and let u be a point in $V \setminus V_i$



Input:

Del(V$_i$), a Delaunay triangulation of V_i

Output:

Del(V$_{i+1}$), a Delaunay triangulation of $V_{i+1} := V_i \cup \{u\}$

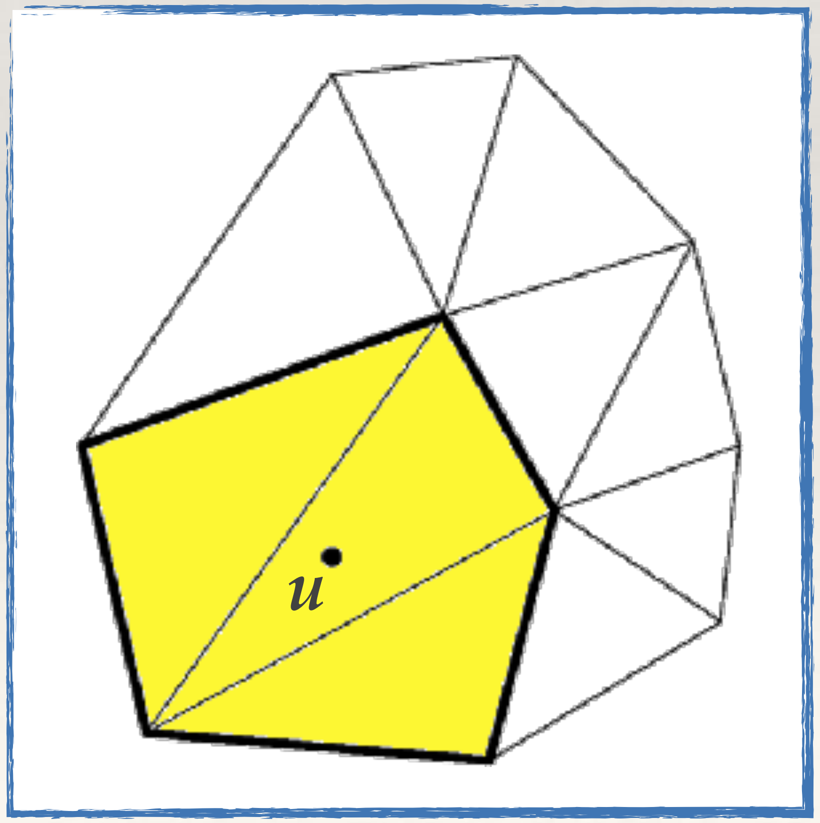
Images from [De Floriani 2003]

Delaunay Triangulation

Watson's Algorithm:

The ***influence region*** $\boldsymbol{R_u}$ of a point u is the region in the plane formed by the union of the triangles in $Del(V_i)$ whose circumcircle contains u in its interior

The ***influence polygon*** $\boldsymbol{P_u}$ of u is the polygon formed by the edges of the triangles of $Del(V_i)$ which bound R_u

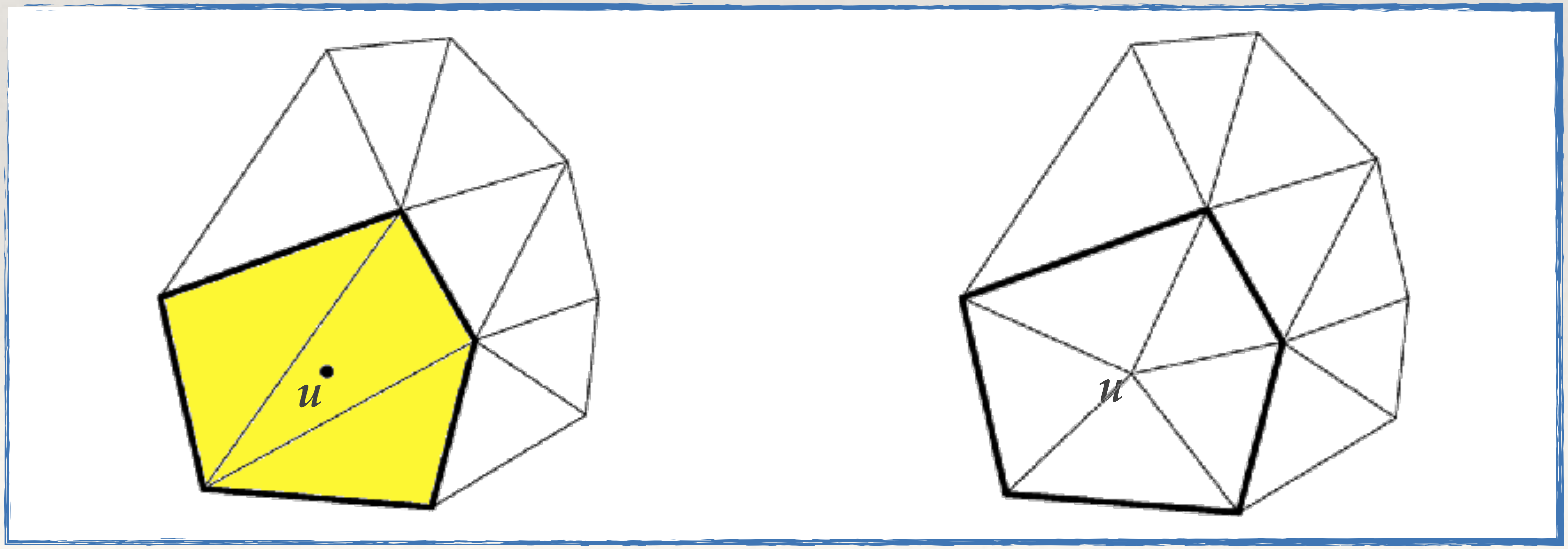


Images from [De Floriani 2003]

Delaunay Triangulation

Watson's Algorithm:

- ***Step 1:*** deletion of the triangles of $Del(V_i)$ forming the ***influence region*** $\boldsymbol{R_u}$
- ***Step 2:*** ***re-triangulation of*** $\boldsymbol{R_u}$ by joining u to the vertices of the influence polygon P_u



Images from [De Floriani 2003]

Delaunay Triangulation

Watson's Algorithm:

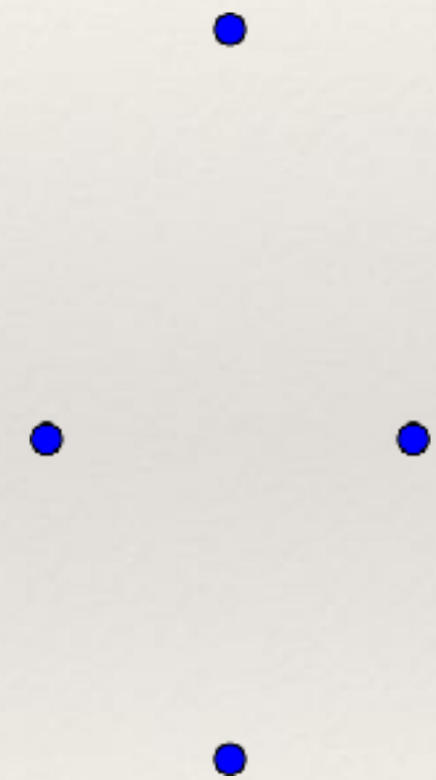
Let $n_i = | V_i |$

- Detection of a triangle σ of $Del(V_i)$ containing the new point u: $O(n_i)$ in the worst case
- Detection of the triangles forming the region of influence through a breadth-first search: $O(| R_u |)$
- Re-triangulation of P_u is in $O(| P_u |)$

- Inserting a point u in a triangulation with n_i vertices: $\mathbf{O(n_i)}$ in the worst case
- Inserting all points of V: $\boldsymbol{O(n^2)}$ in the worst case, where $n = | V |$

Čech Complex

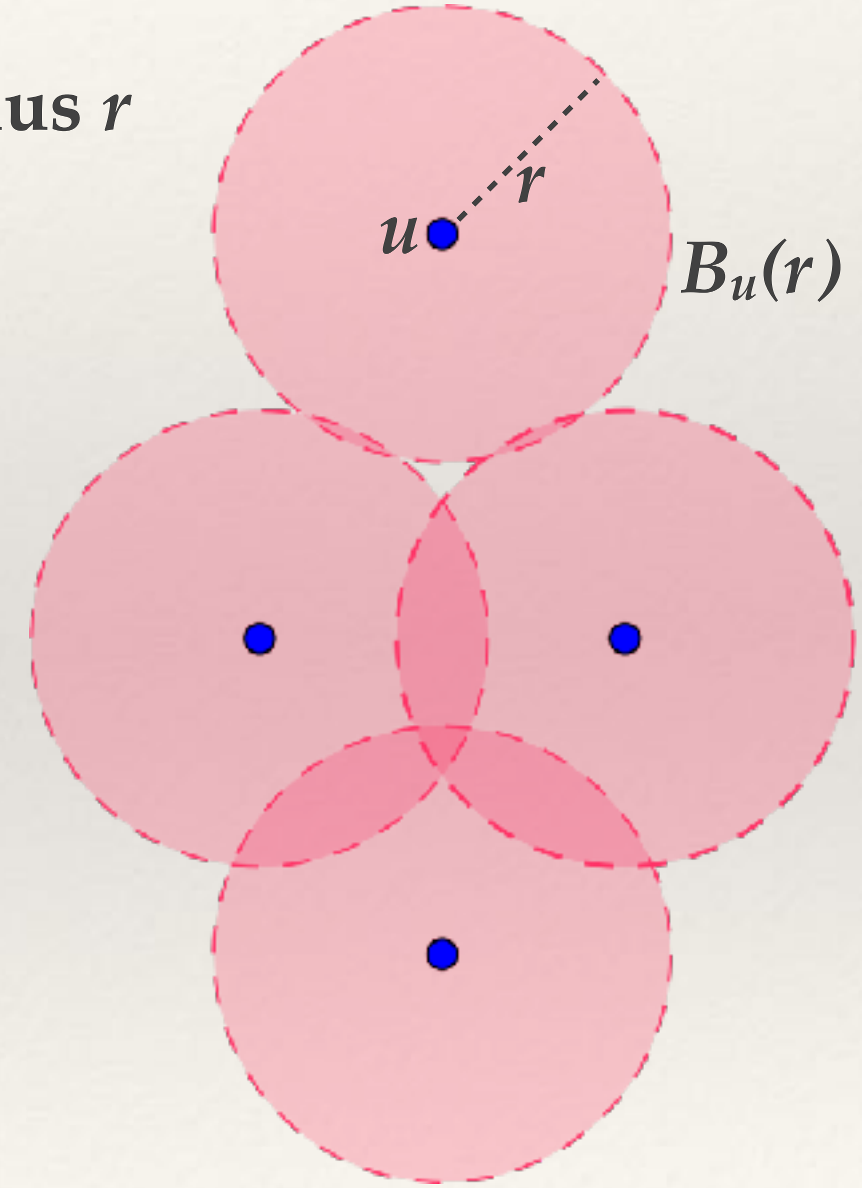
Given a finite set of points V in $\mathbf{R}^d$, let us consider:



Čech Complex

Given a finite set of points V in $\mathbf{R^d}$, let us consider:

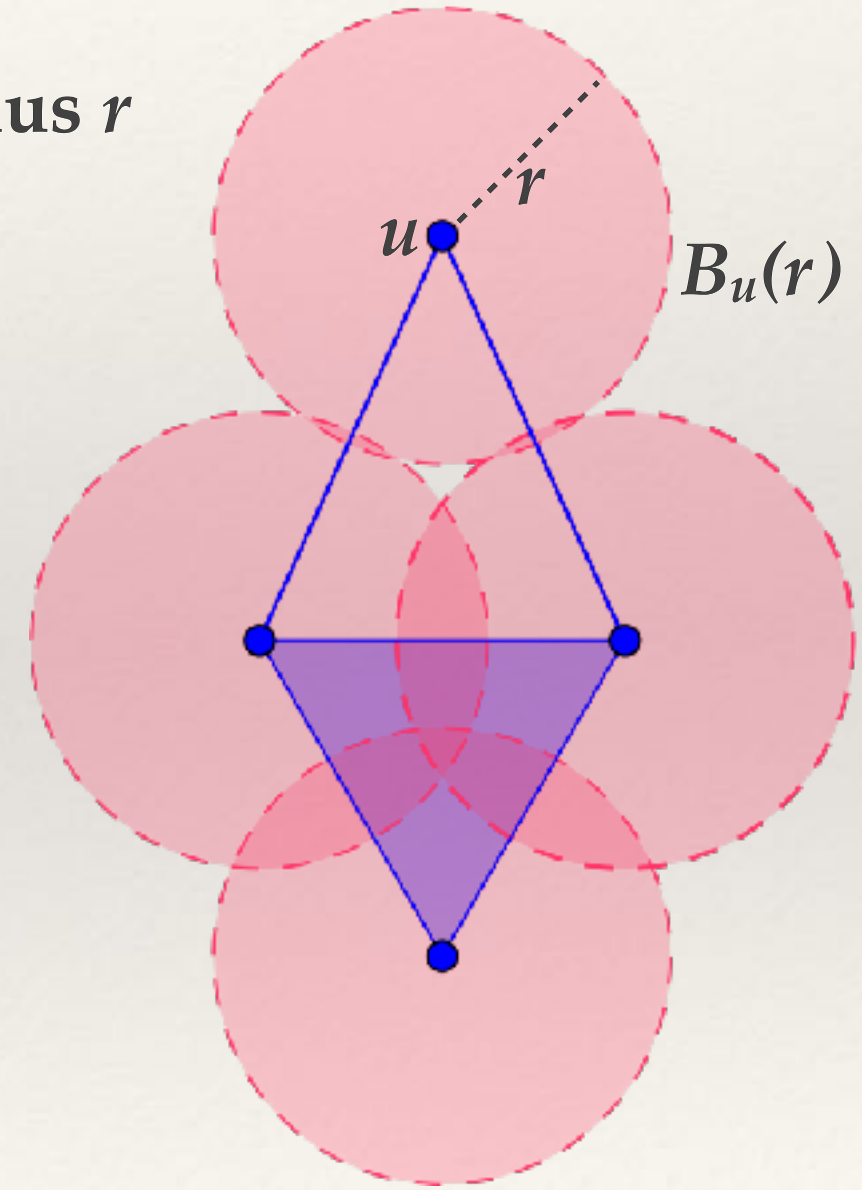
- $B_u(r)$, the **closed ball** with **center** $u \in V$ and **radius** r
- S, the collection of these balls



Čech Complex

Given a finite set of points V in $\mathbf{R}^d$, let us consider:

- $B_u(r)$, the **closed ball** with **center** $u \in V$ and **radius** r
- S, the collection of these balls



The **Čech complex** $\check{C}ech(r)$ of V of radius r is the **nerve of S**

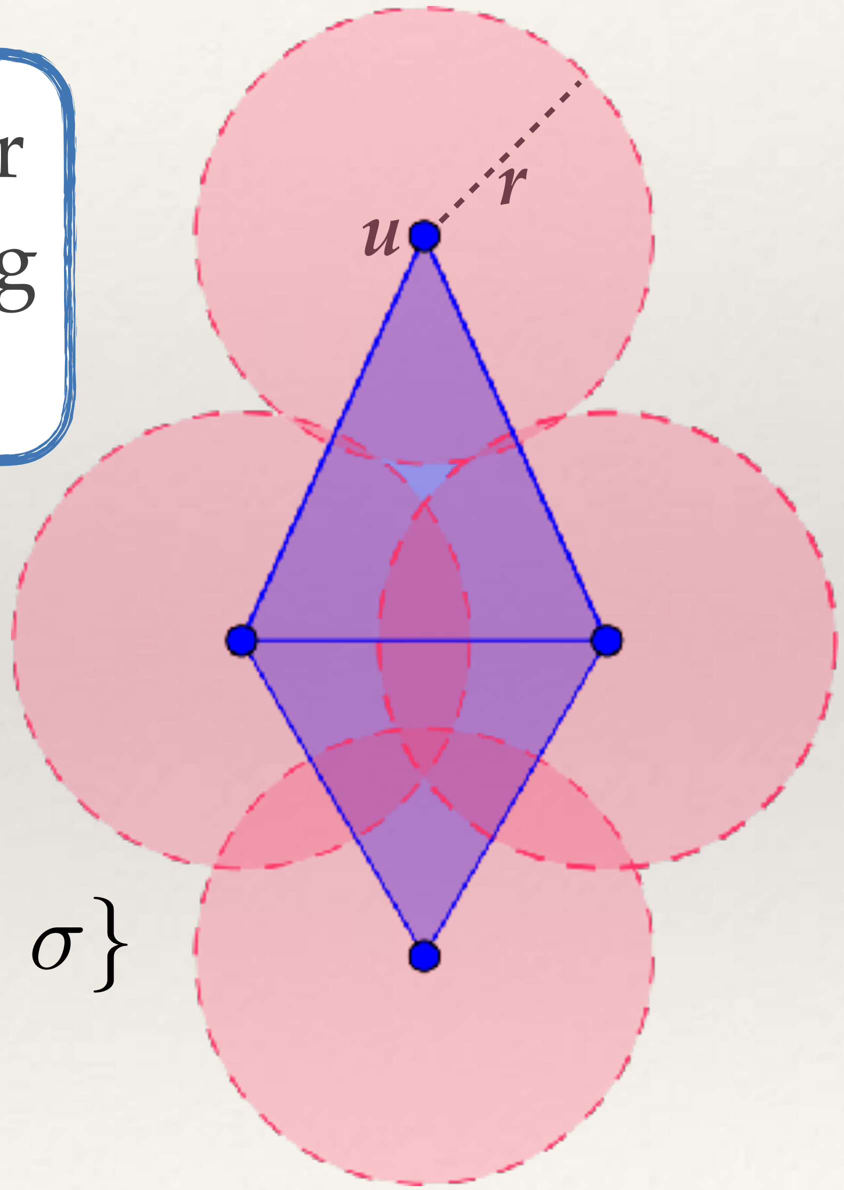
$$\check{C}ech(r) := \{\sigma \subseteq V \mid \bigcap_{u \in \sigma} B_u(r) \neq \emptyset\}$$

In practice, **infeasible construction**

Vietoris-Rips Complex

Given a finite set of points *V* in **R**$^{\mathbf{d}}$,

The **Vietoris-Rips complex** ***VR(r)*** of *V* and r is the ***abstract simplicial complex*** consisting of all ***subsets of diameter at most 2r***



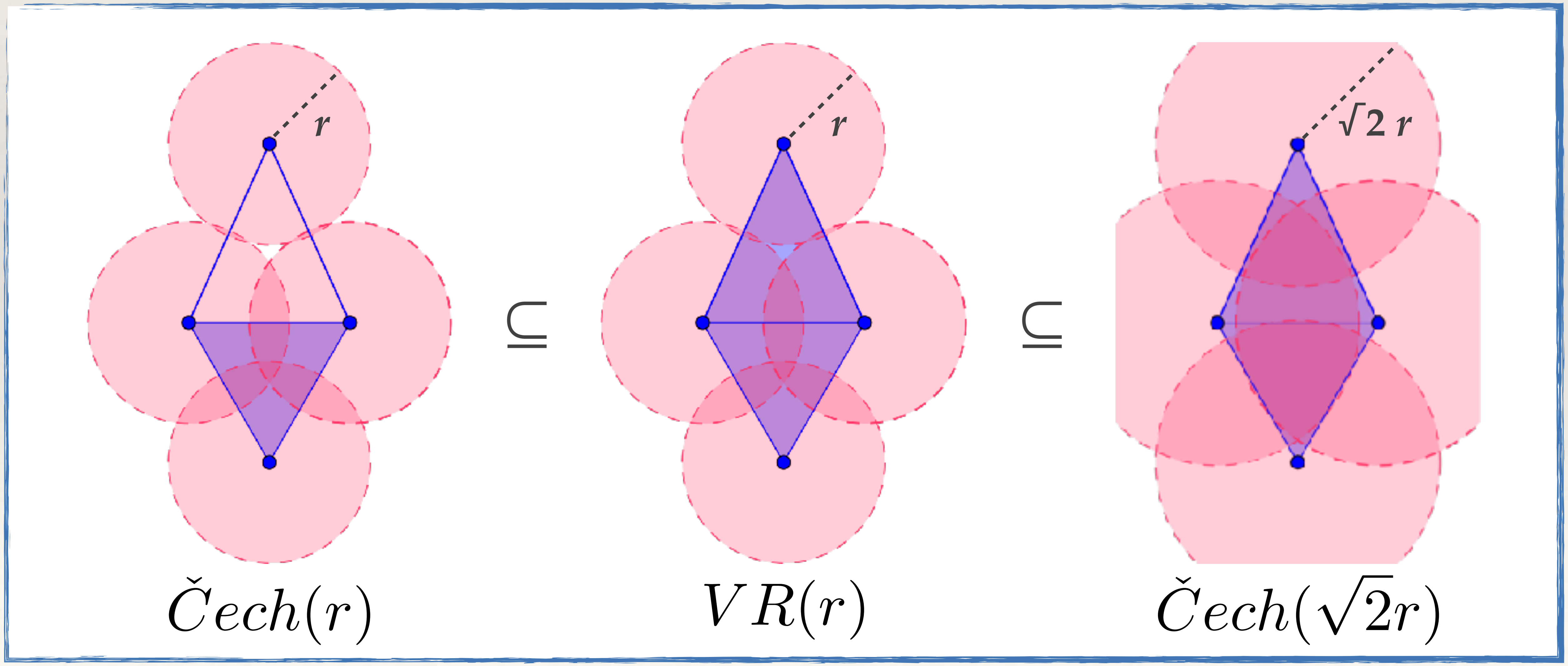
Formally,

$$VR(r) := \{\sigma \subseteq V \mid d(u,v) \leq 2r, \forall\, u, v \in \sigma\}$$

Vietoris-Rips Complex

Properties:

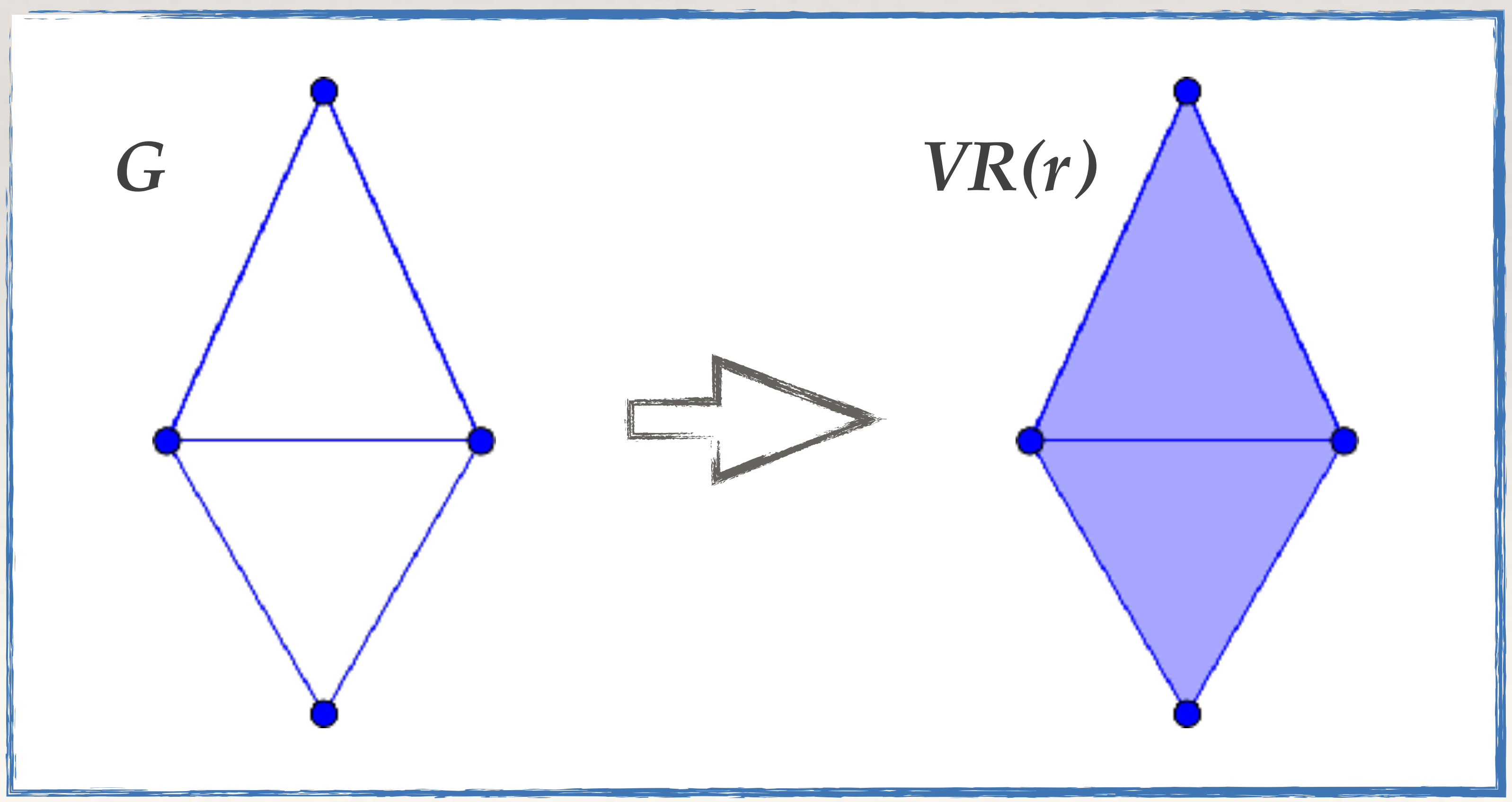
- $\check{C}ech(r) \subseteq VR(r) \subseteq \check{C}ech(\sqrt{2}r)$



Vietoris-Rips Complex

Properties:

- $\check{C}ech(r) \subseteq VR(r) \subseteq \check{C}ech(\sqrt{2}r)$
- ***VR(r)*** is completely determined by its 1-skeleton
  - i.e., the graph *G* of its vertices and its edges



Vietoris-Rips Complex

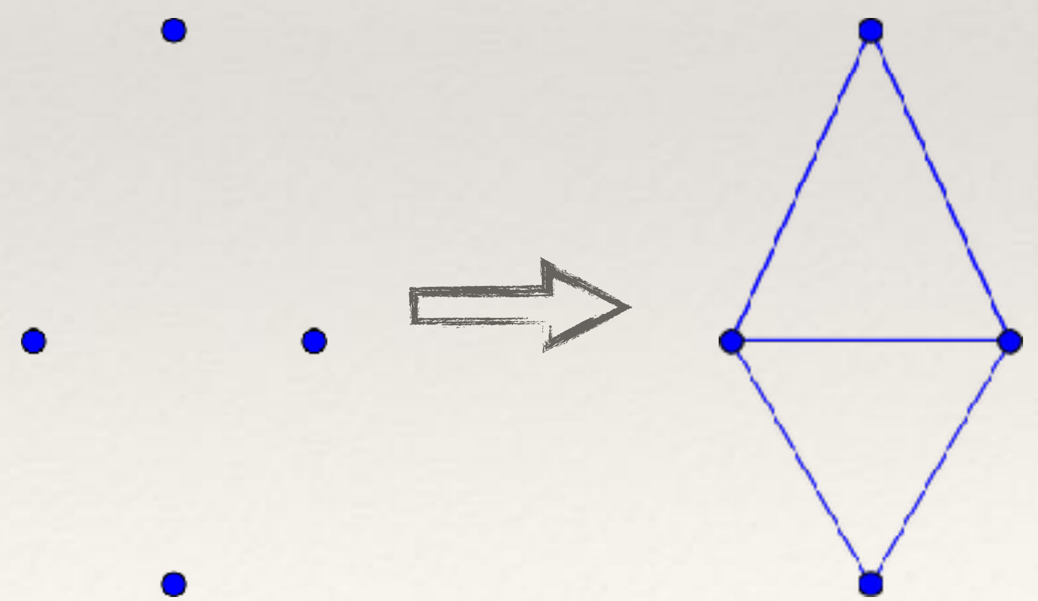
Computation: [Zomorodian 2010]

<u>Input</u>: finite set of points V in R^d and a real positive number r

<u>Output</u>: the Vietoris-Rips complex $VR(r)$

Two-step Algorithm:

- **1-Skeleton Computation**:
  - *Exact* ($O(|V|^2)$ time complexity)
  - *Approximate*
  - *Randomized*
  - *Landmarking*
- **Vietoris-Rips Expansion**:
  - *Inductive*
  - *Incremental*
  - *Maximal*



Vietoris-Rips Complex

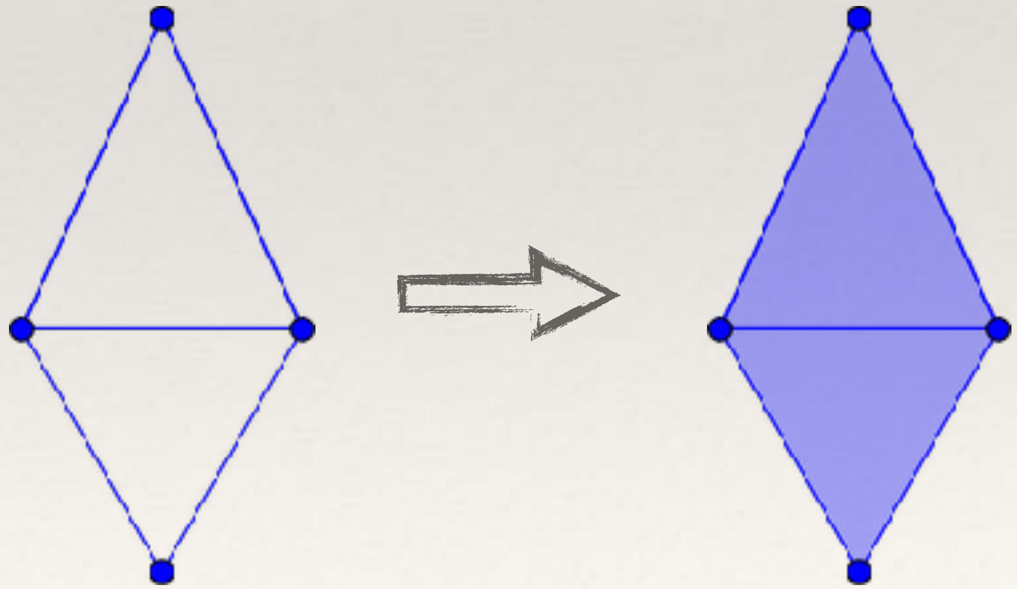
Computation: [Zomorodian 2010]

Input: finite set of points V in R^d and a real positive number r

Output: the Vietoris-Rips complex $VR(r)$

Two-step Algorithm:

- **1-Skeleton Computation**:
  - *Exact* ($O(|V|^2)$ time complexity)
  - *Approximate*
  - *Randomized*
  - *Landmarking*
- **Vietoris-Rips Expansion**:
  - *Inductive*
  - *Incremental*
  - *Maximal*



Vietoris-Rips Complex

Inductive VR expansion:

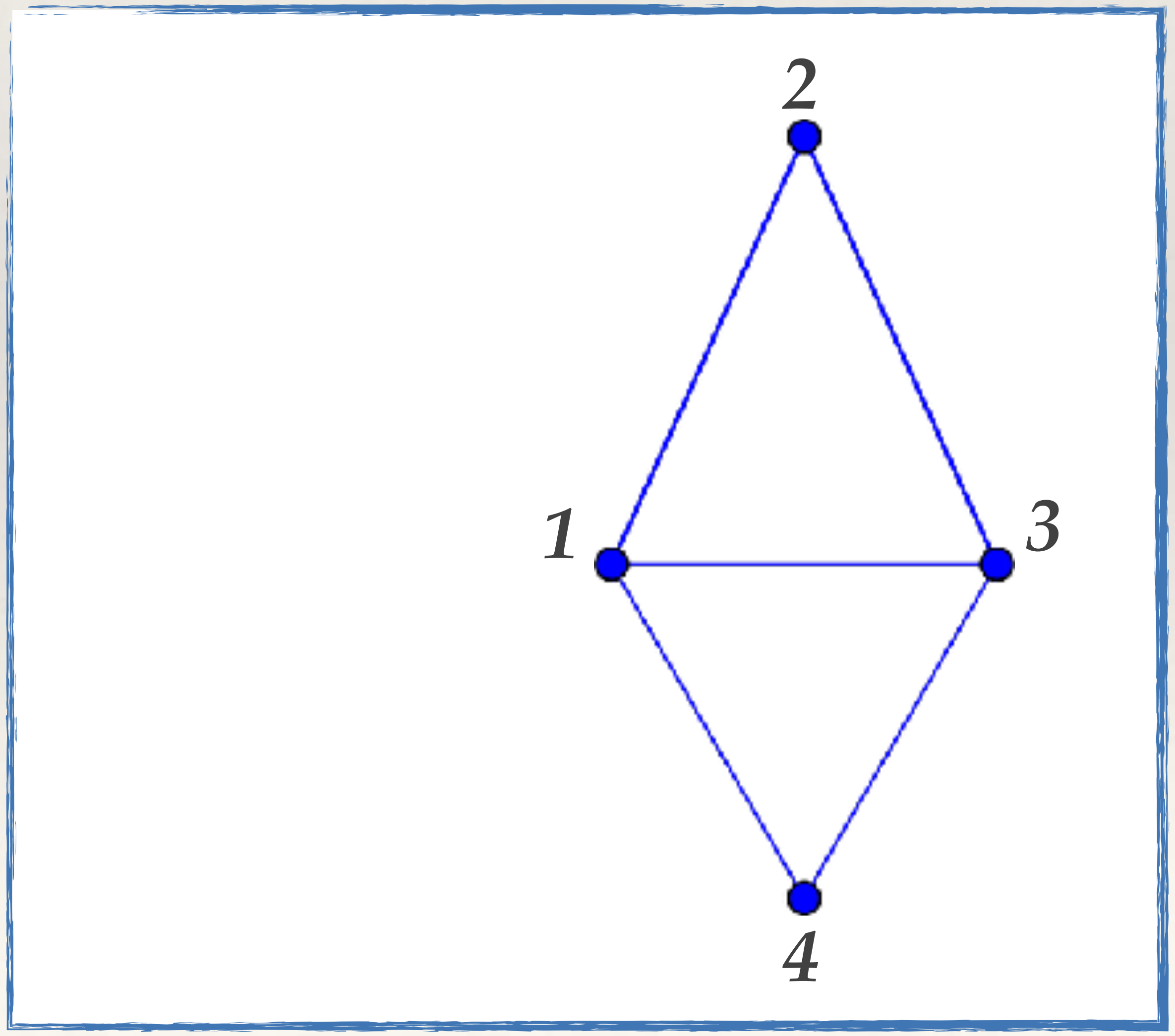
Input: the 1-skeleton $G=(V, E)$ of $VR(r)$

Output: the k-skeleton Σ of the Vietoris-Rips complex $VR(r)$

INDUCTIVE-VR(*G*, *k*)
$\Sigma = V \cup E$
for $i=1$ **to** k
foreach i-simplex $\sigma \in \Sigma$

$N = \bigcap_{u \in \sigma}$ LOWER-NBRS(G, u)
foreach $v \in N$
$\Sigma = \Sigma \cup \{ \sigma \cup \{v\}\}$
return Σ

LOWER-NBRS(*G*, *u*)
return $\{v \in V \mid u > v\ , (u, v) \in E\}$



Vietoris-Rips Complex

Inductive VR expansion:

Input: the 1-skeleton G=(V, E) of $VR(r)$

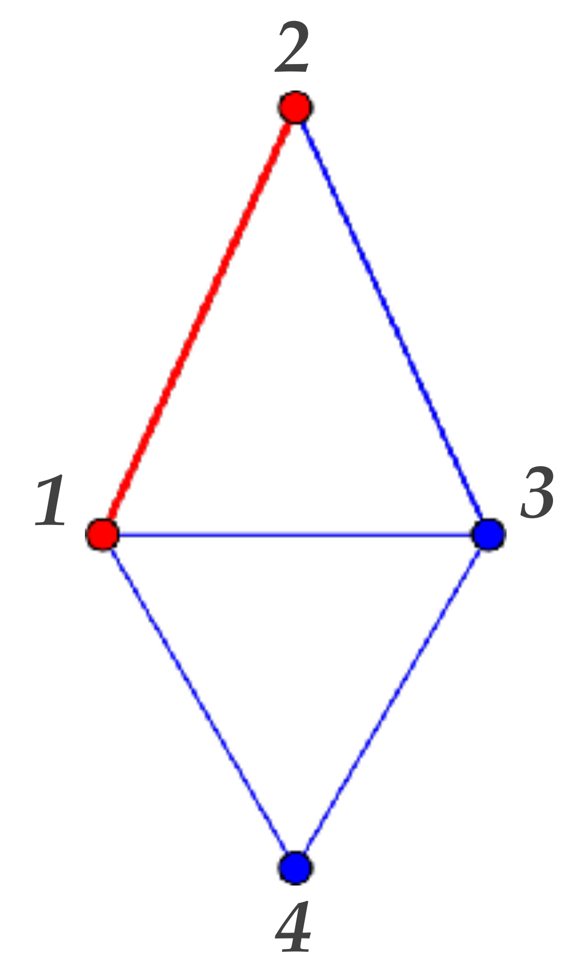
Output: the k-skeleton Σ of the Vietoris-Rips complex $VR(r)$

INDUCTIVE-VR(G, k)
$\Sigma = V \cup E$
for i=1 **to** k
foreach i-simplex $\sigma \in \Sigma$
$N = \bigcap_{u \in \sigma}$ LOWER-NBRS(G, u)
foreach $v \in N$
$\Sigma = \Sigma \cup \{ \sigma \cup \{v\}\}$
return Σ

LOWER-NBRS(G, u)
return $\{v \in V \mid u > v \text{ , } (u, v) \in E\}$

$\sigma = (1, 2)$

$N = \{ \}$



Vietoris-Rips Complex

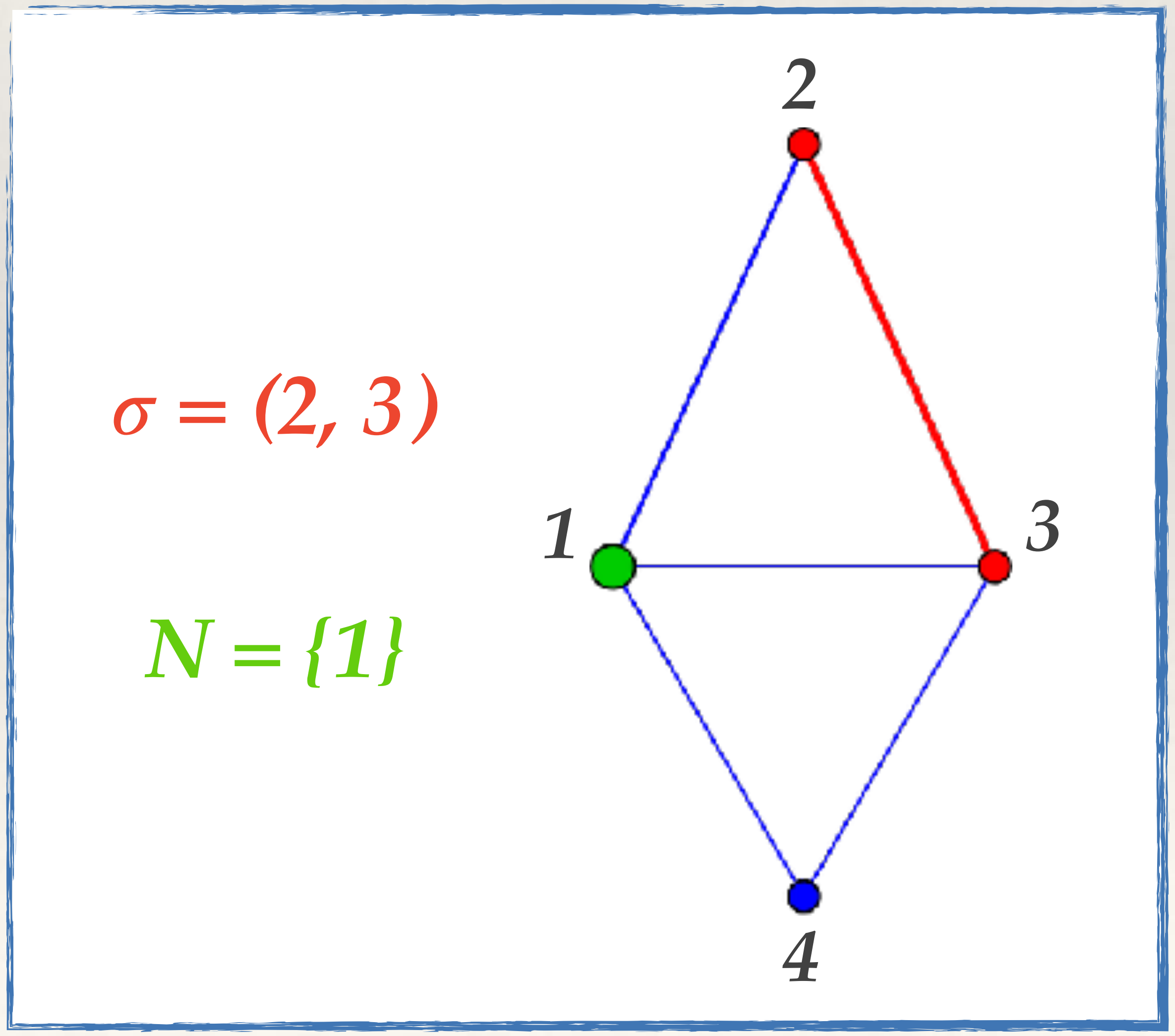
Inductive VR expansion:

<u>Input</u>: the 1-skeleton G=(V, E) of $VR(r)$

<u>Output</u>: the k-skeleton Σ of the Vietoris-Rips complex $VR(r)$

INDUCTIVE-VR(G, k)
  $\Sigma = V \cup E$
  for i=1 **to** k
    foreach i-simplex $\sigma \in \Sigma$
      $N = \bigcap_{u \in \sigma}$ LOWER-NBRS(G, u)
      foreach $v \in N$
        $\Sigma = \Sigma \cup \{ \sigma \cup \{v\}\}$
    return Σ

LOWER-NBRS(G, u)
    return $\{v \in V \mid u > v \,, (u, v) \in E\}$



Vietoris-Rips Complex

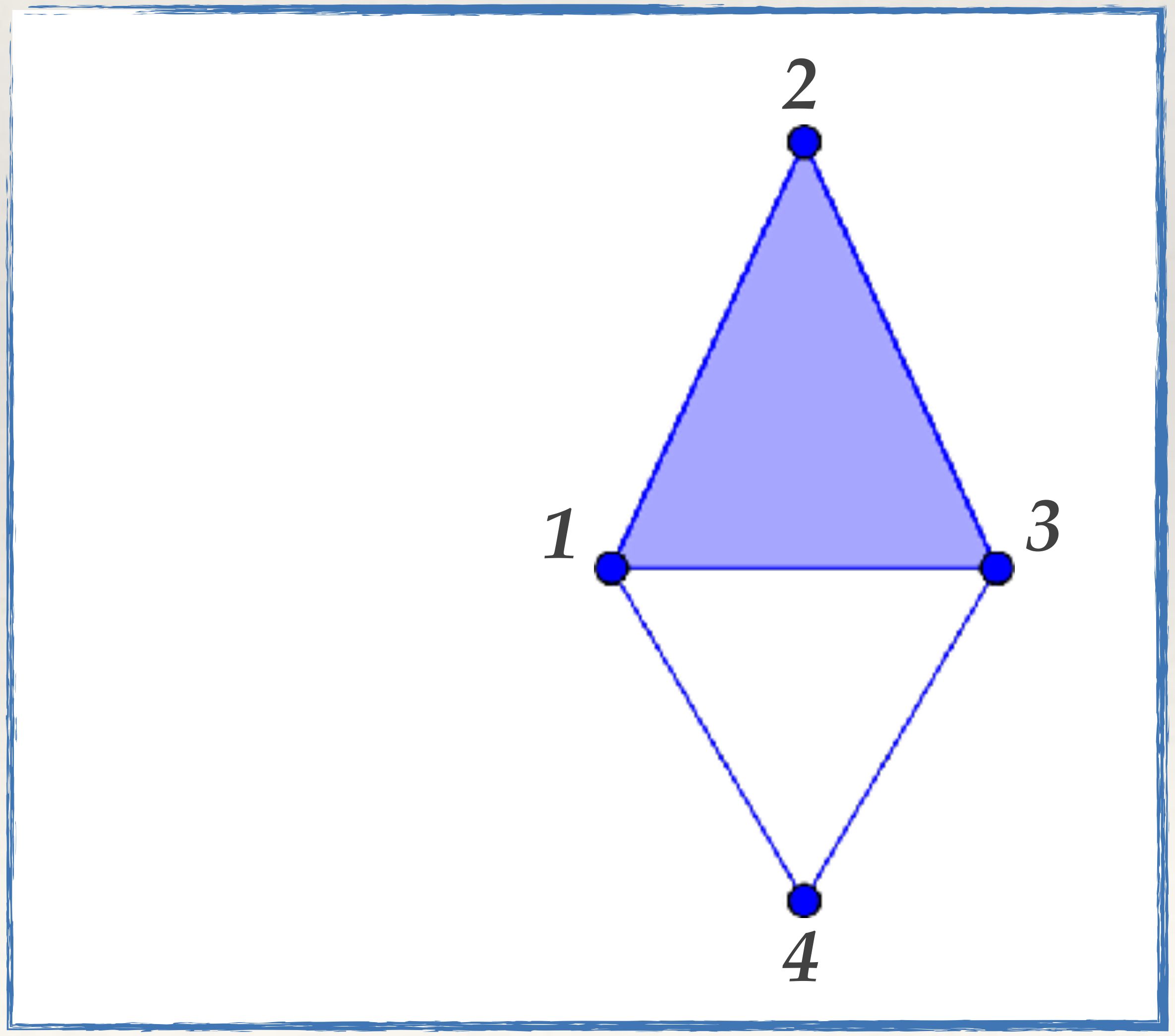
Inductive VR expansion:

Input: the 1-skeleton $G=(V, E)$ of $VR(r)$

Output: the k-skeleton Σ of the Vietoris-Rips complex $VR(r)$

INDUCTIVE-VR(*G*, *k*)
$\Sigma = V \cup E$
for $i=1$ **to** k
foreach i-simplex $\sigma \in \Sigma$
$N = \bigcap_{u \in \sigma}$ LOWER-NBRS(G, u)
foreach $v \in N$
$\Sigma = \Sigma \cup \{ \sigma \cup \{v\}\}$
return Σ

LOWER-NBRS(*G*, *u*)
return $\{v \in V \mid u > v , (u, v) \in E\}$



Vietoris-Rips Complex

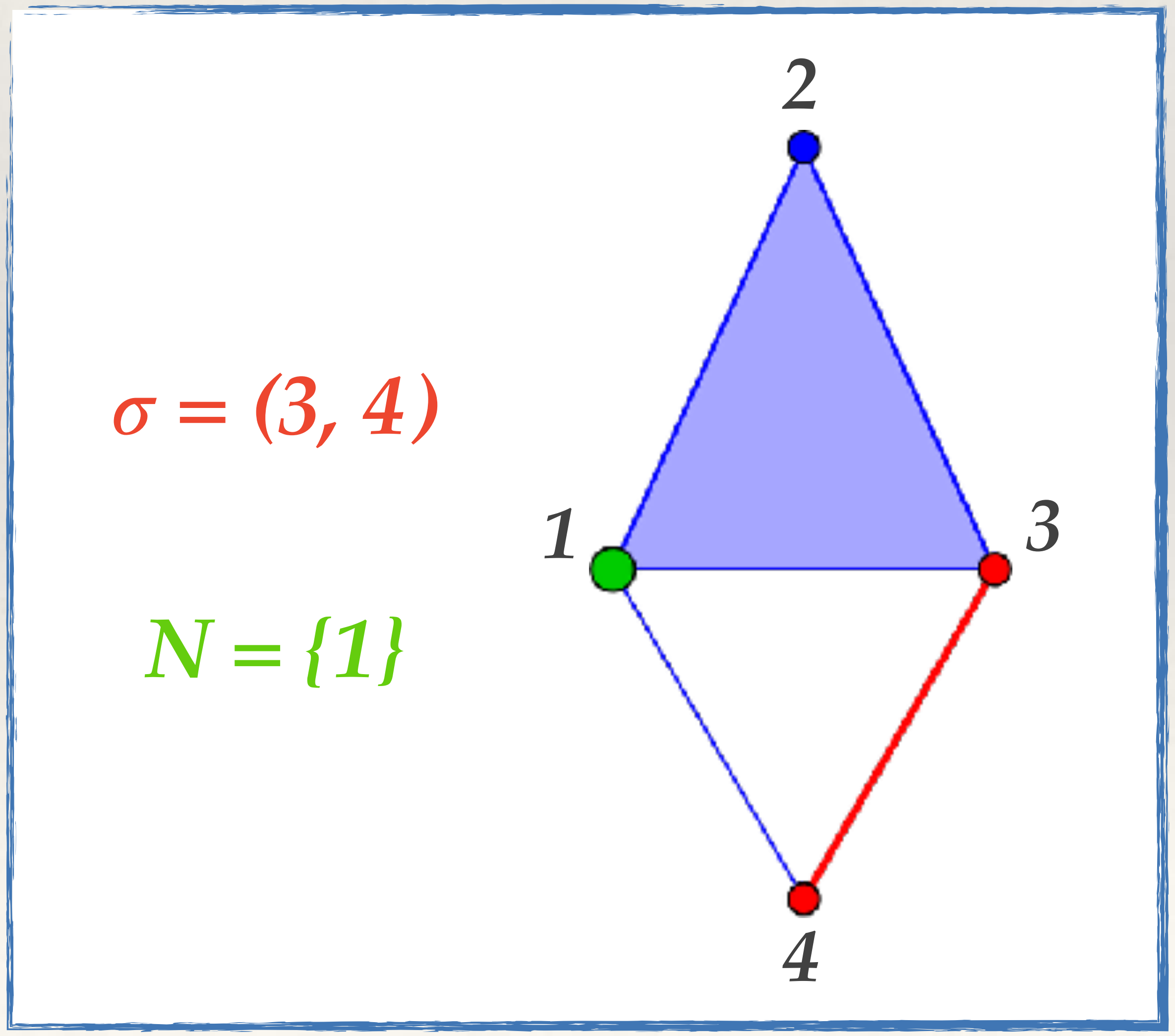
Inductive VR expansion:

Input: the 1-skeleton $G=(V, E)$ of $VR(r)$

Output: the k-skeleton Σ of the Vietoris-Rips complex $VR(r)$

INDUCTIVE-VR(*G*, *k*)
$\Sigma = V \cup E$
for $i=1$ **to** k
foreach i-simplex $\sigma \in \Sigma$
$N = \bigcap_{u \in \sigma}$ LOWER-NBRS(G, u)
foreach $v \in N$
$\Sigma = \Sigma \cup \{ \sigma \cup \{v\}\}$
return Σ

LOWER-NBRS(*G*, *u*)
return $\{v \in V \mid u > v\ , (u, v) \in E\}$



Vietoris-Rips Complex

Inductive VR expansion:

<u>Input</u>: the 1-skeleton $G=(V, E)$ of $VR(r)$

<u>Output</u>: the k-skeleton Σ of the Vietoris-Rips complex $VR(r)$

INDUCTIVE-VR(G, k)

$\Sigma = V \cup E$

for $i=1$ **to** k

foreach i-simplex $\sigma \in \Sigma$

$N = \bigcap_{u \in \sigma}$ LOWER-NBRS(G, u)

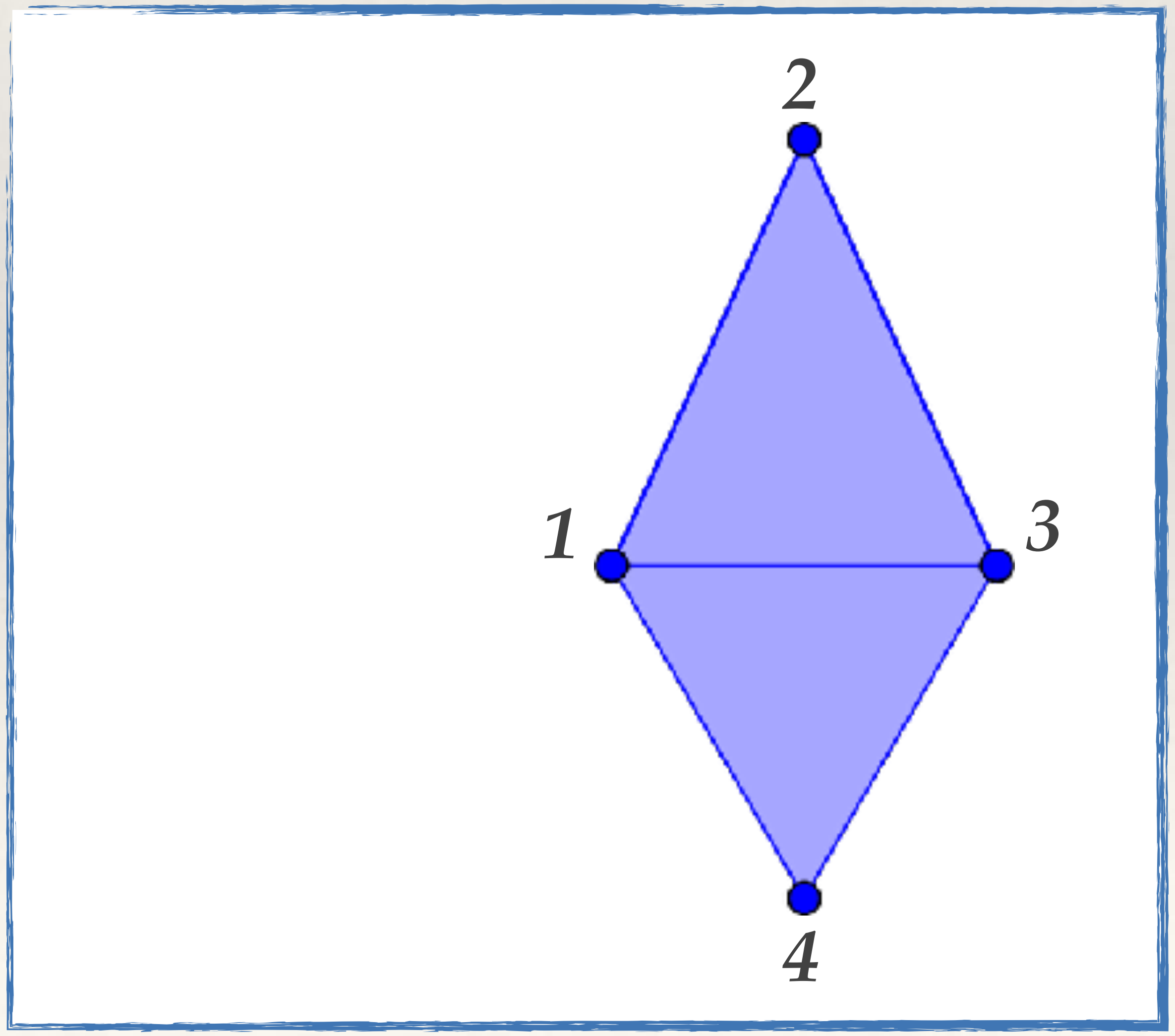
foreach $v \in N$

$\Sigma = \Sigma \cup \{ \sigma \cup \{v\}\}$

return Σ

LOWER-NBRS(G, u)

return $\{v \in V \mid u > v, (u, v) \in E\}$



From a Point Cloud To a Complex

		
Delaunay triangulation	Bounded Dimension	Trivial Homology
Čech complex/VR complex	"Real" Homology	High Dimension Large Size

Alpha-shape

Given a finite set of points V in general position of $\mathbf{R}^d$, let us consider:

- $A_u(r) := B_u(r) \cap R_V(u)$
  - **intersection** of the **closed ball of radius r** centered in u and the **Voronoi region** of u
- S, the collection of these convex sets

The **Alpha-shape** $Alpha(r)$ of V of radius r is the **nerve of S**

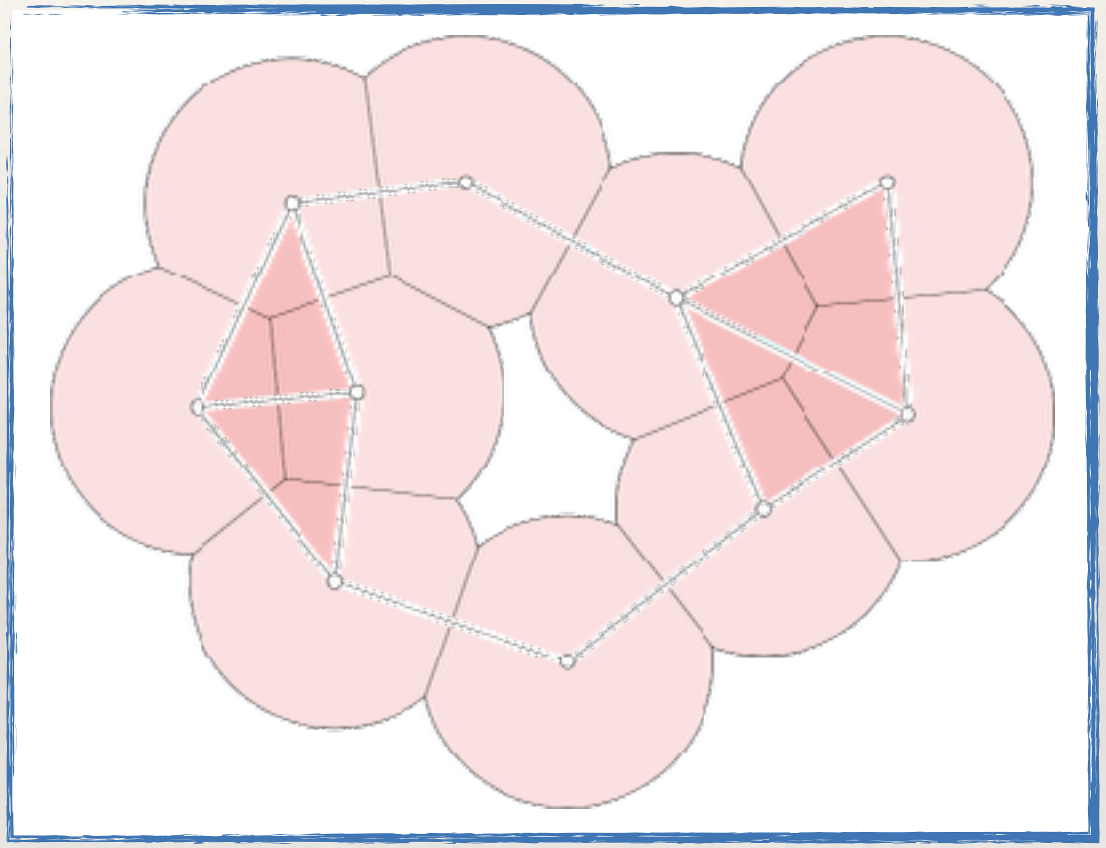


Image from [Edelsbrunner, Harer 2010]

Formally,

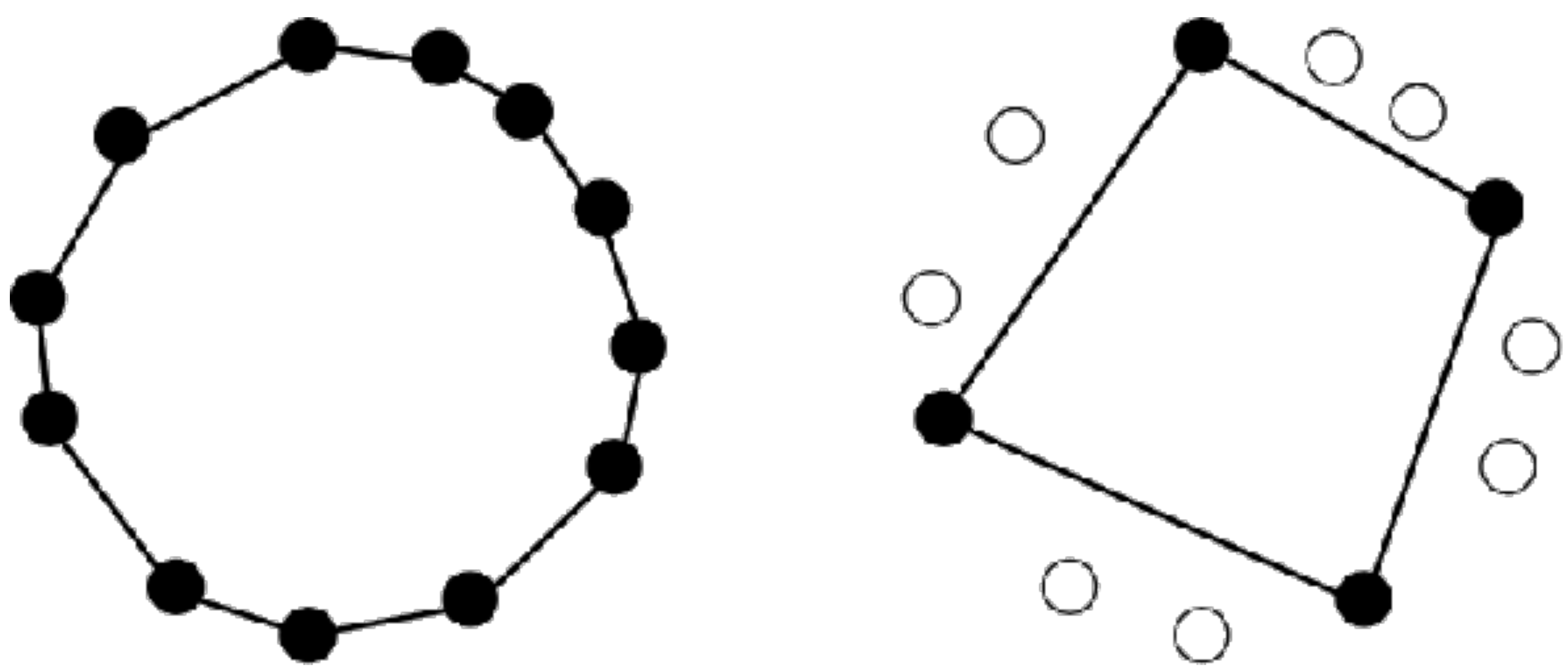
$$Alpha(r) := \{\sigma \subseteq V \mid \bigcap_{u \in \sigma} A_u(r) \neq \emptyset\}$$

$$A_u(r) \subseteq B_u(r) \implies Alpha(r) \subseteq \check{C}ech(r)$$

Witness Complex

Motivation:

Retrieving the **topological information** does not require to consider **all the input points**



- **Landmarks**: selected points
- **Witnesses**: remaining points

Images from [de Silva, Carlsson 2004]

Witness Complex

For each witness w,

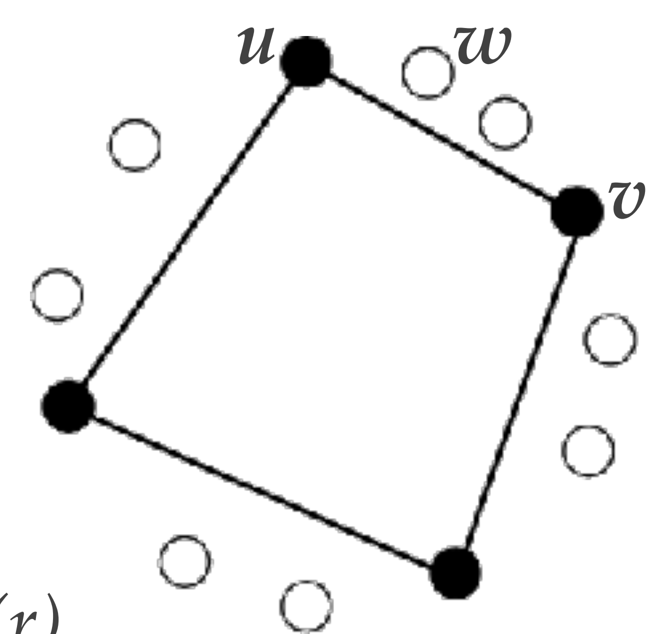
m_w : = the distance of w from the *2nd* **closest landmark**

The **witness complex** ***W(r)*** of radius r is defined by:

- u is in $W(r)$ if u is a landmark
- ***(u,v)*** **is in *W(r)* if there exists a witness *w* such that**

$$max\{d(u,w), d(v,w)\} \leq m_w + r$$

- the i-simplex σ is in $W(r)$ if all its edges belong to $W(r)$



W₀(r) is defined by setting $\mathbf{m_w = 0}$ for any witness w

$$W_0(r) \subseteq VR(r) \subseteq W_0(2r)$$

Images from [de Silva, Carlsson 2004]

Outline

Describing a Shape through Persistence Pairs

From a Point Cloud to a Filtered Simplicial Complex

Thank you

Ulderico Fugacci

***TU Kaiserslautern**, Dept. of Computer Science*