Persistent homology

Computational topology

Julien Tierny
jtierny@sci.utah.edu

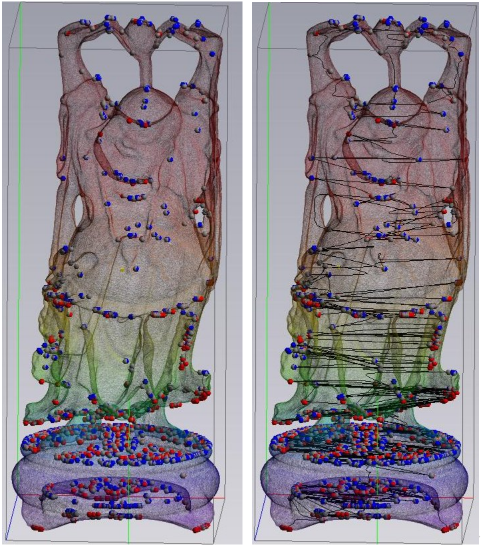
February 13, 2009

Applicative motivations

- Computational topology:
  - Concise topology abstractions for:
    - Computer graphics;
    - **Visualization;**
    - **Data analysis**, etc.

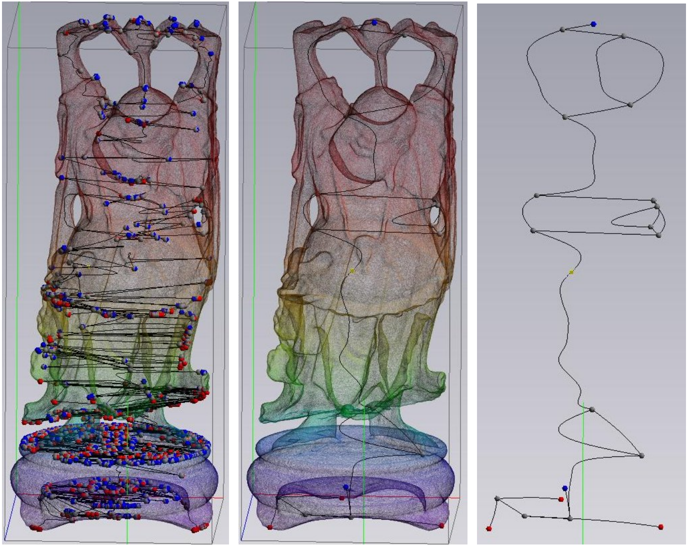
But...

- Honestly, can you **see** anything?



Now,

- Is it any better?

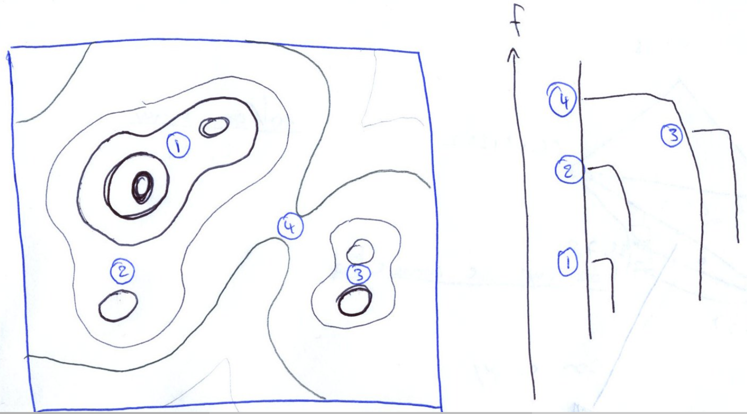


Persistence

- Need to:
  - Cope with the effect of geometrical noise on topology abstractions;
- Yes... but no! How do you define noise then?
  - Let's make it up to the application needs!
- Persistence key ideas:
  - Provide an abstract framework to:
    - Measure scales on topological features;
    - Order topological features in term of importance/noise.
  - How *long* is a topological feature persistent?
    - As long as it *refuses to die*...

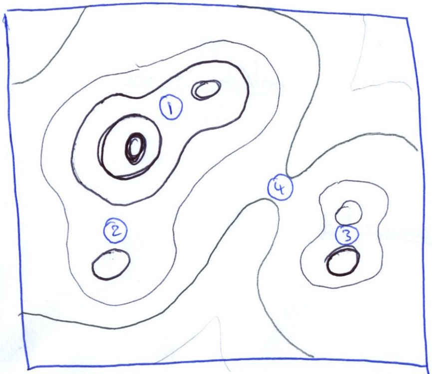
Basic intuition (1/3)

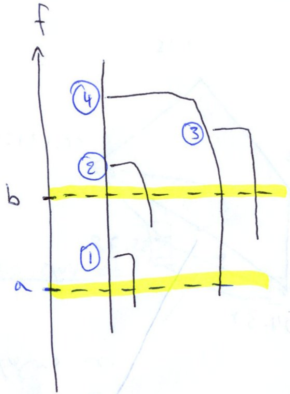
- $f : \mathbb{M} \rightarrow \mathbb{R}$;
- In $\mathcal{R}(f)$, apply the *elder's rule*:
  - Think of arc's lower extremity's value as *birthdate*;
  - At a juncture, the older arc continues and the younger ends.
- Now pick two image values a and b ($a \leq b$).



Basic intuition (1/3)

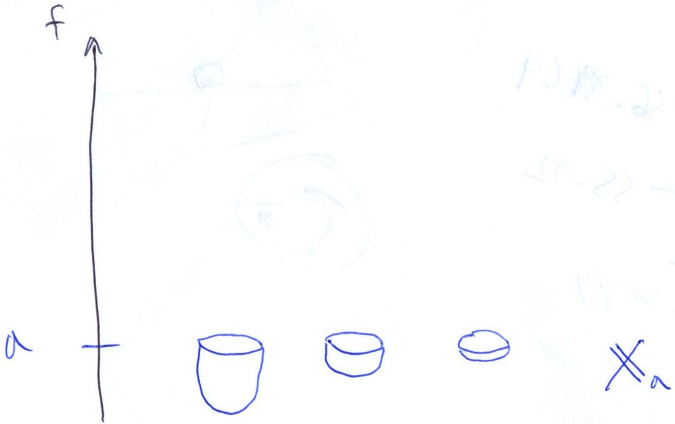
- $f : \mathbb{M} \to \mathbb{R}$;
- In $\mathcal{R}(f)$, apply the *elder's rule*:
  - Think of arc's lower extremity's value as *birthdate*;
  - At a juncture, the older arc continues and the younger ends.
- Now pick two image values a and b ($a \leq b$).





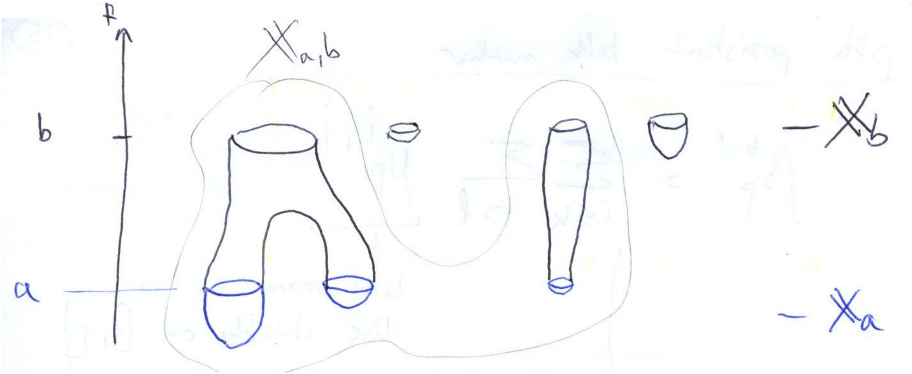
Basic intuition (2/3)

- Consider the sub-level sets $\mathbb{X}_a$ and $\mathbb{X}_b$ of a and b;
- Let $\mathbb{X}_{(a,b)}$ be the union of the connected components of $\mathbb{X}_b$ that have a non-empty intersection with $\mathbb{X}_a$:
- Let $\beta_0(a, b) = \#CC(\mathbb{X}_{(a,b)})$ (here $\beta_0(a, b) = 2$).



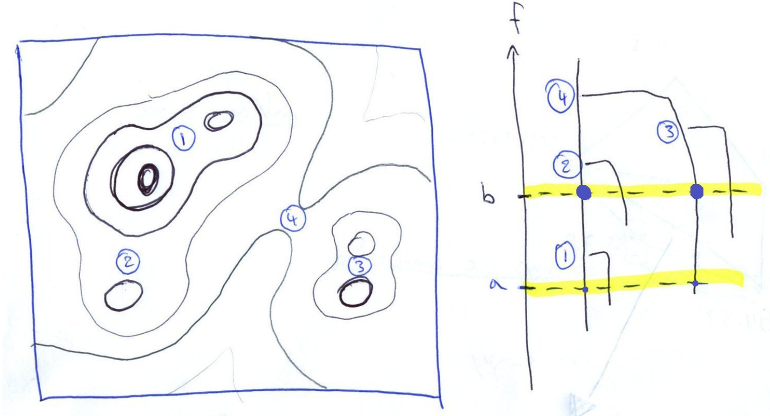
Basic intuition (2/3)

- Consider the sub-level sets $\mathbb{X}_a$ and $\mathbb{X}_b$ of a and b;
- Let $\mathbb{X}_{(a,b)}$ be the union of the connected components of $\mathbb{X}_b$ that have a non-empty intersection with $\mathbb{X}_a$:
- Let $\beta_0(a,b) = \#CC(\mathbb{X}_{(a,b)})$ (here $\beta_0(a,b) = 2$).



Basic intuition (3/3)

- If f is Morse, we can read $\beta_0(a, b)$ on the Reeb graph $\mathcal{R}(f)$:
  - $\beta_0(a, b)$ is the number of arcs that strictly span $[a, b]$.



Mission accomplished!

- By the way, what did we do exactly?
- We've just identified:
  - regarding to f,
  - the connected components with biggest *"life duration"* on $[a, b]$:

Mission accomplished!

- By the way, what did we do exactly?
- We've just identified:
  - regarding to f,
    - **User defined measurement system!**
  - the connected components with biggest *"life duration"* on $[a, b]$:
    - $[a, b]$: **User defined scale/zoom!**
    - $\beta_0(a, b)$: **Topological features**.
- **Classification of topological features wrt the importance suggested by f:**
  - Make the *zoom* $[a, b]$ increase to sort the the arcs of $\mathcal{R}(f)$ by increasing *topological importance*.
  - You only have to get rid progressively of the least *topological important* arcs to filter $\mathcal{R}(f)$...

... Mission really accomplished?

- So far, we introduced:
  - A general framework for:
    - Measuring importance of connected components;
    - Focusing on user defined scales;
    - Classifying connected components by importance.
- How can we extend it to other topological features?
- By the way, what are these other topological features?

... Mission really accomplished?

- So far, we introduced:
  - A general framework for:
    - Measuring importance of connected components;
    - Focusing on user defined scales;
    - Classifying connected components by importance.
- How can we extend it to other topological features?
- By the way, what are these other topological features?
  - Number of connected components: ...β_0;
  -
  -

... Mission really accomplished?

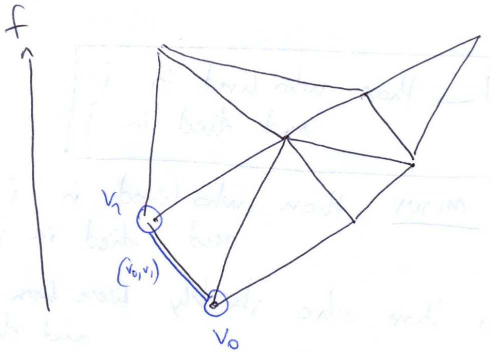
- So far, we introduced:
  - A general framework for:
    - Measuring importance of connected components;
    - Focusing on user defined scales;
    - Classifying connected components by importance.
- How can we extend it to other topological features?
- By the way, what are these other topological features?
  - Number of connected components: ...β_0;
  - Let's generalize to the other Betti numbers! **:)**
  -

... Mission really accomplished?

- So far, we introduced:
  - A general framework for:
    - Measuring importance of connected components;
    - Focusing on user defined scales;
    - Classifying connected components by importance.
- How can we extend it to other topological features?
- By the way, what are these other topological features?
  - Number of connected components: ...β_0;
  - Let's generalize to the other Betti numbers! **:)**
  - Notion of **persistent homology groups.**

Filtration

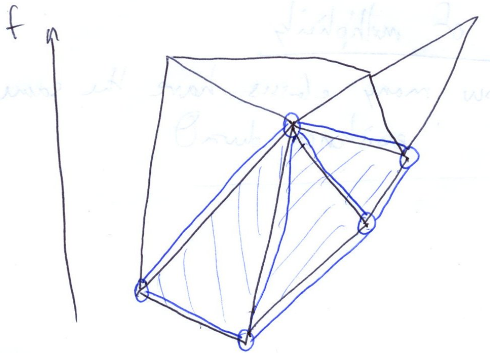
- $f : K \rightarrow \mathbb{R}$, such that f is injective and *monotonic*:
  - $f(\sigma) \leq f(\tau) \quad \forall (\sigma, \tau) \in K \mid \sigma \leq \tau$.
  - Example:
    - $f : VertK \rightarrow \mathbb{R}$;
    - $f(\tau) = max_{\sigma \leq \tau}(f(\sigma)) + \epsilon, \quad \epsilon \rightarrow 0$.
- **Filtration:** sequence of the sub-complexes K_i of $f^{-1}(-\infty, a_i]$.



- $K_0 = \emptyset$;
- $K_1 = \{v_0\}$;
- $K_2 = \{v_0, v_1\}$;
- $K_3 = \{v_0, v_1, (v_0, v_1)\}$.

Filtration

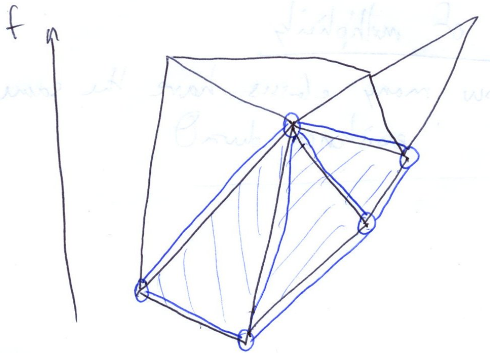
- $f : K \to \mathbb{R}$, such that f is injective and *monotonic*:
  - $f(\sigma) \leq f(\tau) \quad \forall(\sigma, \tau) \in K \mid \sigma \leq \tau$.
  - Example:
    - $f : VertK \to \mathbb{R}$;
    - $f(\tau) = max_{\sigma \leq \tau}(f(\sigma)) + \epsilon, \quad \epsilon \to 0$.
- **Filtration:** sequence of the sub-complexes K_i of $f^{-1}(-\infty, a_i]$.



- This is K_i.
- What is i equal to?:
  -
  -
  -
  -

Filtration

- $f : K \to \mathbb{R}$, such that f is injective and *monotonic*:
  - $f(\sigma) \leq f(\tau) \quad \forall(\sigma, \tau) \in K \mid \sigma \leq \tau$.
  - Example:
    - $f : VertK \to \mathbb{R}$;
    - $f(\tau) = max_{\sigma \leq \tau}(f(\sigma)) + \epsilon, \quad \epsilon \to 0$.
- **Filtration:** sequence of the sub-complexes K_i of $f^{-1}(-\infty, a_i]$.



- This is K_i.
- What is i equal to?:
  - Progressive f span,
  - one simplex / it;
  - We have:
    - 5 vertices,
    - 7 edges,
    - 3 triangles.
  - This is K_{15}.

The filtration as a measurement sequence

- Filtration:
  - $\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$
  - This defines a natural **measurement sequence** (with regard to f);
- Given some user-defined scale $[a, b]$ on f, we want to:
  - See how the topological features (Betti numbers) evolve.
- Simple!
  - Let's look at the homology groups at each step of the sequence!
    - Finest scale.
  - Look at this evolution on arbitrary $[a_i, a_j]$ such that $i \leq j$:
    - Here is the scale :)

Homomorphisms induced by the filtration

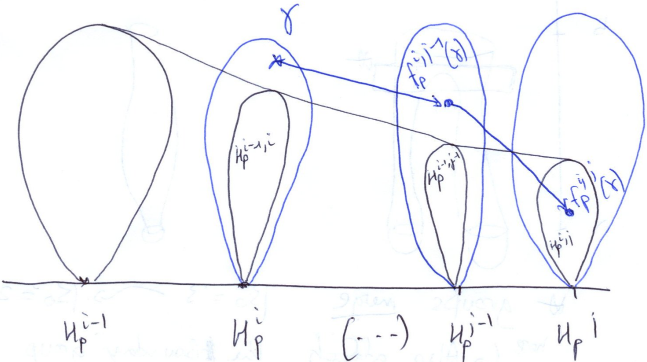
- The filtration induces a sequence of inclusion maps:
  - $|K_0| \rightarrow |K_1| \rightarrow \cdots \rightarrow |K|$;
- ... and then a sequence of homomorphisms on the homology groups:
  - $0 = H_p(K_0) \rightarrow H_p(K_1) \rightarrow \cdots \rightarrow H_p(K_n) = H_p(K)$
- $f_p^{i,j} : H_p(K_i) \rightarrow H_p(K_j)$:
  - Maps some classes from $H_p(K_i)$ to **some** of $H_p(K_j)$;
  - **some**: those who still *live* in $H_p(K_j)$.
- but... hold on a second...
  - This is the exact idea of *incremental Betti numbers computation* [DE93]!

Persistent homology groups

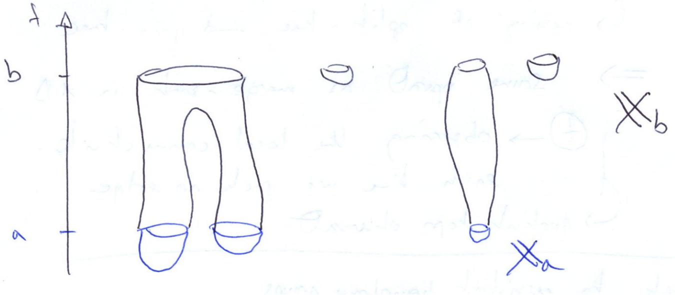
Definition (p^{th} persistent homology groups)

The p^{th} *persistent homology groups* are the images of the homomorphisms induced by inclusion: $H_p^{i,j} = im\ f_p^{i,j}, \quad 0 \leq i \leq j \leq n$.
The corresponding p^{th} *Betti numbers* are the rank of these groups:

- $\beta_p^{i,j} = rank(H_p^{i,j})$.

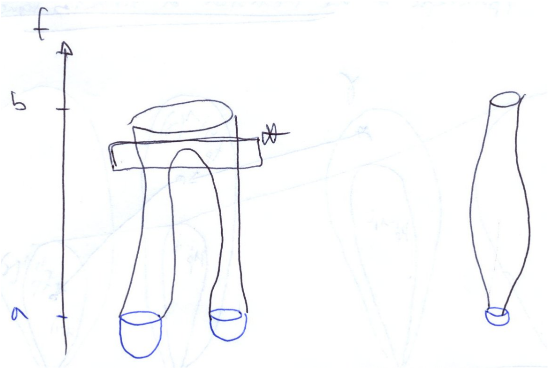


In pictures



- Image by inclusion: $H_0^{a,b} = im\ f_0^{a,b}$;
- A class of H_0 merges with another one in $*$, and then *dies*!
- $\beta_0(\mathbb{X}_a) = 3$;
- $\beta_0^{a,b} = 2$!
- This is the exact idea of the contour tree algorithm [CSA00].

In pictures



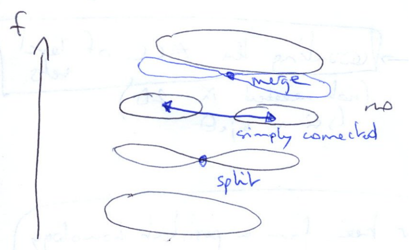
- Image by inclusion: $H_0^{a,b} = im\ f_0^{a,b}$;
- A class of H_0 merges with another one in $*$, and then *dies*!
- $\beta_0(\mathbb{X}_a) = 3$;
- $\beta_0^{a,b} = 2$!
- This is the exact idea of the contour tree algorithm [CSA00].

Contour trees from a persistent homology point of view

- Why does the contour tree algorithm really work?
- Let's have a look at the Reeb graph algorithm first:
  1. Reeb graphs are obtained by *"quotienting"* contours,
  2. **plus** by considering the resulting quotient topology.
- As a result from Morse theory [Mil63], branching in $\mathcal{R}(f)$ only occurs at critical values [Ree46]:
  - Warning! the inverse is not true in dimensions higher than 2.
- To know how classes connect to each other (2^{nd} part):
  - Observe how the connected components of level sets evolve;
  - ... especially at critical values (branching)!

Contour trees from a persistent homology point of view

- ...but simply-connected domains are very particular:
  - When two contours **merge**:
    - There's no way these two contours were connected before;
    - This would mean they had taken *"individual disconnected paths"*;
    - Impossible since the domain is simply-connected.

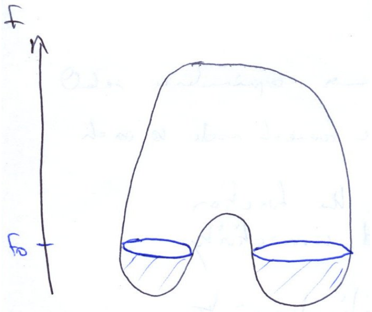


- Contours continuously pill on each other to form sub-level sets;
- ... without disconnecting sub-level sets!

- The classes of the 0^{th} persistent homology groups and of the 1^{st} persistent boundary groups evolve the same way!

Contour trees from a persistent homology point of view

- ...but simply-connected domains are very particular:
  - When two contours **merge**:
    - There's no way these two contours were connected before;
    - This would mean they had taken *"individual disconnected paths"*;
    - Impossible since the domain is simply-connected.

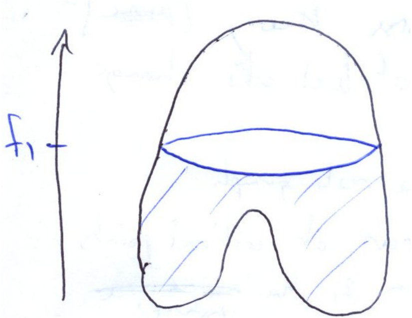


- Contours continuously pill on each other to form sub-level sets;
- ... without disconnecting sub-level sets!

- The classes of the 0^{th} persistent homology groups and of the 1^{st} persistent boundary groups evolve the same way!

Contour trees from a persistent homology point of view

- ...but simply-connected domains are very particular:
  - When two contours **merge**:
    - There's no way these two contours were connected before;
    - This would mean they had taken *"individual disconnected paths"*;
    - Impossible since the domain is simply-connected.



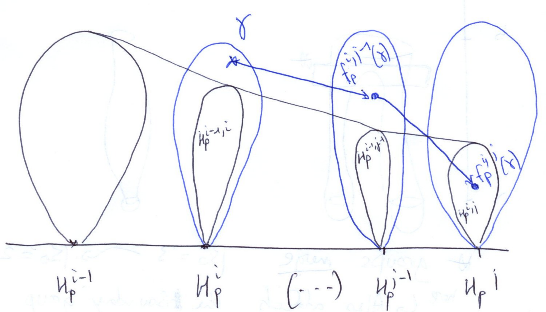
- Contours continuously pill on each other to form sub-level sets;
- ... without disconnecting sub-level sets!

- The classes of the 0^{th} persistent homology groups and of the 1^{st} persistent boundary groups evolve the same way!

Contour trees from a persistent homology point of view

- Then, we no longer need to keep track of contours;
- ... but of the connected component of the **sub-level sets**!
  - A UF structure on the filtration is now sufficient :)
- The same holds at split configurations (opposite of f).
- This give the *quotient topology* at critical values;
- What about regular values:
  - Merging the join-tree and the split-tree:
    - This is nothing but a **merge-sort**! (filtration);
    - Observe local connectivity every time we pick an edge.

Back to persistent homology groups

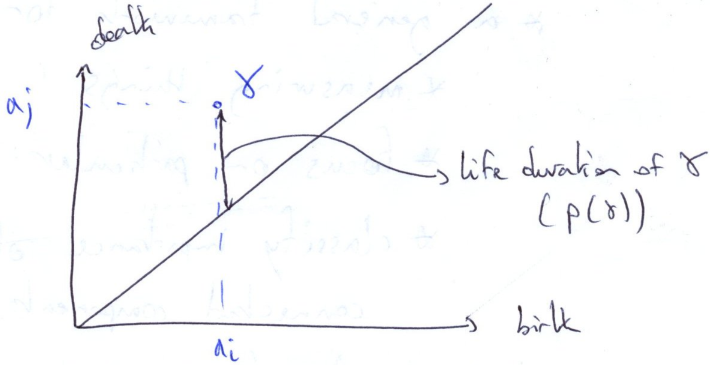


- $H_p^{i,j}$: homology classes living in K_i and still living in K_j;
- A given class $\gamma \in H_p^i$:
  - was born in K_i: $\gamma \notin H_p^{i-1,i}$;
  - died in K_j:
    - $f_p^{i,j-1}(\gamma) \notin H_p^{i-1,j-1}$;
    - $f_p^{i,j}(\gamma) \in H_p^{i-1,j}$.
  - Its life duration, **its persistence**, is $p(\gamma) = a_j - a_i$;
  - **Importance of a topological feature!**

This is great! ...but what's the point?

- So far:
  - Given a measuring system (f function),
  - We are able to evaluate scales on topological features,
  - And decide of their importance.
- But the super cool thing about homology is Betti numbers, right?
  - What about the persistent Betti numbers?

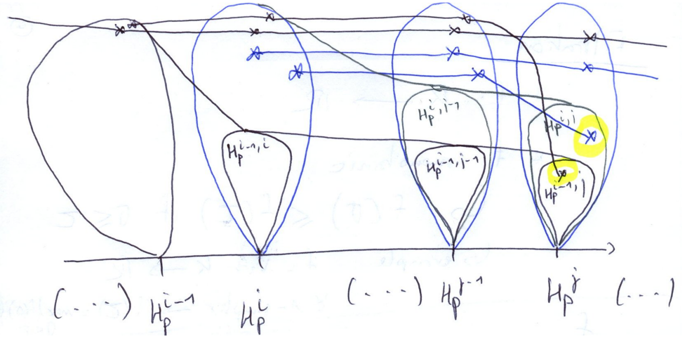
Persistence diagrams



- Draw classes in the plane, in function of their birth and death;
- Several classes can occur on the same spot! (same life);

Multiplicity (of life)

- Enumerate the classes born in K_i and dead in K_j (same spot):
    - $\mu_p^{i,j} = (\beta_p^{i,j-1} - \beta_p^{i,j}) - (\beta_p^{i-1,j-1} - \beta_p^{i-1,j})$.



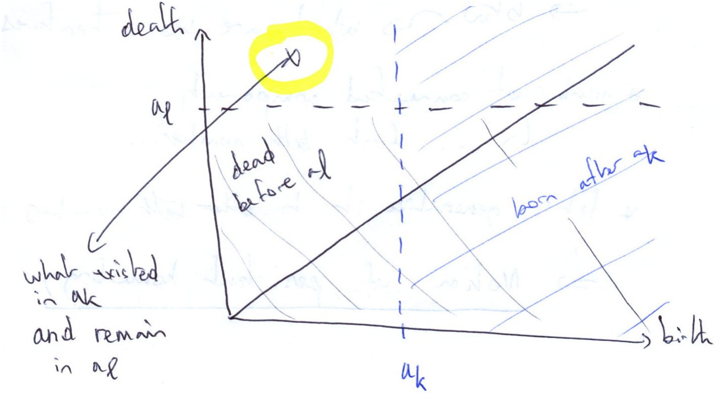
- $(\beta_p^{i,j-1} - \beta_p^{i,j})$: those living in K_i and dead in K_j (2 circles);
- $(\beta_p^{i-1,j-1} - \beta_p^{i-1,j})$: those living in K_{i-1} and dead in K_j (1 circle).

Persistent Betti numbers

Definition (p^{th} persistent Betti numbers)

For every pair of indices $0 \leq k \leq l \leq n$ and every dimension p, the p^{th} persistent Betti number is:

- $\beta_p^{k,l} = \sum_{i \leq k} \sum_{j > l} \mu_p^{i,j}$.



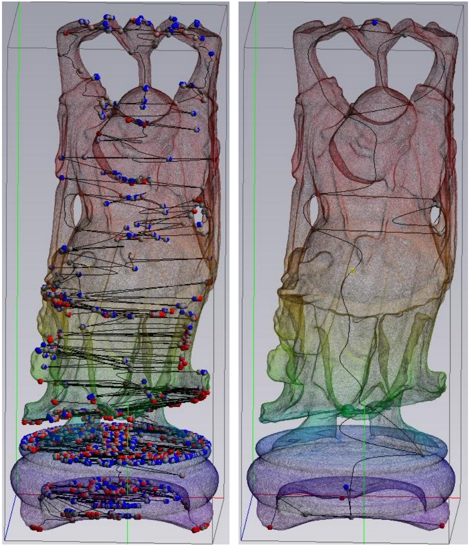
Yes, but how can we compute them then?

- Matrix reduction :) (still);
- Do we have to compute the Smith Normal form of the boundary matrices at each step of the filtration sequence?!!!
- It turns out that no :)
  - Run a slightly different reduction algorithm;
  - All the information we need appears;
  - See Herbert Edelsbrunner's course notes for more details.

Intermediary conclusion

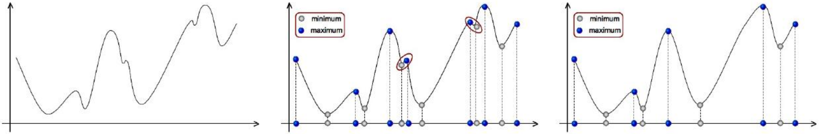
- Persistent homology brings a general framework for:
  - Measuring user-defined noise (f function);
  - On a user-defined scale;
  - To classify topological features (Betti numbers) by importance.
- Back to real life:
  - Great! We can filter topological noise now!

What's the trick here?



Persistence based Reeb graph simplification

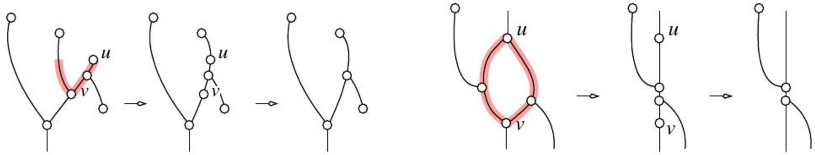
- Sort the arcs in term of persistence;
- Remove them one at a time:
  - Update adjacent arcs connectivity and persistence (elder's rule);
  - Until the user defined *persistence scale* is reached.
- 1-manifold example:



[GND*07]

Persistence based simplification in higher dimensions

- Some trivial cases:
  - Minimum - Joining saddle arc;
  - Maximum - Splitting saddle arc.
- Others:

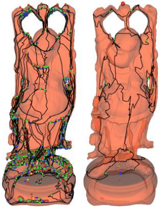


[PSBM07]

- The result is a filtered Reeb graph :)
- What about the initial function? Is it filtered too?

Back to geometry, everything's related :)

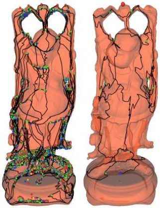
- Let's take the buddha example (2-manifold);
- Given the consistent filtered Reeb graph $\mathcal{R}(\hat{f})$:
  - How can we obtain the filtered version $\hat{f}$ of f?



[NGH04]

Back to geometry, everything's related :)

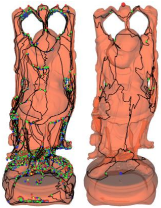
- Let's take the buddha example (2-manifold);
- Given the consistent filtered Reeb graph $\mathcal{R}(\hat{f})$:
  - How can we obtain the filtered version $\hat{f}$ of f?
- We need to constraint f so $\hat{f}$ admits critical values only at the critical nodes of $\mathcal{R}(\hat{f})$;



[NGH04]

Back to geometry, everything's related :)

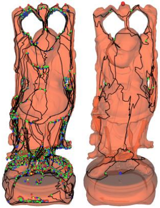
- Let's take the buddha example (2-manifold);
- Given the consistent filtered Reeb graph $\mathcal{R}(\hat{f})$:
  - How can we obtain the filtered version $\hat{f}$ of f?
- We need to constraint f so $\hat{f}$ admits critical values only at the critical nodes of $\mathcal{R}(\hat{f})$;
- Heat propagation process;
-
-
-
-



[NGH04]

Back to geometry, everything's related :)

- Let's take the buddha example (2-manifold);
- Given the consistent filtered Reeb graph $\mathcal{R}(\hat{f})$:
  - How can we obtain the filtered version $\hat{f}$ of f?
- We need to constraint f so $\hat{f}$ admits critical values only at the critical nodes of $\mathcal{R}(\hat{f})$;
- Heat propagation process;
- Laplace equation with non-homogeneous Dirichlet conditions:
  - $\Delta \hat{f}(p) = 0$
  - $\hat{f}(p) = f(p)$ if p corresponds to a critical node in $\mathcal{R}(\hat{f})$;
- Also a matrix reduction process :)
- See the *"Fair morse functions"* paper [NGH04].
- Now, is $\mathcal{R}(\hat{f})$ always the Reeb graph of $\hat{f}$?



[NGH04]

CARR H., SNOEYLINK J., AXEN U.:
Computing contour trees in all dimensions.
In *ACM Symposium on Discrete Algorithms* (2000), pp. 918–926.

DELFINADO C. J. A., EDELSBRUNNER H.:
An incremental algorithm for Betti numbers of simplicial complexes.
In *ACM Symposium on Computational Geometry* (1993), pp. 232–239.

GYULASSY A., NATARAJAN V., DUCHAINEAU M., PASCUCCI P., BRINGA E., HIGGINBOTHAM A., HAMANN B.:
Topologically clean distance fields.
IEEE Transactions on Visualization and Computer Graphics 13 (2007), 1432–1439.

MILNOR J.:
Morse Theory.
Princeton University Press, 1963.

NI X., GARLAND M., HART J.:

Fair Morse functions for extracting the topological structure of a surface mesh.
ACM Transactions on Graphics 23 (2004), 613–622.

Pascucci V., Scorzelli G., Bremer P. T., Mascarenhas A.:
Robust on-line computation of Reeb graphs: simplicity and speed.
ACM Transactions on Graphics 26 (2007), 58.1–58.9.

Reeb G.:
Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique.
Comptes-rendus des Séances de l'Académie des Sciences 222 (1946), 847–849.