Computational (co)homology - applications and recent progress in computations.

Paweł Dłotko
Institute of Computer Science, Jagiellonian University, Kraków, Poland.

(Joint work with Hubert Wagner)

ATMCS 2012.

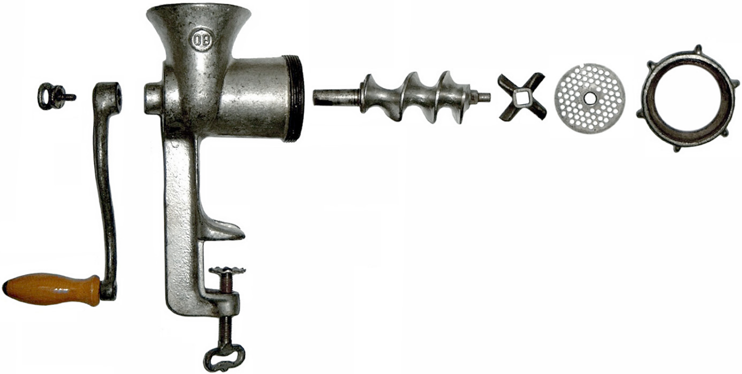
Motivation.

1. Text mining, persistence for text mining in cooperation with Google (Hubert's talk).
2. Persistent homology as a rigorous tool to analyze medical images.
3. Amount of data to process too big for available hardware $\Longrightarrow$ new algorithmic techniques needed.

Summary of the talk.

1. Iterated Morse complex and homology (over a field) – reminder form Hubert's talk.
2. Iterated Morse complex and persistent homology simplification.
3. Obtaining persistent intervals by Discrete Morse Theory.
4. Parallel and distributed Morse complex construction.
5. Sketch on distributed algorithms to TDA.

Summary of the talk.



Iterated Morse Complex.

- Homology over a field $\Longrightarrow$ pairing between A, B can be made iff $\kappa(A, B) \neq 0$.
- Algorithm to construct Morse complex – a functor $\mathbb{M} : \mathbb{C} \to \mathbb{C}$.
- $\mathbb{C}$ category of chain complexes.
- **Assumption:** if there are some Morse pairings in C, at least one of them is made in $\mathbb{M}(C)$.
- E.g. $\mathbb{M}$ procedure search for a single possible pairing and do it.

Iterated Morse Complex and homology.

- *Iterated Morse complex* – iterated application of $\mathbb{M}$.
- Homology is preserved, homotopy type is not.
- Iterated Morse complex gives homology of chain complex C over a field.
- $\underset{n \in \mathbb{N}}{\exists} \mathbb{M}^n(C) = \mathbb{M}^{n+1}(C) = \ldots = \mathbb{M}^{\infty}(C)$.
- Every cell $A \in \mathbb{M}^{\infty}(C)$, boundary and coboundary of A are empty.
- $\beta_i(C) = \#\{$ cells in $\mathbb{M}^{\infty}(C)$ of dimension $i\}$.
- Generators can be obtained from this procedure.
- Idea of iterating Morse complex construction for homology simplification – Rutgers, Krakow.

Persistence preserving simplification.

- Iterated Morse complex can be used to compute homology.
- Can it be useful for computations of persistence?
- Can we do Morse complex construction so that persistent homology is preserved?

State of the art.

- Vanessa Robins, optimal Morse complex for 3 dimensional gray-scale images.
- Optimality - every cell of iterated Morse complex creates or kills a nonzero length persistent interval.
- Result: minimal cell complex that store the same information about persistence as the original one.
- Vidit Nanda and Konstantin Mischaikow – Morse Theory for persistence.

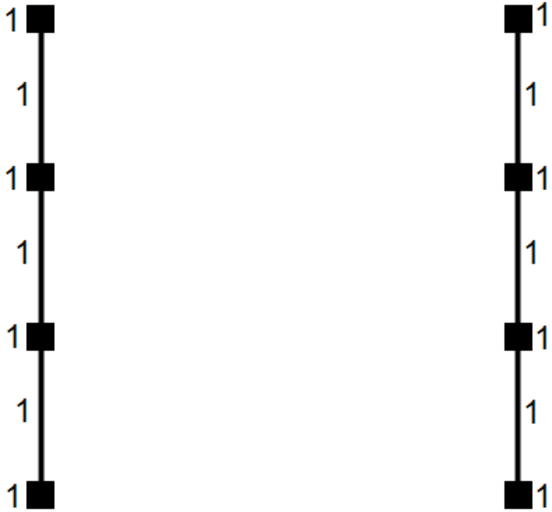
Morse for persistence.

- C – chain complex with filtration $g : C \to \mathbb{Z}$.
- s.t. $a, b \in C$, $a < b \Longrightarrow g(a) \leq g(b)$.
- Morse pairing $v : C \nrightarrow C$ is *compatible with filtration* if $g(a) = g(v(a))$ for every paired a.
- V-paths cannot go upwards filtration.
- **Assumption:** $\mathbb{M}$ constructs only a vector fields compatible with filtration.
- Persistence of C and $\mathbb{M}(C)$ are the same.

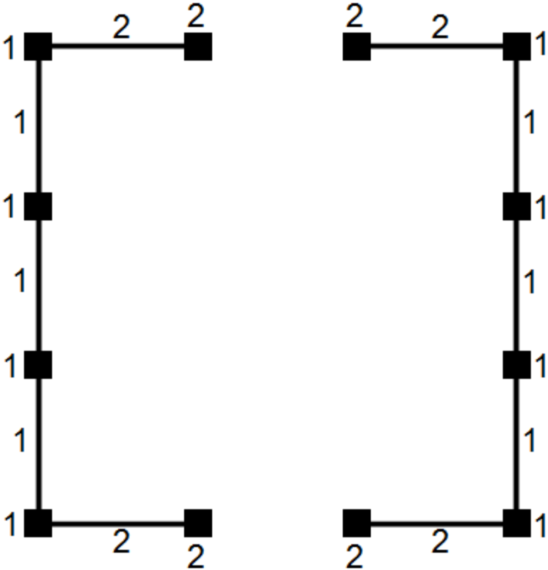
Iterated Morse Complex for persistence.

- Iterated Morse complex construction can be applied for filtered complex C.
- $\underset{n \in \mathbb{N}}{\exists} \mathbb{M}^n(C) = \mathbb{M}^{n+1}(C) = \ldots = \mathbb{M}^\infty(C)$.
- $\mathbb{M}^\infty(C)$ is optimal in sense of Vanessa.
- Every cell of $\mathbb{M}^\infty(C)$ either creates or kills nonzero persistent interval.

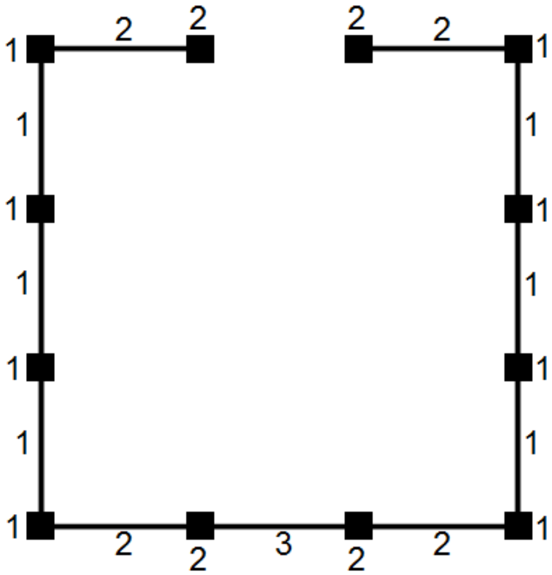
Persistent preserving simplification.



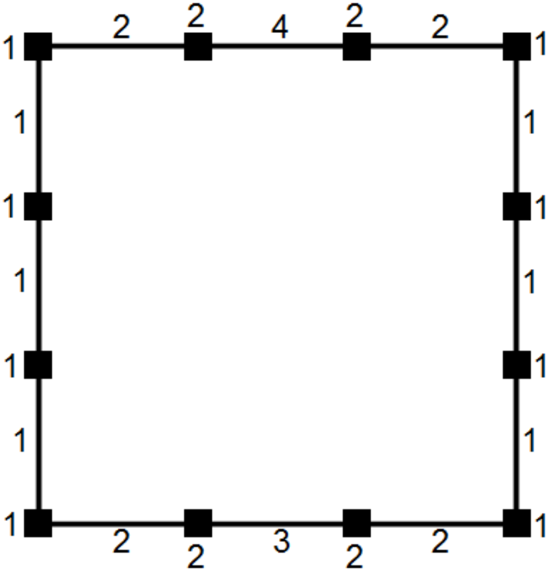
Persistent preserving simplification.



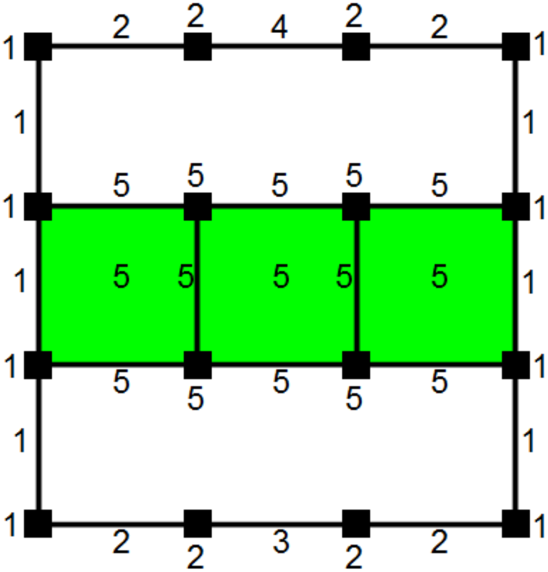
Persistent preserving simplification.



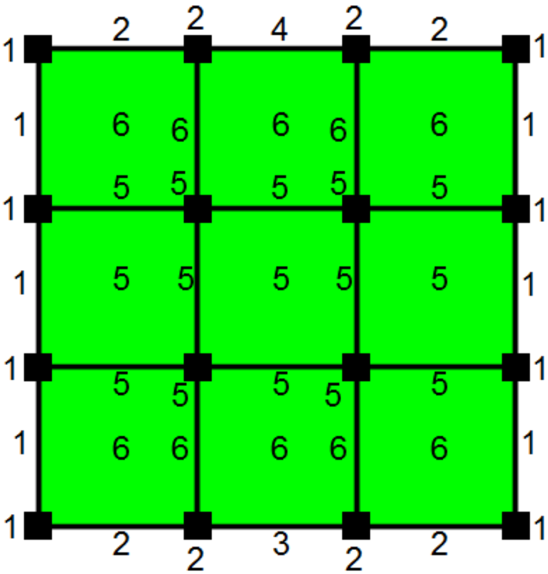
Persistent preserving simplification.



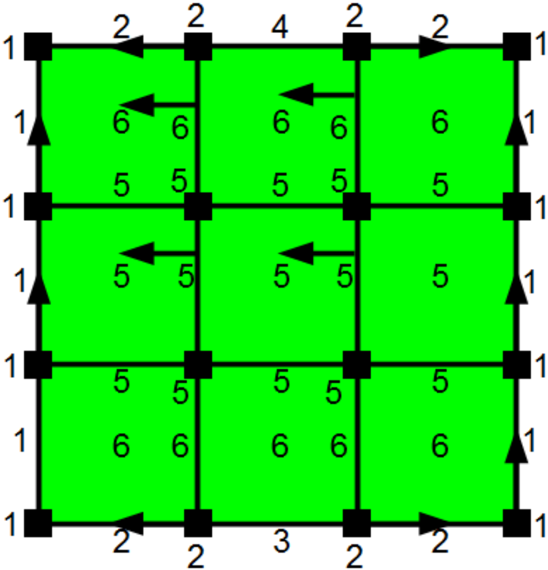
Persistent preserving simplification.



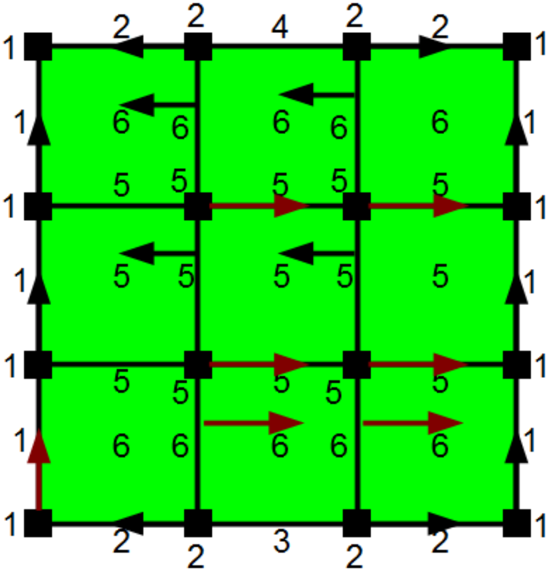
Persistent preserving simplification.



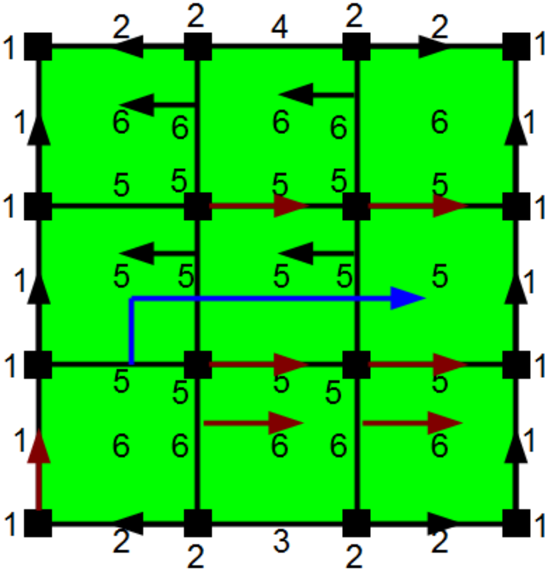
Persistent preserving simplification.



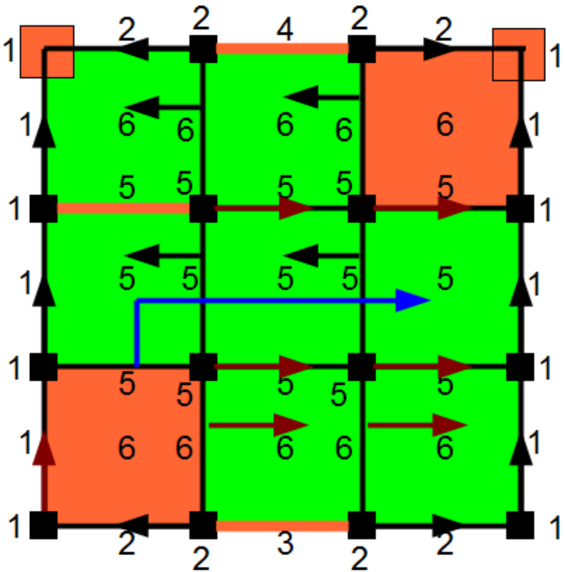
Persistent preserving simplification.



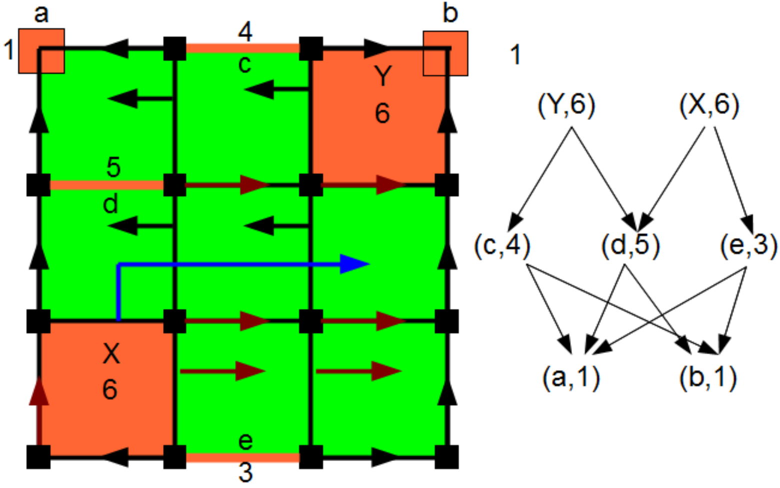
Persistent preserving simplification.



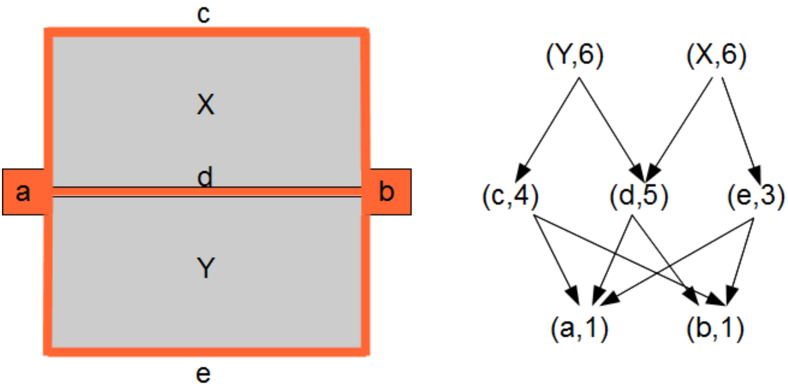
Persistent preserving simplification.



Persistent preserving simplification.



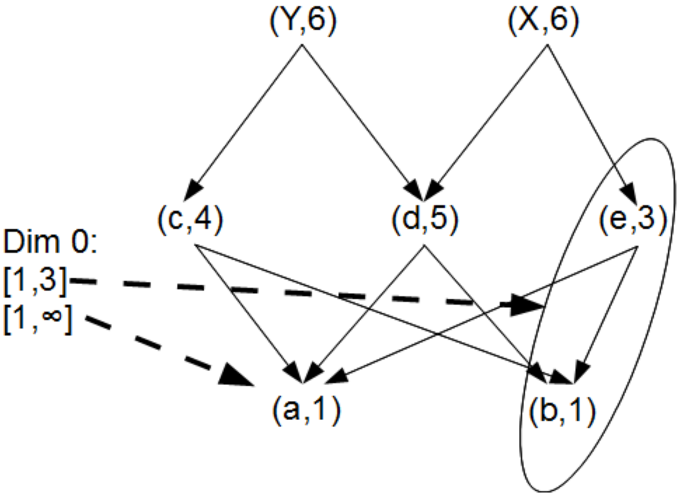
Persistent preserving simplification.



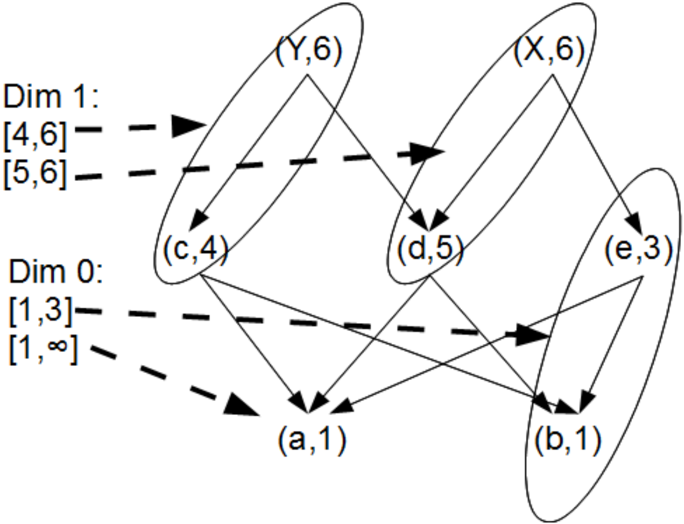
Observations.

- $A \in M^\infty(C)$, and $B_1, \ldots, B_n$ be in boundary of A in $M^\infty(C)$.
- $g(A) > g(B_1), \ldots, g(B_n)$.
- $M^\infty(C)$ is the minimal cell complex with the same persistence as C.

Observations.



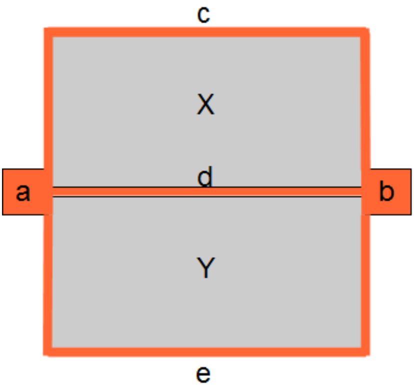
Observations.

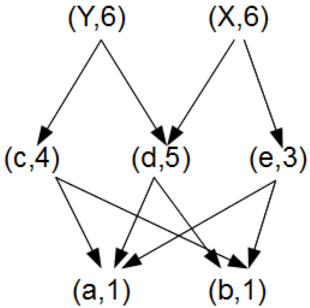


Persistent Homology via DMT.

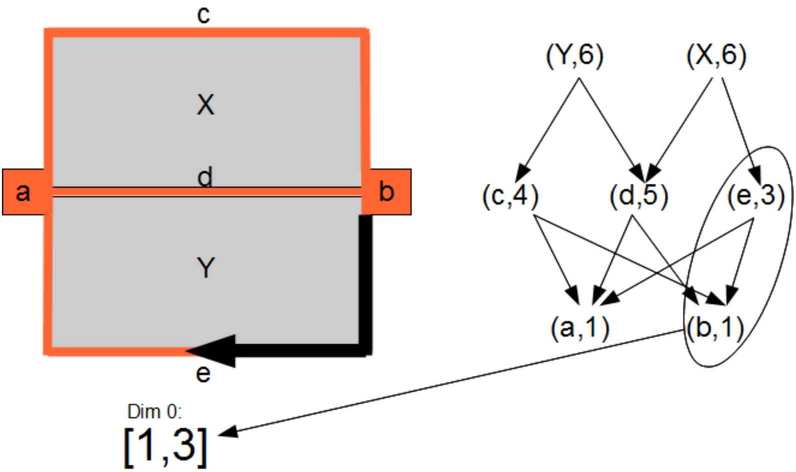
- Based on Morse theory one can obtain persistent intervals.
- No need to change representation for one suitable for matrix algorithm.
- Unlike the simplification phase, cells of *different filtration value* are paired and nonzero persistent intervals are reported.
- Pairings between cells of different filtration value – allowed (to some extent).
- Iterated Morse complex construction in the increasing order of filtration values.
- Eldest rule: Cell A is always paired with maximal (in sense of filtration) cell in its boundary.
- This is equivalent to a variation of the algebraic algorithm.

Level 0.

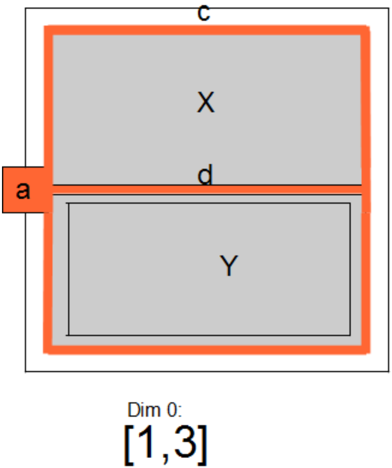


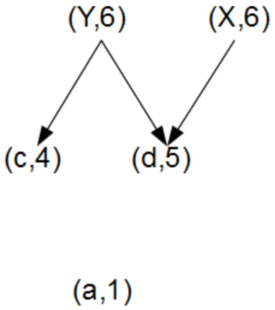


Level 3.

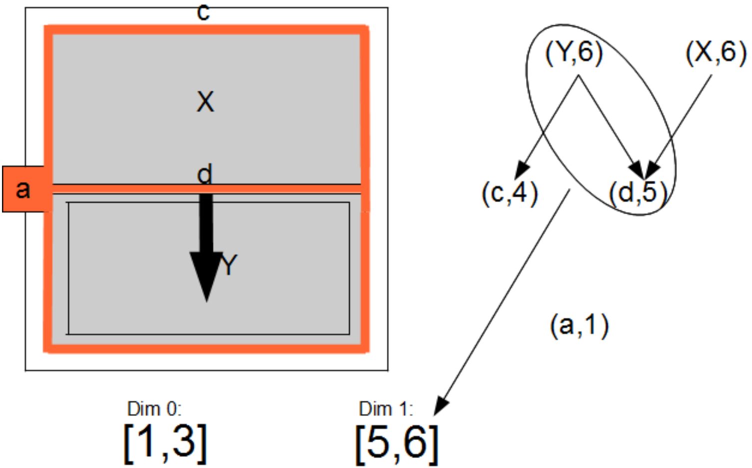


Level 3.

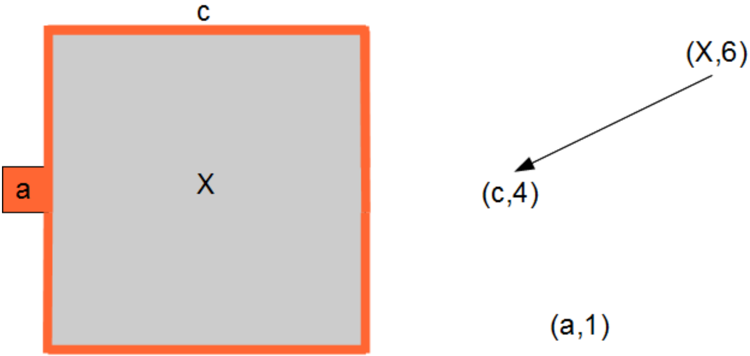




Level 6.



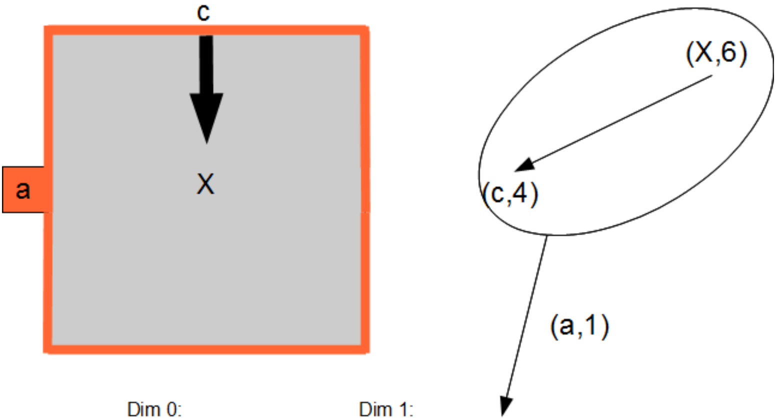
Level 6.



Dim 0:
[1,3]

Dim 1:
[5,6]

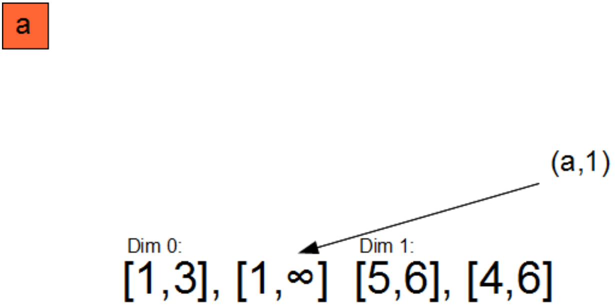
Level 6.



Dim 0:
[1,3]

Dim 1:
[5,6], [4,6]

Level ∞.



What if...?

- ... we run out of RAM?
- Typical problems in TDA, image analysis.
- Solution – parallel or distributed algorithms for (persistent) homology.
- Mikael V. Johansson, Primoz Skraba – spectral sequences and M-V theorem for persistence (work in progress).
- Morse theory approach to parallelization / distribution?

Distributed, parallel.

- For a moment we **go back** from persistent **to standard homology**.
- Single Morse pairing – very local operation.
- DAG property required – somehow global (ensure no directed cycles).
- Computing incidence index in Morse complex – somehow global.
- Distribution / parallelization require good strategy of doing pairings so that:
  - Resulting vector field is acyclic – *consistent*.
  - Easy/trivial to compute incidence index locally – *locally indexable*.
  - As long as some pairings can be done, at least one will be done – *vital*.

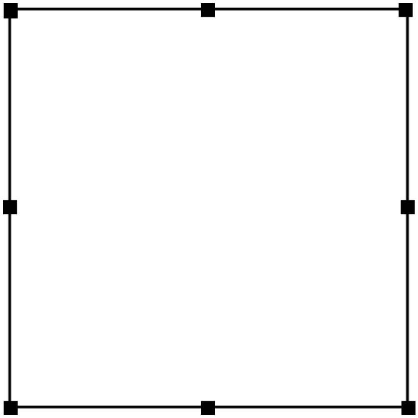
Algorithms.

1. Cone contraction algorithm (very large data).
2. Spanning tree pairing algorithm (medium data).

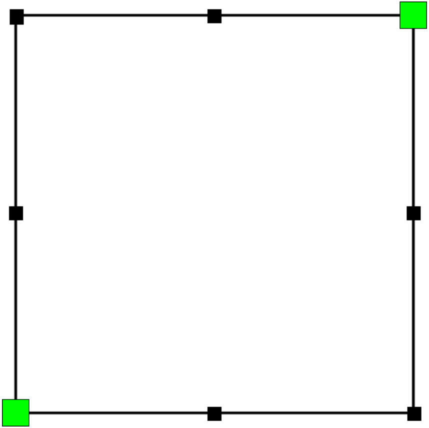
Cone contraction algorithm.

1. Boundaryless cell with nonempty coboundary – `cone`.
2. Let us have a set of cones in our complex lying at least 3 hops one from another.
3. Iterative Morse pairings of simplices in the star of the cone.
4. Later the state of the complex is written back to HDD.
5. When there are no more cones, finish.
6. Consistent, locally indexable, vital.

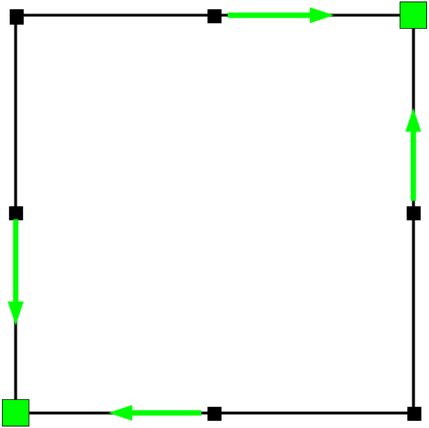
Simple graph example.



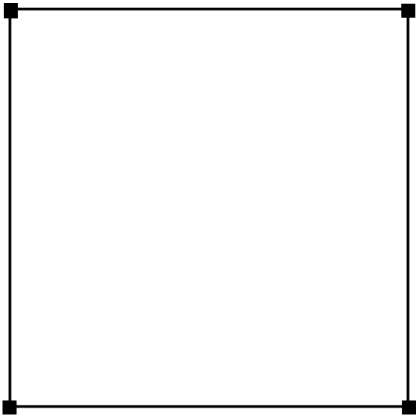
Simple graph example.



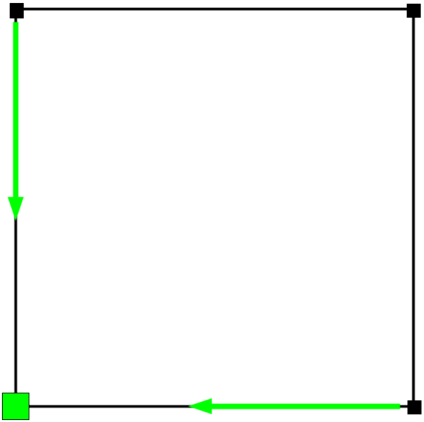
Simple graph example.



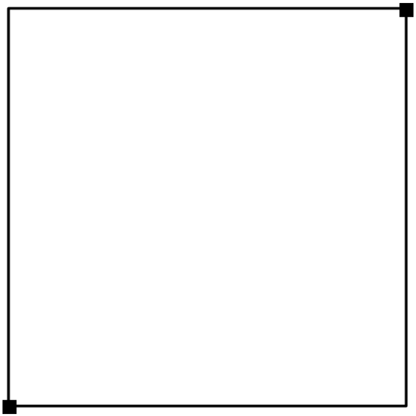
Simple graph example.



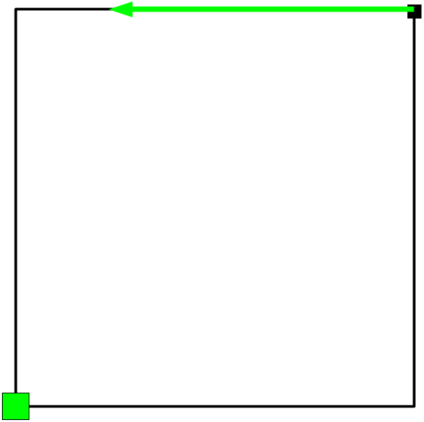
Simple graph example.



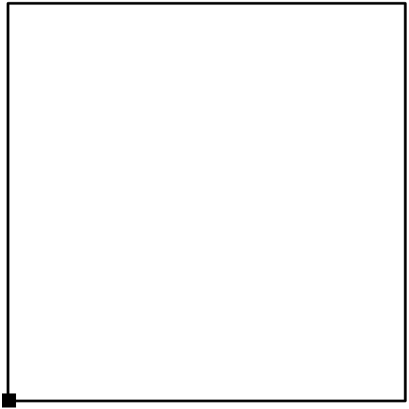
Simple graph example.



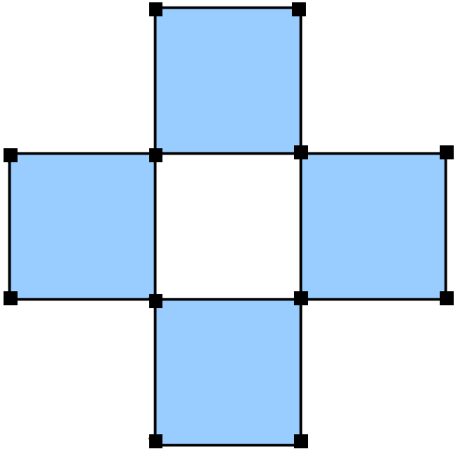
Simple graph example.



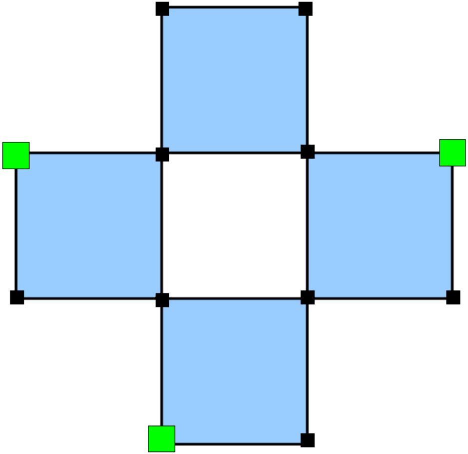
Simple graph example.



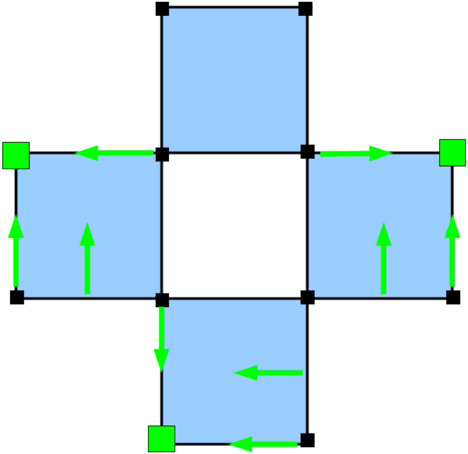
More complicated example.



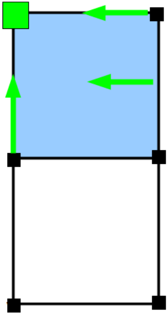
More complicated example.



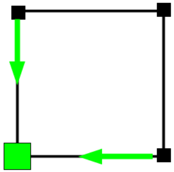
More complicated example.



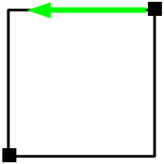
More complicated example.



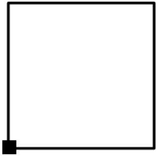
More complicated example.



More complicated example.



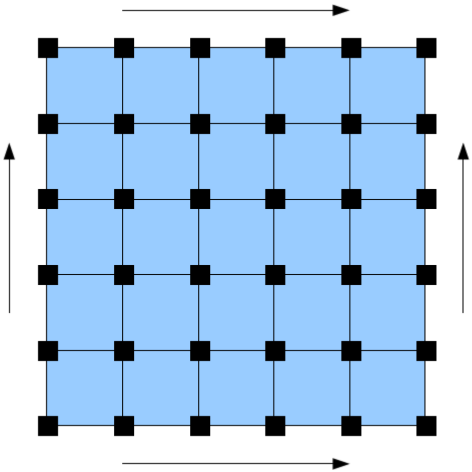
More complicated example.



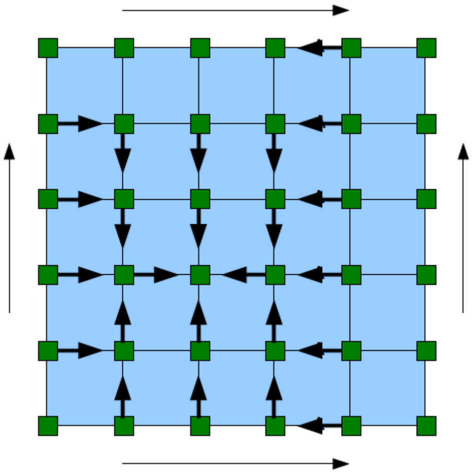
Spanning tree pairing algorithm.

1. Usage of distributed / parallel algorithms to construct spanning trees of pairings (spanning tree protocol).
2. Require partial synchronization:
   2.1 Pairings between cells in dimensions $(0, 1), (2, 3), (4, 5), \ldots$.
   2.2 Pairings between cells in dimensions $(1, 2), (3, 4), (5, 6), \ldots$.
3. Domain on which trees can be created are distinct.

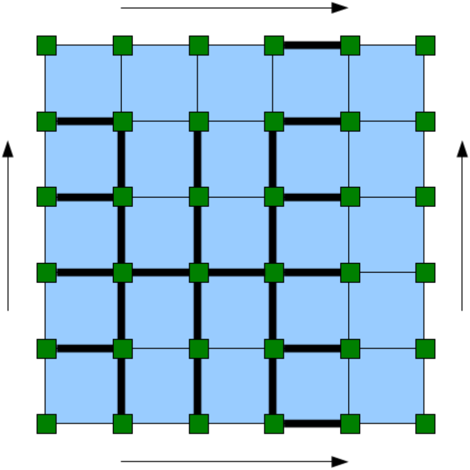
Domain.



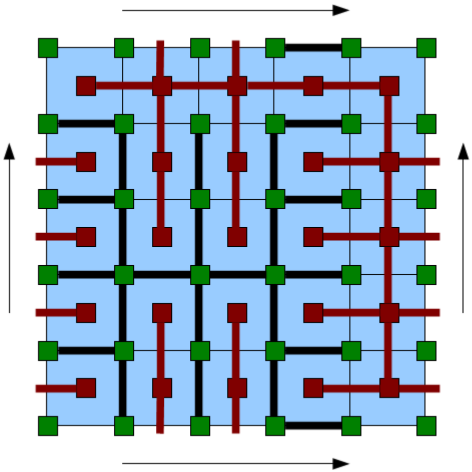
Phase 1.



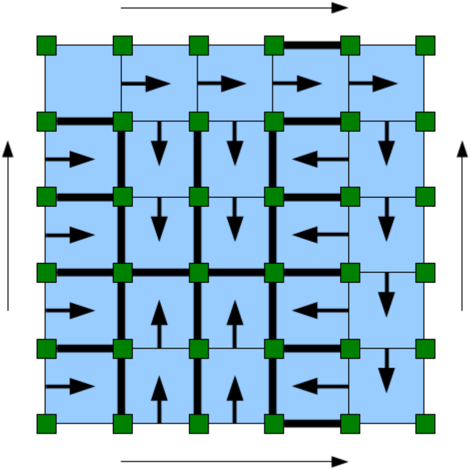
Phase 2.



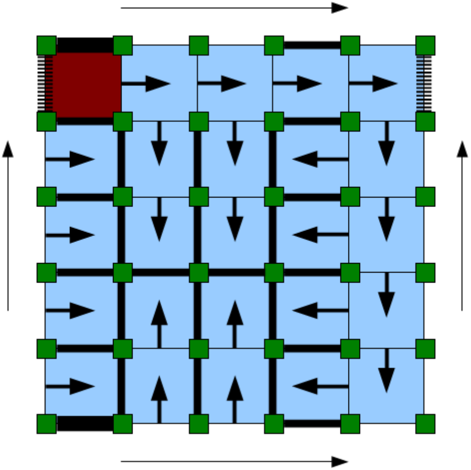
Phase 2.



Phase 2.



We are done.



Spanning tree pairing algorithm.

1. Generalization of tree-cotree technique.
2. For 2 dimensional complexes and 2 dimensional manifold two steps suffices.
3. In the "writing to HDD" phase, require some extra work to tie the boundaries (technicality).
4. Requires keeping only two constitutive dimensions in memory.
5. Consistent, vital, **not locally indexable**.

Back to persistent!

- We know how to compute persistent homology with Morse theory.
- We know how to distribute Morse complex computations.
- Time to go back to persistence, and see how to distribute it with the help of Discrete Morse Theory.

Trivial/nontrivial in persistent homology parallelization.

- Two phases in persistent homology computations via Morse:
  1. Pairings between cells of the same filtration value, removing intervals of persistence zero.
  2. Pairings between cells of different filtration value (in increasing order) + reporting intervals.
- Every cell in iterated Morse complex (after first phase) either creates or kills nonzero length persistent interval.
- **First phase** – trivially parallelizable – adaptation of any presented technique.
- Longest phase when we do not have to bother about filtration order.
- **Second phase** – levels of filtration need to be processed in order (require synchronization).

First phase.

- Pairs between elements of the same level of filtration.
- In the cone contraction and spanning tree we add this restriction.
- Usually in cone/tree pairs have very different filtration values.
- This step is done as long as possible, $\mathcal{M}^\infty$ obtained (provided strategy is vital).

Second phase.

- Global synchronization is required.
- Let 2 be the second lowest level of filtration.
- Presented algorithms are used to do all the pairings between cells of filtration ≤ 2 in $\mathcal{M}^{\infty}$.
- Global synchronization – broadcasting filtration levels that are allowed in pairings.

Distributed construction of VR complex.

1. Want to apply this technique to TDA.
2. Computing homology of point cloud with VR complex - NP-hard. No widely used alternative so far.
3. Need a distributed algorithm to construct huge VR complexes.
4. Similar as construction of VR complex from sensor network (P.D., R. Ghrist, M. Juda, M. Mrozek).

Point Cloud

TDA pipeline.

Point Cloud

(P.D., R. Ghrist, M. Juda, M. Mrozek)

Vietoris-Rips Complex

TDA pipeline.

Point Cloud

(P.D., R. Ghrist, M. Juda, M. Mrozek)

Vietoris-Rips Complex

Iterated Morse complex

(with H. Wagner)

TDA pipeline.

Point Cloud

(P.D., R. Ghrist, M. Juda, M. Mrozek)

Vietoris-Rips Complex

Iterated Morse complex

Persistent intervals

(with H. Wagner)

TDA pipeline.

Point Cloud

(P.D., R. Ghrist, M. Juda, M. Mrozek)

Vietoris-Rips Complex

Distributed / parallel Morse complex construction.

Persistent intervals

(with H. Wagner)

TDA pipeline.

Point Cloud

(P.D., R. Ghrist, M. Juda, M. Mrozek)

Vietoris-Rips Complex

Distributed / parallel Morse complex construction.

Storing on HD / DFS

(work in progress, with M. Robinson)

Persistent intervals

(with H. Wagner)

Other applications.

- First phase of persistence computation – optimal complex with the given persistence. Visualization? Data compression?
- Implementation of those methods in sensor networks:
  1. Coverage problems in 2/3/4d.
  2. Computing Euler characteristic by sensor network.
- ...

Take home message.

- Homology and persistence computed via Morse.
- Can be viewed just as variation of matrix reduction, but...
- Iterated Morse complex – **intuitive** way of computing (persistent) homology.
- All is **combinatorial**, easy to explain to students.
- Based on **graph theory** – lots of available algorithms.
- In theory no big problems with **parallelization** and **distribution**. No overhead (n^3).

The end.

Thank you for your attention!

Contact info: Paweł Dłotko, Institute of Computer Science, Jagiellonian University, Kraków, Poland.
mail : pawel.dlotko@ii.uj.edu.pl
www.ii.uj.edu.pl/~dlotko