A Heuristic for Short Homology Basis of Digital Objects

Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari

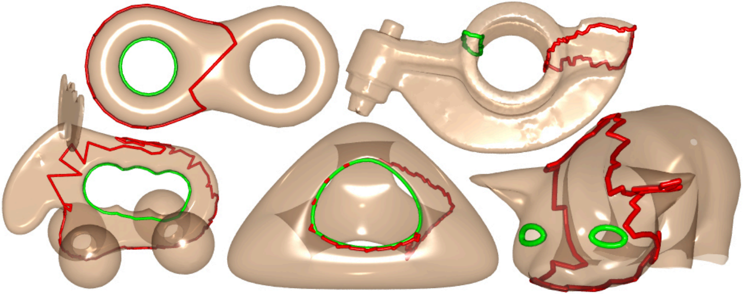
Aix-Marseille Université / CNRS, LIS UMR 7020

October 26, 2022

Outline

Outline

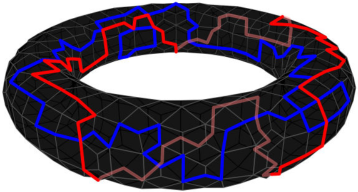
Problem : visualize the holes in a digital object

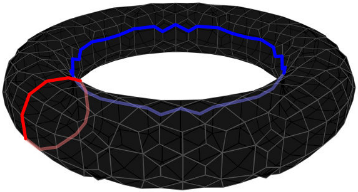


Why ? Identify errors, measure holes

Approach:

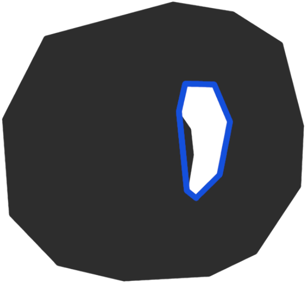
- Visualize holes as homology cyles ($\sim$ discrete manifolds)
- Short cycles are better (tight, well located)





Computing a minimal homology basis

- size of cycle: number of cells
- $q = 1$: $\mathcal{O}(n^{\omega} + n^2 \cdot \beta_1)$[1]
- $q > 1$: NP-hard[2]

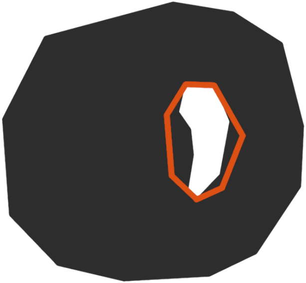


[1] Dey et al: Efficient algorithms for computing a minimal homology basis (2018)

[2] Chen and Friedman: Hardness results for homology localization (2011)

Computing a minimal radius homology basis

- size of cycle: radius of smallest geodesic ball containing it
- $q \geq 1$: $\mathcal{O}(\beta_q \cdot n^4)$ [3], then $\mathcal{O}(n^{\omega+1})$ [4]
- Theoretical algorithm, not implemented



[3] Chen and Friedman: Measuring and computing natural generators for homology groups (2011)

[4] Dey et al: Efficient algorithms for computing a minimal homology basis (2018)

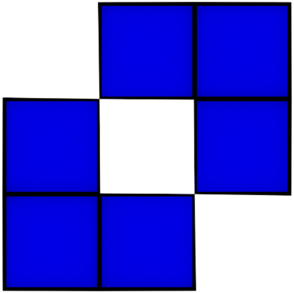
Goal: we want an algorithm that

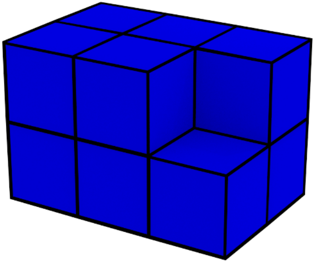
1. can be implemented in practice
2. is faster (even if it is worse)

Outline

Digital object

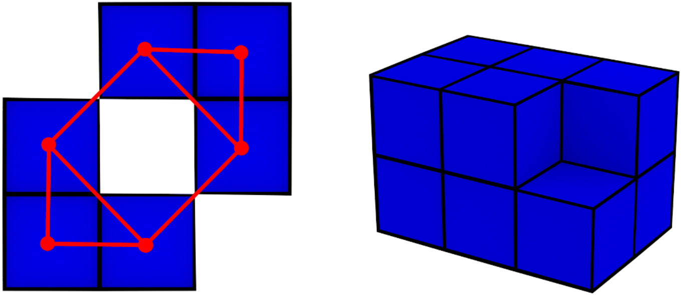
A nD digital object is a subset of $\mathbb{Z}^n$





Digital object

A nD digital object is a subset of $\mathbb{Z}^n$



Connectivity relation: $(3^n - 1)$-connectivity (or $2n$).

Signed distance transform

Let O be a digital object,

$$sdt_O(x) = \begin{cases} -d(x, O^c) & \text{if } x \in O \\ d(x, O) & \text{if } x \notin O \end{cases}$$

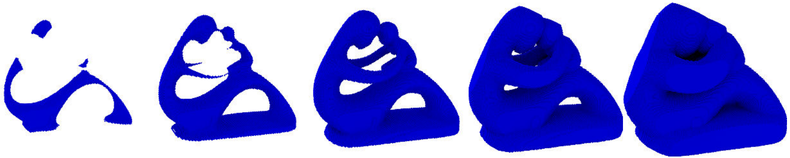
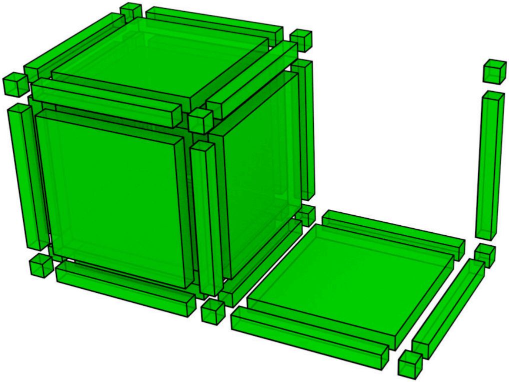


Figure: Sublevel sets of the signed distance transform

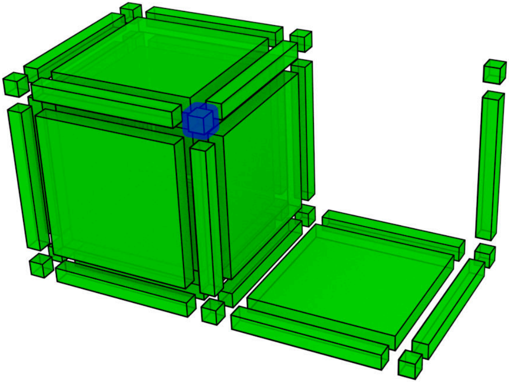
Cubical complex

Union of points, edges, squares, cubes, ... (cubes) closed under inclusion



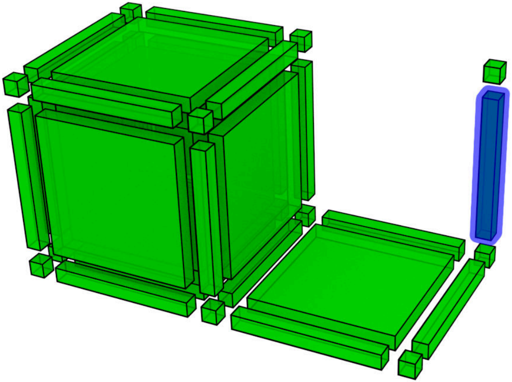
Cubical complex

Union of points, edges, squares, cubes, ... (cubes) closed under inclusion



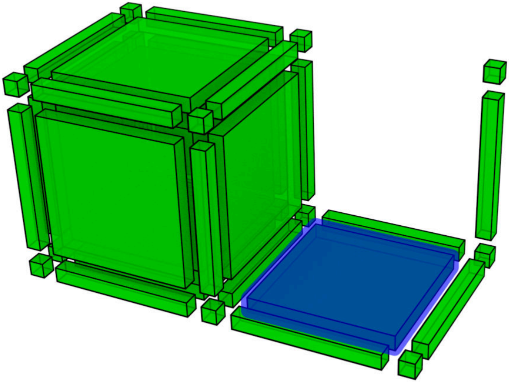
Cubical complex

Union of points, edges, squares, cubes, ... (cubes) closed under inclusion



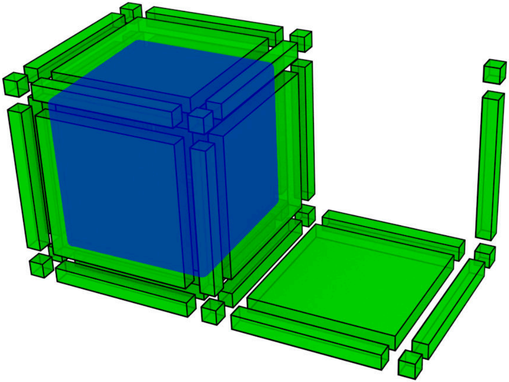
Cubical complex

Union of points, edges, squares, cubes, ... (cubes) closed under inclusion

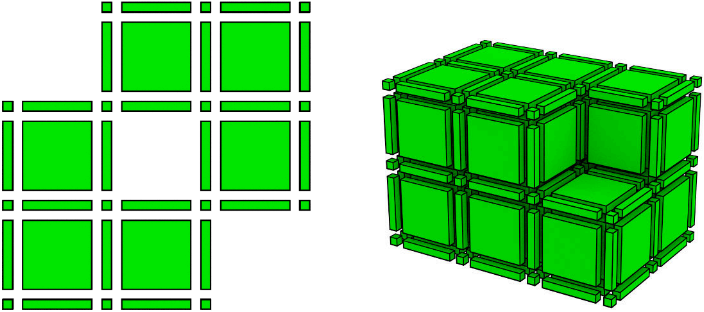


Cubical complex

Union of points, edges, squares, cubes, ... (cubes) closed under inclusion

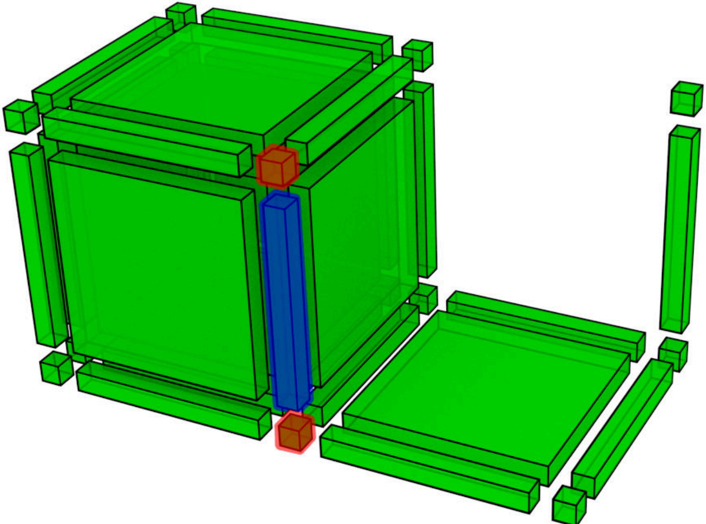


Digital object $\longrightarrow$ cubical complex ($(3^n - 1)$-connectivity)



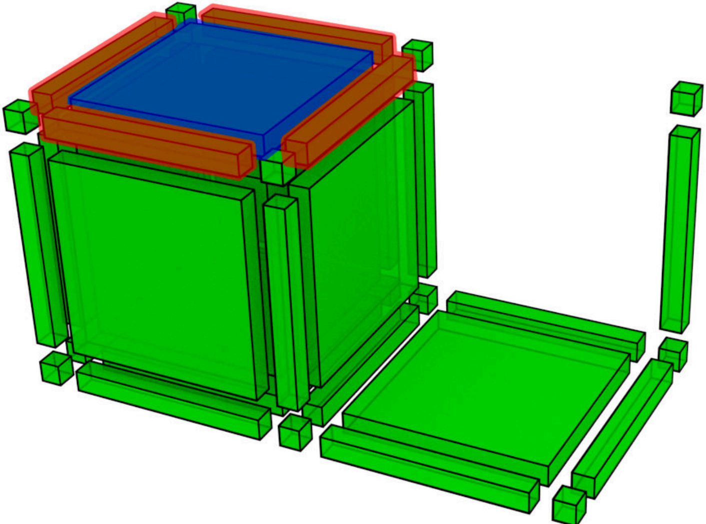
Blue: 1-cube

Red: its boundary (faces)



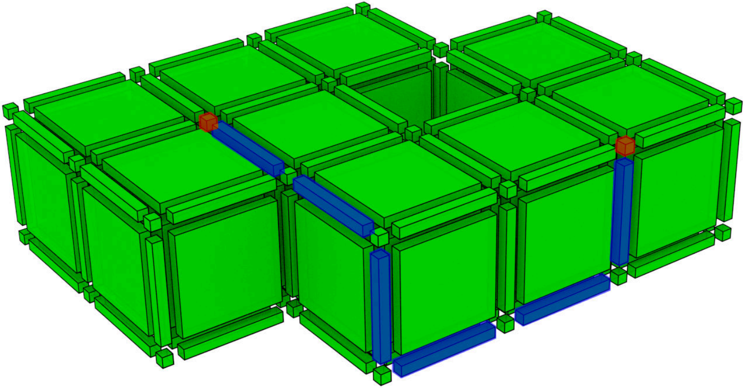
Blue: 2-cube

Red: its boundary (faces)

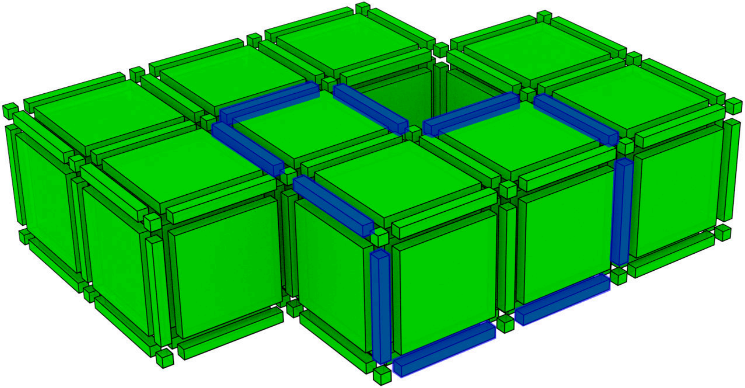


Blue: 1-chain

Red: its boundary

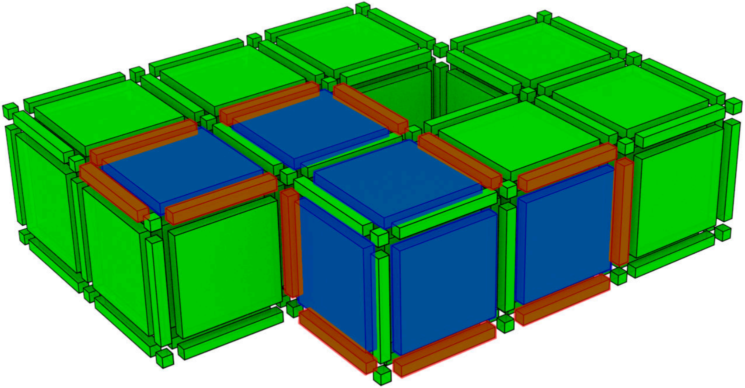


Blue: 1-chain (1-cycle)
Red: its boundary ($= \emptyset$)



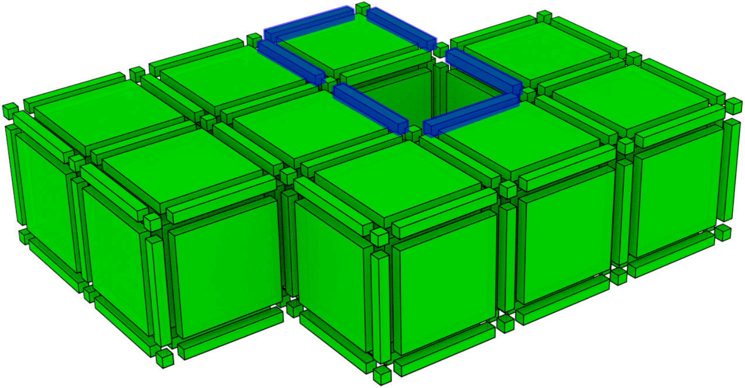
Blue: 2-chain

Red: its boundary (1-cycle)



Blue: 1-chain (1-cycle, but not boundary)

Red: its boundary ($= \emptyset$)



- K cubical complex
- Chain complex of K

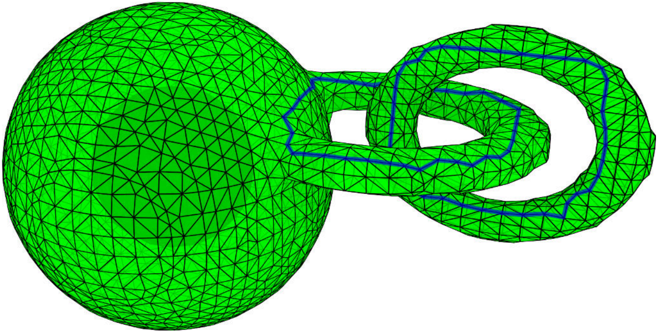
$$\cdots C_3 \xrightarrow{d_3} C_2 \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{d_0} 0$$

where $d_q d_{q+1} = 0 \Rightarrow \mathrm{im}(d_{q+1}) \subset \ker(d_q)$

- q-dimensional homology group (vector space) $H_q(K) := \ker(d_q)/\mathrm{im}(d_{q+1})$[5] $= (\mathbb{F}_2)^{\beta_q}$

[5] $\forall x, y \in \ker(d_q), x \sim y \Leftrightarrow x + y \in \mathrm{im}(d_{q+1})$

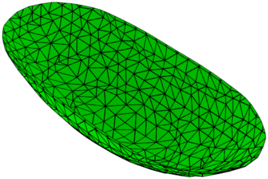
- $\beta_q := \dim(H_q(K))$: number of holes (connected components, tunnels, cavities...)
- Homology basis: set of independent classes $\rightarrow$ cycles

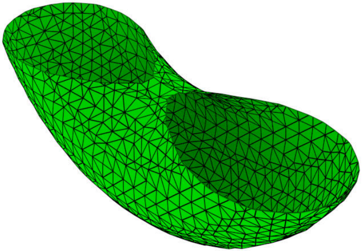


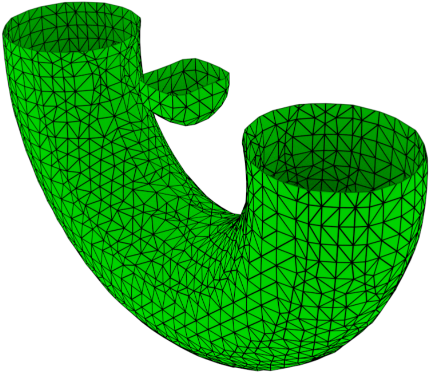
- Filtration F: $K_1 \subset K_2 \subset K_3 \subset \cdots$

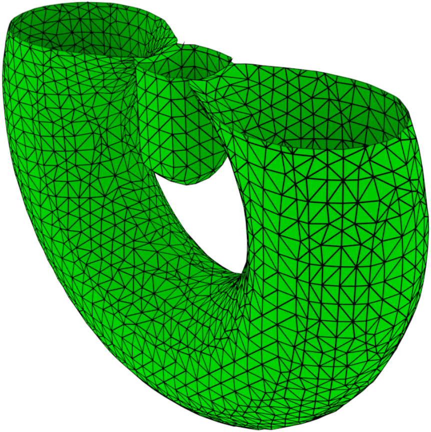
$$\begin{array}{ccccccc} K_1 & \xrightarrow{\iota} & K_2 & \xrightarrow{\iota} & K_3 & \xrightarrow{\iota} & \cdots \\ \downarrow & & \downarrow & & \downarrow & & \\ H(K_1) & \xrightarrow{\iota_*} & H(K_2) & \xrightarrow{\iota_*} & H(K_3) & \xrightarrow{\iota_*} & \cdots \end{array}$$

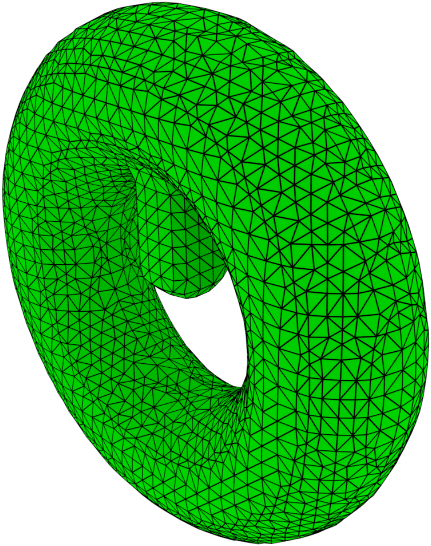
- Usually defined by a function $f : K \to \mathbb{R}$
- $\beta_{i,j} = \dim(\iota : H(K_i) \to H(K_j))$
  number of holes in K_i still in K_j
- $\mu_{i,j} = \beta_{i,j} - \beta_{i,j+1} - \beta_{i-1,j} + \beta_{i-1,j+1}$
  number of holes born in K_i and dying in K_j











Sketch of persistent homology computation:

- Sort cells according to the filtration
- For each cell, associate it with one of the previous ones
- Each of these pairs makes a persistence pair

Persistence pairs

$PD(F) = \{(i, j) \text{ with multiplicity } \mu_{i,j}\}$

We also obtain:

- A partial matching of cells
- A cycle associated to each pair ($\simeq$ hole)

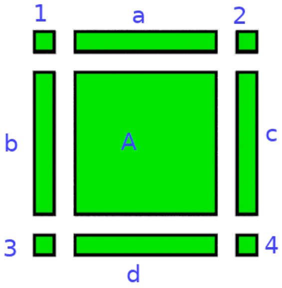
Homological Discrete Vector Field

Let K be a cubical complex and $P, S \subset K, P \cap S = \emptyset$.
(P, S) is a HDVF if $j_P \circ \partial \circ i_S$ is an isomorphism.

where

- $i_S : S \to K$ inclusion
- $j_P : K \to P$ projection

$(j_P \circ \partial \circ i_S)$ is an invertible submatrix of the boundary matrix



$$\partial_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \partial_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$P = \{1, 2, 3, 4\}, S = \{a, b, c, d\}$ is not a HDVF (not invertible)
$P = \{1, 2, 3, a\}, S = \{a, b, c, A\}$ is not a HDVF ($P \cap S \neq \emptyset$)
$P = \{1, 2, 3, d\}, S = \{a, b, c, A\}$ is a HDVF

A maximal HDVF provides:

- A homology basis
- A cohomology basis
- Check if a cycle is trivial
- Check if two cycles are homologous
- ...

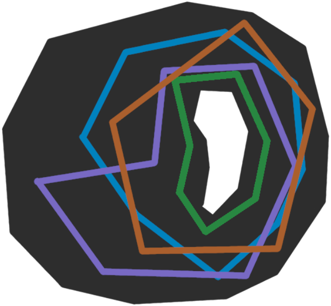
Outline

Simplest case: only one hole

1. Get some candidate (non trivial) cycles
2. Keep the shortest one

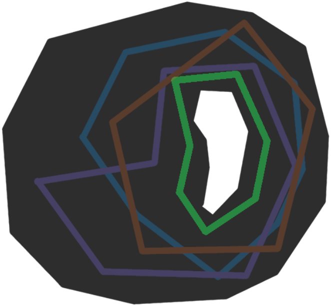
Simplest case: only one hole

1. Get some candidate (non trivial) cycles
2. Keep the shortest one



Simplest case: only one hole

1. Get some candidate (non trivial) cycles
2. Keep the shortest one

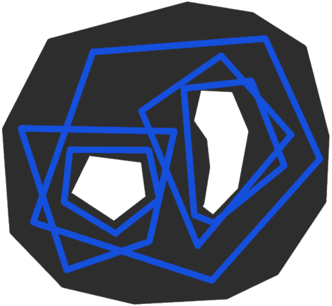


General case: more than one hole

1. Get some candidate (non trivial) cycles
2. Keep the shortest cycles that make a homology basis

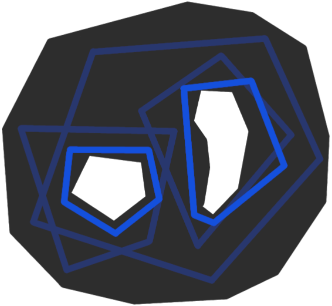
General case: more than one hole

1. Get some candidate (non trivial) cycles
2. Keep the shortest cycles that make a homology basis



General case: more than one hole

1. Get some candidate (non trivial) cycles
2. Keep the shortest cycles that make a homology basis



Pitfalls:

1. How to find good candidate cycles?
2. A cycle is trivial or not?
3. A set of cycles is a homology basis ?

Our algorithm has 4 steps

1. Breadth balls (good starting points)
2. Compute filtrations (candidate cycles)
3. Sort and annotate (cycles to classes)
4. Earliest basis (shortest homology basis)

Notation :

- K is the cubical complex of a digital object O
- We fix $q \geq 1$ and we look for a short homology basis for $H_q(K)$
- $g := \dim(H_q(K))$

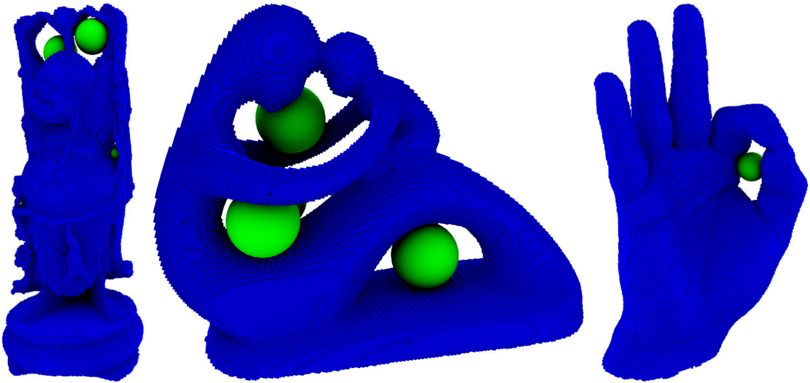
Step 1 - Breadth balls

- Full[6] cubical complex + signed distance function f_{sdt} = filtration
- Compute partial matching $M = \{(\sigma_1, \tau_1) \ldots\}$
- Get $\mathcal{B}$ = cells τ such that
  1. $(\sigma, \tau) \in M$
  2. $f_{sdt}(\sigma) < 0 < f_{sdt}(\tau)$
  3. $\dim(\sigma) = q$

These are the centers of the breadth balls of the digital object O

[6] Other approaches: convex hull, homotopical closing

Step 1 - Breadth balls



Step 2 - Compute filtrations

For each $\tau \in \mathcal{B}$ ($\tau \notin K$), we compute a discrete geodesic transform:

Discrete geodesic transform

Let K be a cubical complex, $\tau_0 \in K$ a 0-cube,

$$f_{K,\tau_0}(\sigma) = \begin{cases} \text{size of shortest path to } \tau_0 & \text{if } \dim(\sigma) = 0 \\ \max\{f_{K,\tau_0}(\sigma') : \sigma' \subset \sigma, \dim(\sigma') = 0\} & \text{if } \dim(\sigma) > 0 \end{cases}$$

Step 2 - Compute filtrations

Then, for each $\tau \in \mathcal{B}$

- Let $\tau_0 \in K$ be a closest 0-cube to τ
- f_{K,τ_0} is a filtration over K
- Persistent homology computation $\rightarrow \{x_1 \ldots x_g\}$ a homology basis for K
- $|\mathcal{B}| = q \Rightarrow$ we compute g homology bases $= g^2$ cycles

Step 3 - Sort and annotate

We sort the g^2 cycles by size: $\{x_1 \ldots x_{g^2}\}$

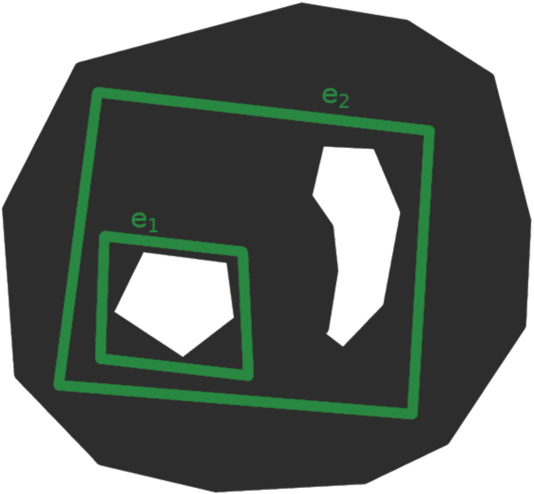
To find a basis, we have to write these cycles in a homology basis: an *annotation*

Annotation

Homomorphism $a : K_q \rightarrow H_q(C)$ such that for any two q-cycles x, y, $a(x) = a(y)$ iff $[x] = [y]$.

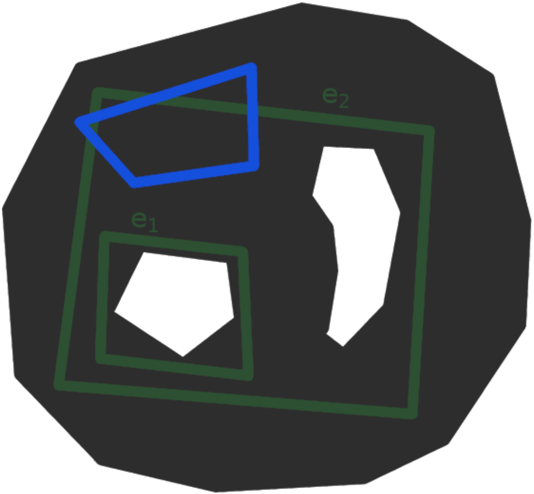
This is simply a linear map that projects each cycle onto the homology group, a $g \times |K_q|$ matrix.

Step 3 - Sort and annotate



$$H_1(K) = \langle e_1, e_2 \rangle$$

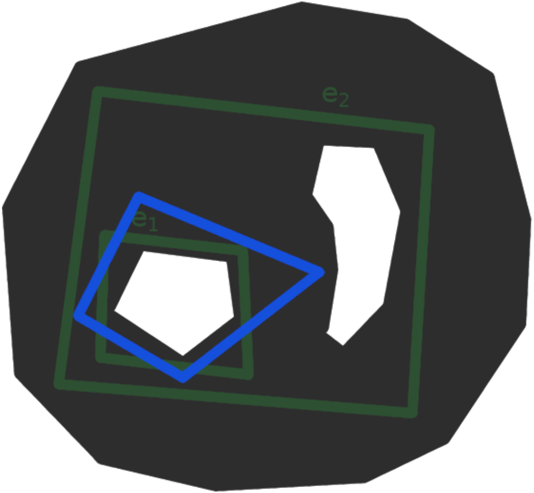
Step 3 - Sort and annotate



$$H_1(K) = \langle e_1, e_2 \rangle$$

$$a(x) = 0$$

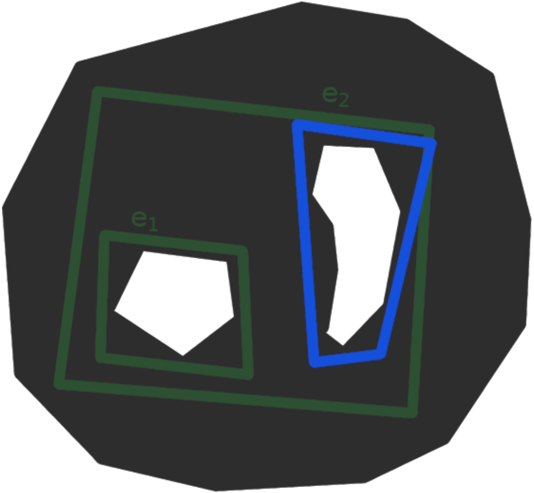
Step 3 - Sort and annotate



$$H_1(K) = \langle e_1, e_2 \rangle$$

$$a(x) = e_1$$

Step 3 - Sort and annotate



$$H_1(K) = \langle e_1, e_2 \rangle$$

$$a(x) = e_1 + e_2$$

Step 3 - Sort and annotate

We compute an annotation with a HDVF:

1. Let $M = \bigcup_{i \in I} \{(\sigma_i, \tau_i)\}$ be the partial matching of cells computed by any of the filtrations in step 2
2. Define $P := \{\sigma : (\sigma, \tau) \in M, \dim(\sigma) = q\}$ and $S := \{\tau : (\sigma, \tau) \in M, \dim(\tau) = q + 1\}$
3. (P, S) is a HDVF and the linear map (matrix)

$$a = (j_C \circ \partial_{q+1} \circ i_S) \circ (j_P \circ \partial_{q+1} \circ i_S)^{-1} \circ j_P + j_C$$

   is an annotation
4. Computation: get $A = (j_C \circ \partial_{q+1} \circ i_S)$, $B = (j_P \circ \partial_{q+1} \circ i_S)$, and compute $A \cdot B^{-1}$

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1) \dots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

col:

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1) \ldots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

col: 1

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1) \ldots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

col: 1

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1)\dots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

col: 2

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1) \dots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

col: 2

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1) \dots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

col: 6

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1) \dots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

col: 6

Step 4 - Earliest basis

- Let Y be the $g \times g^2$ matrix with columns $a(x_1) \dots a(x_{g^2})$
- Get the first columns to make a basis with gaussian elimination
- Return the corresponding cycles

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

col: 6

Short homology basis: $\{x_1, x_2, x_6\}$

Focus on the algorithm by Dey et al (2018):

1. Compute one filtration per 0-cube (fast matrix multiplication)
2. Sort all $|K_0| \cdot g$ cycles
3. Compute annotation (LSP-decomposition, fast matrix multiplication)
4. Earliest basis using (LSP-decomposition)

This is a theoretical algorithm using fast matrix multiplication

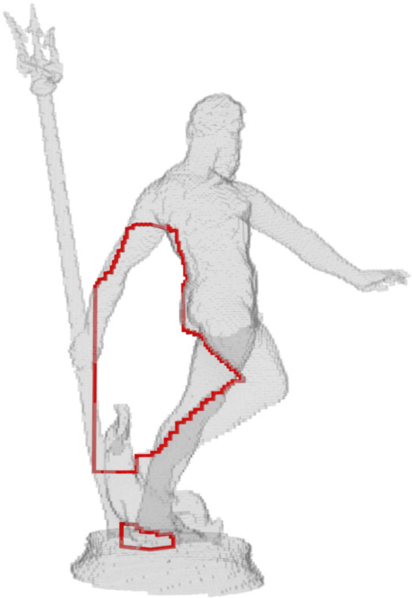
We simplify it to compare it with ours:

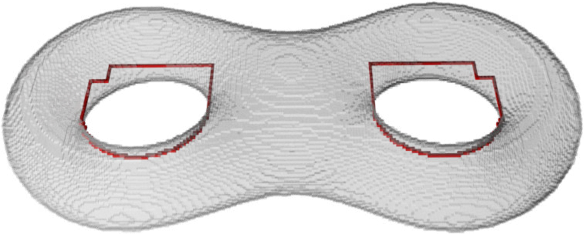
1. Compute g filtrations from random 0-cubes
2. Sort all $g \cdot g$ cycles
3. Compute annotation (with a HDVF)
4. Earliest basis (with gaussian elimination)

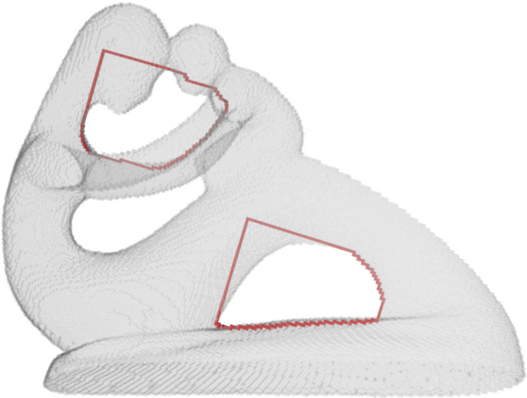
This adaptation can be computed and has a similar time complexity

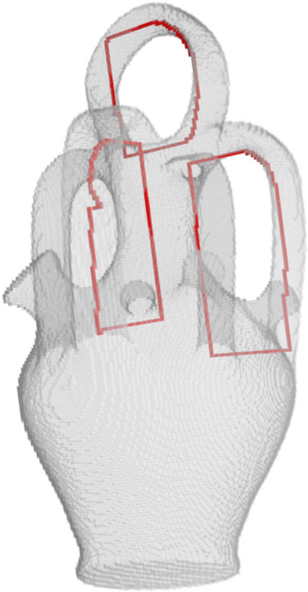
Table: Average sizes of the homology bases produced by both algorithms

Object	Us	Dey et al*
Amphora	930	1068.0
Dancing	902	1042.6
Eight	356	417.0
Fertility	962	999.8
Neptune	640	663.0
Pegasus	1032	1182.8









Outline

Conclusion:

- Heuristic for short homology generators
- Dey et al is not that bad
- $\mathcal{O}(m^3 + g \cdot n^3)$ time complexity
- Easy to implement, check our code

Perspectives:

- Two step algorithm: initial solution + optimization
- Improve length (area, volume. . .) estimation
- Faster algorithm for the annotation

Perspectives:

- Two step algorithm: initial solution + optimization
- Improve length (area, volume...) estimation
- Faster algorithm for the annotation

Thanks! Questions?