Fast Computation of Zigzag Persistence

Tamal K. Dey, Tao Hou https://arxiv.org/abs/2204.11080

Problem: Despite having the same time complexity, computation of zigzag persistence is more involved and generally takes much longer than ordinary persistence

Idea: transform zigzag sequence into a regular sequence in such a way that the barcode of the original sequence can be read from that of the new one, then apply optimized algorithms for regular persistence

Note: each inclusion $K_i \leftrightarrow K_{i+1}$ is an addition or deletion of a *single* cell;

$$\mathcal{F}: \varnothing = K_0 \xleftrightarrow{\sigma_0} K_1 \xleftrightarrow{\sigma_1} \cdots \xleftrightarrow{\sigma_{m-1}} K_m = \varnothing$$

$$\hat{\mathcal{E}}: L_0 \cup \{\omega\} \hookrightarrow \cdots \hookrightarrow L_n \cup \{\omega\} = \hat{K} \cup \omega \cdot L_{2n} \hookrightarrow \hat{K} \cup \omega \cdot L_{2n-1} \hookrightarrow \cdots \hookrightarrow \hat{K} \cup \omega \cdot L_n$$

Overview

1. Convert $\mathcal{F}$ into a *non-repetitive* zigzag filtration of Δ-complexes.
2. Convert the non-repetitive filtration to an *up-down* filtration.
3. Convert the up-down filtration to a *non-zigzag* filtration with the help of an *extended persistence* filtration.

$$\mathcal{F} : \varnothing = K_0 \overset{\sigma_0}{\longleftrightarrow} K_1 \overset{\sigma_1}{\longleftrightarrow} \cdots \overset{\sigma_{m-1}}{\longleftrightarrow} K_m = \varnothing$$

$$\hat{\mathcal{F}} : \varnothing = \hat{K}_0 \overset{\hat{\sigma}_0}{\longleftrightarrow} \hat{K}_1 \overset{\hat{\sigma}_1}{\longleftrightarrow} \cdots \overset{\hat{\sigma}_{m-1}}{\longleftrightarrow} \hat{K}_m = \varnothing$$

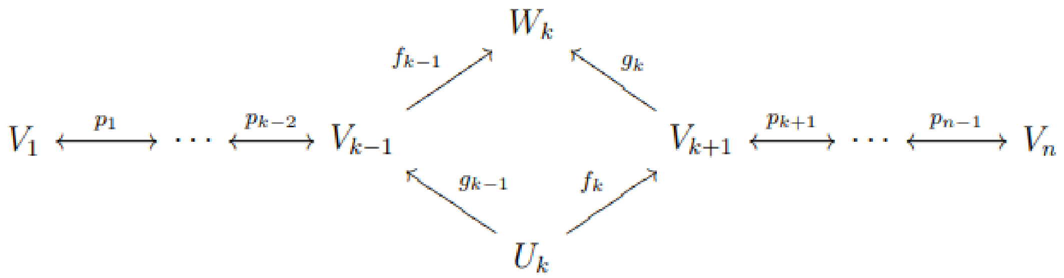
$$\mathcal{U} : \varnothing = L_0 \hookrightarrow L_1 \hookrightarrow \cdots \hookrightarrow L_n \hookleftarrow L_{n+1} \hookleftarrow \cdots \hookleftarrow L_{2n} = \varnothing$$

$$\mathcal{E} : \varnothing = L_0 \hookrightarrow \cdots \hookrightarrow L_n = (\hat{K}, L_{2n}) \hookrightarrow (\hat{K}, L_{2n-1}) \hookrightarrow \cdots \hookrightarrow (\hat{K}, L_n) = (\hat{K}, \hat{K})$$

$$\hat{\mathcal{E}} : L_0 \cup \{\omega\} \hookrightarrow \cdots \hookrightarrow L_n \cup \{\omega\} = \hat{K} \cup \omega \cdot L_{2n} \hookrightarrow \hat{K} \cup \omega \cdot L_{2n-1} \hookrightarrow \cdots \hookrightarrow \hat{K} \cup \omega \cdot L_n$$

Diamond Principle

From: Carlsson & De Silva, Zigzag Persistence



Definition 5.5. We say that the diagram

$$\begin{array}{ccc} V_{k+1} & \xrightarrow{g_k} & W_k \\ {\scriptstyle f_k}\uparrow & & \uparrow{\scriptstyle f_{k-1}} \\ U_k & \xrightarrow{g_{k-1}} & V_{k-1} \end{array}$$

is **exact** if $\mathrm{Im}(D_1) = \mathrm{Ker}(D_2)$ in the following sequence

$$U_k \xrightarrow{D_1} V_{k-1} \oplus V_{k+1} \xrightarrow{D_2} W_k$$

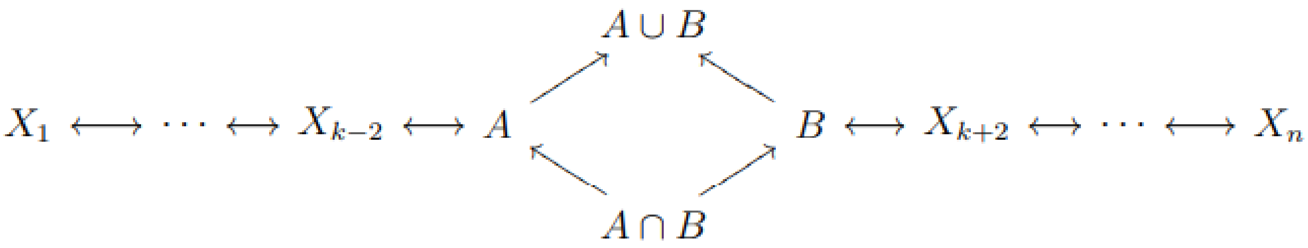
where $D_1(u) = g_{k-1}(u) \oplus f_k(u)$ and $D_2(v \oplus v') = f_{k-1}(v) - g_k(v')$.

Theorem 5.6 (The Diamond Principle). *Given $\mathbb{V}^+$ and $\mathbb{V}^-$ as above, suppose that the middle diamond is exact. Then there is a partial bijection of the multisets $\mathrm{Pers}(\mathbb{V}^+)$ and $\mathrm{Pers}(\mathbb{V}^-)$, with intervals matched according to the following rules:*

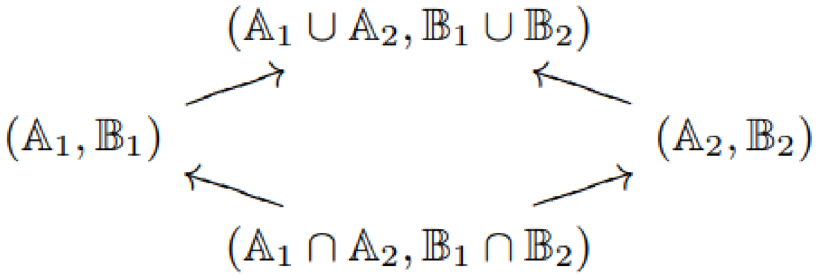
- *Intervals of type $[k, k]$ are unmatched.*
- *Type $[b, k]$ is matched with type $[b, k-1]$ and vice versa, for $b \le k-1$.*
- *Type $[k, d]$ is matched with type $[k+1, d]$ and vice versa, for $d \ge k+1$.*
- *Type $[b, d]$ is matched with type $[b, d]$, in all other cases.*

Mayer-Vietoris Diamond

Mayer-Vietoris



Relative Mayer-Vietoris



Mayer-Vietoris Diamond

Definition 5 (Mayer-Vietoris diamond [6]). Two cell-wise filtrations $\mathcal{F}$ and $\mathcal{F}'$ are related by a *Mayer-Vietoris diamond* if they are of the following forms (where $\sigma \neq \tau$):

$$
\begin{array}{lccccccc}
\mathcal{F}: & & & & K_j & & & \\
 & & & \overset{\sigma}{\nearrow} & & \overset{\tau}{\nwarrow} & & \\
 & K_0 \longleftrightarrow \cdots \longleftrightarrow & K_{j-1} & & & & K_{j+1} & \longleftrightarrow \cdots \longleftrightarrow K_m \\
 & & & \underset{\tau}{\nwarrow} & & \underset{\sigma}{\nearrow} & & \\
\mathcal{F}': & & & & K'_j & & &
\end{array}
\qquad (1)
$$

Theorem 7 (Diamond Principle [6]). *Given two cell-wise filtrations $\mathcal{F}, \mathcal{F}'$ related by a Mayer-Vietoris diamond as in Equation (1), there is a bijection from $\mathsf{Pers}_*(\mathcal{F})$ to $\mathsf{Pers}_*(\mathcal{F}')$ as follows:*

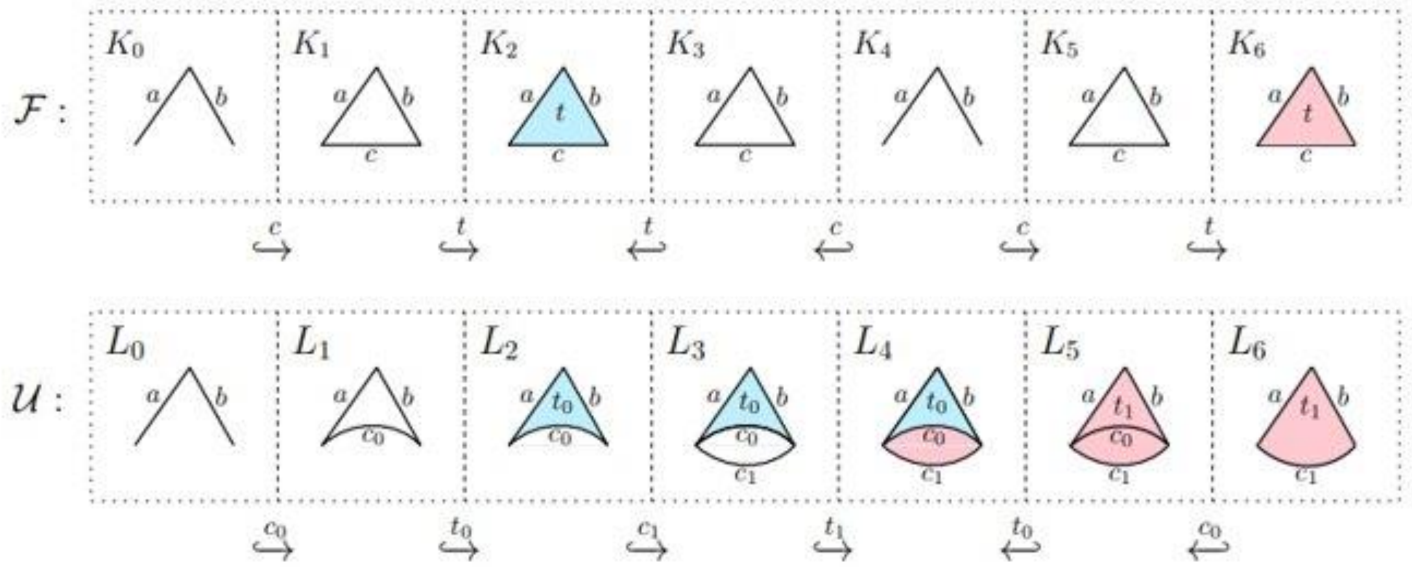
$\mathsf{Pers}_*(\mathcal{F})$		$\mathsf{Pers}_*(\mathcal{F}')$
$[b, j-1]$; $b \leq j-1$	$\mapsto$	$[b, j]$
$[b, j]$; $b \leq j-1$	$\mapsto$	$[b, j-1]$
$[j, d]$; $d \geq j+1$	$\mapsto$	$[j+1, d]$
$[j+1, d]$; $d \geq j+1$	$\mapsto$	$[j, d]$
$[j, j]$ of dimension p	$\mapsto$	$[j, j]$ of dimension $p-1$
$[b, d]$; all other cases	$\mapsto$	$[b, d]$

Non-repetitive filtration

Step 1 is achieved by simply treating each repeatedly added simplex in $\mathcal{F}$ as a new cell in the converted filtration (see also [23]). Throughout the section, we denote the converted non-repetitive, cell-wise filtration as

$$\hat{\mathcal{F}} : \varnothing = \hat{K}_0 \overset{\hat{\sigma}_0}{\longleftrightarrow} \hat{K}_1 \overset{\hat{\sigma}_1}{\longleftrightarrow} \cdots \overset{\hat{\sigma}_{m-1}}{\longleftrightarrow} \hat{K}_m = \varnothing.$$

Notice that since each $\hat{K}_i$ in $\hat{\mathcal{F}}$ is homeomorphic to K_i in $\mathcal{F}$, the two filtrations $\mathcal{F}$ and $\hat{\mathcal{F}}$ are essentially the same (with different numbering of the simplices/cells). Hence, $\mathsf{Pers}_*(\mathcal{F}) = \mathsf{Pers}_*(\hat{\mathcal{F}})$.



Up-down filtration

$$\hat{\mathcal{F}} : \varnothing = \hat{K}_0 \xleftrightarrow{\hat{\sigma}_0} \hat{K}_1 \xleftrightarrow{\hat{\sigma}_1} \cdots \xleftrightarrow{\hat{\sigma}_{m-1}} \hat{K}_m = \varnothing$$

$$\mathcal{U} : \varnothing = L_0 \hookrightarrow L_1 \hookrightarrow \cdots \hookrightarrow L_n \hookleftarrow L_{n+1} \hookleftarrow \cdots \hookleftarrow L_{2n} = \varnothing$$

Proof. Let $\hat{K}_i \xhookleftarrow{\hat{\sigma}_i} \hat{K}_{i+1}$ be the first deletion in $\hat{\mathcal{F}}$ and $\hat{K}_j \xhookrightarrow{\hat{\sigma}_j} \hat{K}_{j+1}$ be the first addition after that. That is, $\hat{\mathcal{F}}$ is of the form

$$\hat{\mathcal{F}} : \hat{K}_0 \hookrightarrow \cdots \hookrightarrow \hat{K}_i \xhookleftarrow{\hat{\sigma}_i} \hat{K}_{i+1} \xhookleftarrow{\hat{\sigma}_{i+1}} \cdots \xhookleftarrow{\hat{\sigma}_{j-2}} \hat{K}_{j-1} \xhookleftarrow{\hat{\sigma}_{j-1}} \hat{K}_j \xhookrightarrow{\hat{\sigma}_j} \hat{K}_{j+1} \leftrightarrow \cdots \leftrightarrow \hat{K}_m$$

$$\hat{K}_0 \hookrightarrow \cdots \hookrightarrow \hat{K}_i \xhookleftarrow{\hat{\sigma}_i} \hat{K}_{i+1} \xhookleftarrow{\hat{\sigma}_{i+1}} \cdots \xhookleftarrow{\hat{\sigma}_{j-2}} \hat{K}_{j-1} \xhookrightarrow{\hat{\sigma}_j} \hat{K}'_j \xhookleftarrow{\hat{\sigma}_{j-1}} \hat{K}_{j+1} \leftrightarrow \cdots \leftrightarrow \hat{K}_m$$

Mayer-Vietoris Diamond

Up-down filtration complexity

This looks like it has worst-time complexity O(m[2]). But there is a simpler, linear way to compute it

In a cell-wise filtration, for a cell σ, let its addition (insertion) be denoted as i:σ and its deletion (removal) be denoted as r:σ. From the proof of Proposition 9, we observe the following: during the transition from $\hat{\mathcal{F}}$ to $\mathcal{U}$, for any two additions i:σ and i:σ' in $\hat{\mathcal{F}}$ (and similarly for deletions), if i:σ is before i:σ' in $\hat{\mathcal{F}}$, then i:σ is also before i:σ' in $\mathcal{U}$. We then have the following fact:

Fact 10. *Given the filtration $\hat{\mathcal{F}}$, to derive $\mathcal{U}$, one only needs to scan $\hat{\mathcal{F}}$ and list all the additions first and then the deletions, following the order in $\hat{\mathcal{F}}$.*

Definition 12 (Creator and destroyer)**.** For any interval $[b,d] \in \mathsf{Pers}_*(\hat{\mathcal{F}})$, if $\hat{K}_{b-1} \xleftrightarrow{\hat{\sigma}_{b-1}} \hat{K}_b$ is forward (resp. backward), we call i:$\hat{\sigma}_{b-1}$ (resp. r:$\hat{\sigma}_{b-1}$) the *creator* of $[b,d]$. Similarly, if $\hat{K}_d \xleftrightarrow{\hat{\sigma}_d} \hat{K}_{d+1}$ is forward (resp. backward), we call i:$\hat{\sigma}_d$ (resp. r:$\hat{\sigma}_d$) the *destroyer* of $[b,d]$.

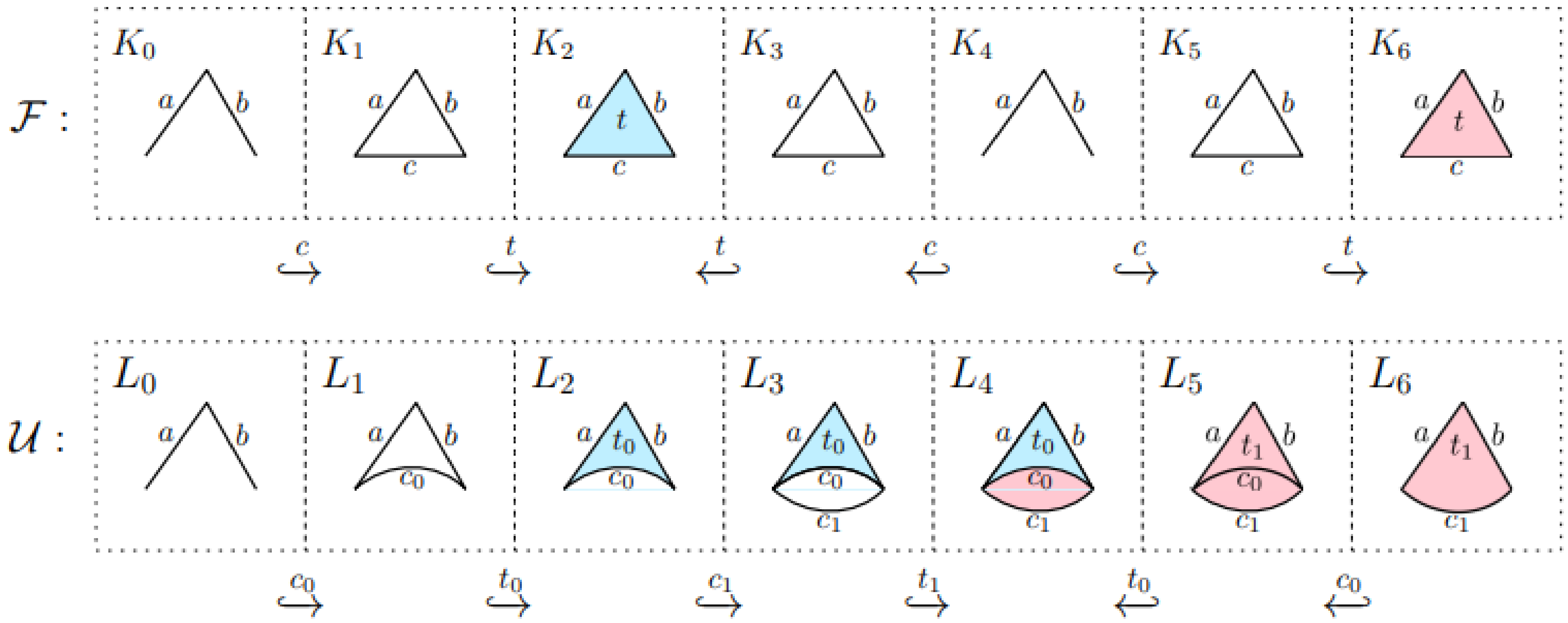
Proposition 13. *For two cell-wise filtrations $\mathcal{L}, \mathcal{L}'$ related by a Mayer-Vietoris diamond, any two intervals of $\mathsf{Pers}_*(\mathcal{L})$ and $\mathsf{Pers}_*(\mathcal{L}')$ mapped by the Diamond Principle have the same set of creator and destroyer, though the creator and destroyer may swap. This implies that there is a bijection from $\mathsf{Pers}_*(\mathcal{U})$ to $\mathsf{Pers}_*(\hat{\mathcal{F}})$ s.t. any two corresponding intervals have the same set of creator and destroyer.*

Up-down filtration complexity

Proposition 15. *There is a bijection from* $\mathsf{Pers}_*(\mathcal{U})$ *to* $\mathsf{Pers}_*(\hat{\mathcal{F}})$ *which maps each* $[b, d] \in \mathsf{Pers}_p(\mathcal{U})$ *by the following rule:*

Type	Condition		Type	Interval in $\mathsf{Pers}_*(\hat{\mathcal{F}})$	Dim
closed-open	-	$\mapsto$	closed-open	$\left[\mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{i}{:}\tau_{b-1}) + 1, \mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{i}{:}\tau_d)\right]$	p
open-closed	-	$\mapsto$	open-closed	$\left[\mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{r}{:}\tau_{b-1}) + 1, \mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{r}{:}\tau_d)\right]$	p
closed-closed	$\mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{i}{:}\tau_{b-1}) < \mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{r}{:}\tau_d)$	$\mapsto$	closed-closed	$\left[\mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{i}{:}\tau_{b-1}) + 1, \mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{r}{:}\tau_d)\right]$	p
	$\mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{i}{:}\tau_{b-1}) > \mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{r}{:}\tau_d)$	$\mapsto$	open-open	$\left[\mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{r}{:}\tau_d) + 1, \mathtt{id}_{\hat{\mathcal{F}}}(\mathsf{i}{:}\tau_{b-1})\right]$	$p-1$

Example



$\mathcal{F}$:
K_0
K_1
K_2
K_3
K_4
K_5
K_6
a
b
c
t
$\mathcal{U}$:
L_0
L_1
L_2
L_3
L_4
L_5
L_6
c_0
c_1
t_0
t_1

Non-zigzag filtration

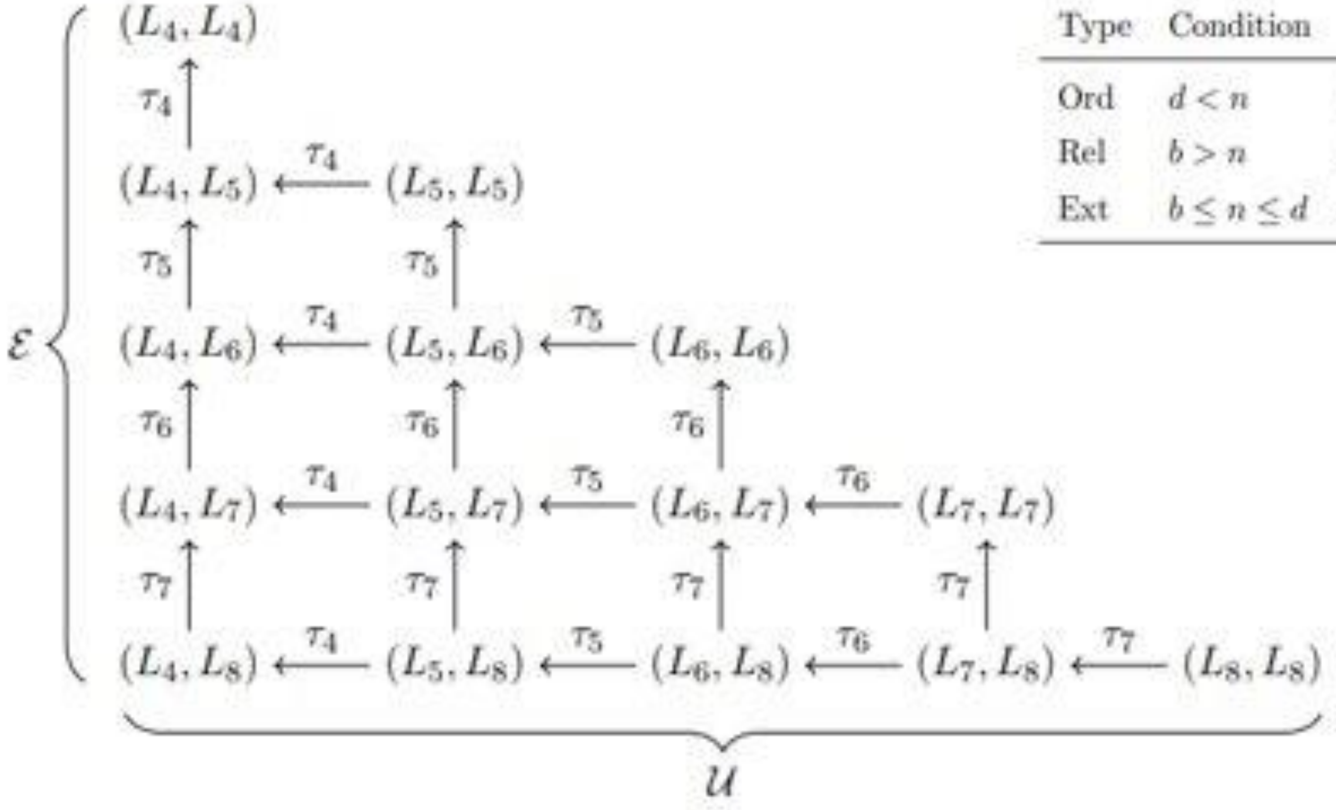
$$\mathcal{U} : \varnothing = L_0 \xhookrightarrow{\tau_0} \cdots \xhookrightarrow{\tau_{n-1}} L_n \xhookleftarrow{\tau_n} \cdots \xhookleftarrow{\tau_{2n-1}} L_{2n} = \varnothing$$

$$\mathcal{E} : \varnothing = L_0 \hookrightarrow \cdots \hookrightarrow L_n = (\hat{K}, L_{2n}) \hookrightarrow (\hat{K}, L_{2n-1}) \hookrightarrow \cdots \hookrightarrow (\hat{K}, L_n) = (\hat{K}, \hat{K})$$

$$\begin{array}{ccc} (L_n, L_{2n-1}) & \longleftarrow & (L_{n+1}, L_{2n-1}) \\ \uparrow & & \uparrow \\ (L_n, L_{2n}) & \longleftarrow & (L_{n+1}, L_{2n}) \end{array}$$

$$\begin{array}{ccccc} & \nearrow & (\mathbb{A}_1 \cup \mathbb{A}_2, \mathbb{B}_1 \cup \mathbb{B}_2) & \nwarrow & \\ (\mathbb{A}_1, \mathbb{B}_1) & & & & (\mathbb{A}_2, \mathbb{B}_2) \\ & \nwarrow & (\mathbb{A}_1 \cap \mathbb{A}_2, \mathbb{B}_1 \cap \mathbb{B}_2) & \nearrow & \end{array}$$

Non-zigzag filtration

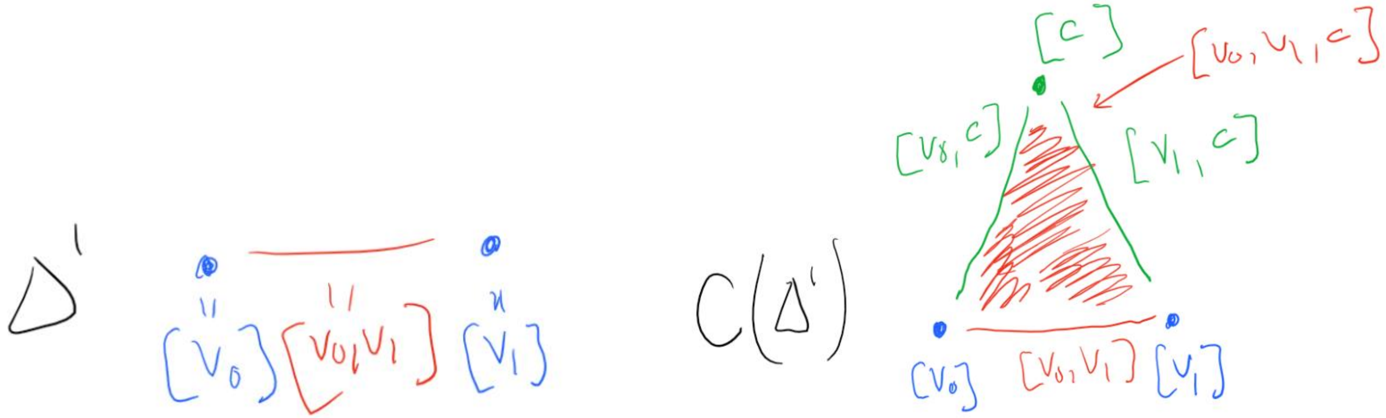


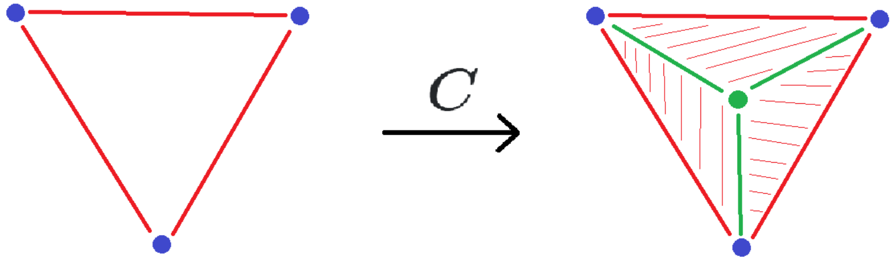
Proposition 18. *There is a bijection from* $\mathsf{Pers}_*(\mathcal{E})$ *to* $\mathsf{Pers}_*(\mathcal{U})$ *which maps each* $[b, d] \in \mathsf{Pers}_*(\mathcal{E})$ *of dimension* p *by the following rule:*

Type	Condition		Type	Interv. in $\mathsf{Pers}_*(\mathcal{U})$	Dim
Ord	$d < n$	$\mapsto$	closed-open	$[b, d]$	p
Rel	$b > n$	$\mapsto$	open-closed	$[3n - d, 3n - b]$	$p - 1$
Ext	$b \leq n \leq d$	$\mapsto$	closed-closed	$[b, 3n - d - 1]$	p

Coning

$$\{\sigma_i\}_i \xrightarrow{C} \bigcup_i \{\sigma_i, [c], \tilde{\sigma}_i\}$$





Coning

$$\tilde{H}_n(X/A) = H_n(X, A)$$

i.e. the cone

$$H_n(\hat{K}, L_{2n-1}) = \tilde{H}_n(\hat{K}/L_{2n-1}) = \tilde{H}_n(\hat{K} \cup \omega \cdot L_{2n-1})$$

$$\mathcal{E} : \varnothing = L_0 \hookrightarrow \cdots \hookrightarrow L_n = (\hat{K}, L_{2n}) \hookrightarrow (\hat{K}, L_{2n-1}) \hookrightarrow \cdots \hookrightarrow (\hat{K}, L_n) = (\hat{K}, \hat{K})$$

$$\hat{\mathcal{E}} : L_0 \cup \{\omega\} \hookrightarrow \cdots \hookrightarrow L_n \cup \{\omega\} = \hat{K} \cup \omega \cdot L_{2n} \hookrightarrow \hat{K} \cup \omega \cdot L_{2n-1} \hookrightarrow \cdots \hookrightarrow \hat{K} \cup \omega \cdot L_n$$

Summary

$$\mathcal{F} : \varnothing = K_0 \xleftrightarrow{\sigma_0} K_1 \xleftrightarrow{\sigma_1} \cdots \xleftrightarrow{\sigma_{m-1}} K_m = \varnothing$$

$$\hat{\mathcal{F}} : \varnothing = \hat{K}_0 \xleftrightarrow{\hat{\sigma}_0} \hat{K}_1 \xleftrightarrow{\hat{\sigma}_1} \cdots \xleftrightarrow{\hat{\sigma}_{m-1}} \hat{K}_m = \varnothing$$

$$\mathcal{U} : \varnothing = L_0 \hookrightarrow L_1 \hookrightarrow \cdots \hookrightarrow L_n \hookleftarrow L_{n+1} \hookleftarrow \cdots \hookleftarrow L_{2n} = \varnothing$$

$$\mathcal{E} : \varnothing = L_0 \hookrightarrow \cdots \hookrightarrow L_n = (\hat{K}, L_{2n}) \hookrightarrow (\hat{K}, L_{2n-1}) \hookrightarrow \cdots \hookrightarrow (\hat{K}, L_n) = (\hat{K}, \hat{K})$$

$$\hat{\mathcal{E}} : L_0 \cup \{\omega\} \hookrightarrow \cdots \hookrightarrow L_n \cup \{\omega\} = \hat{K} \cup \omega \cdot L_{2n} \hookrightarrow \hat{K} \cup \omega \cdot L_{2n-1} \hookrightarrow \cdots \hookrightarrow \hat{K} \cup \omega \cdot L_n$$

Performance

Table 1: Running time of Dionysus2, Gudhi, and FZZ on different filtrations of similar lengths with various repetitiveness. All tests were run on a desktop with Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz, 16GB memory, and Linux OS.

No.	Length	Dim	Rep	T_{Dio2}	T_{Gudhi}	T_{FZZ}	Speedup
1	5,260,700	5	1.0	2h02m46.0s	–	8.9s	873
2	5,254,620	4	1.0	19m36.6s	–	11.0s	107
3	5,539,494	5	1.3	3h05m00.0s	45m47.0s	3m20.8s	13.7
4	5,660,248	4	2.0	2h59m57.0s	29m46.7s	4m59.5s	6.0
5	5,327,422	4	3.5	43m54.8s	10m35.2s	3m32.1s	3.0
6	5,309,918	3	5.1	5h46m03.0s	1h32m37.0s	19m30.2s	4.7
7	5,357,346	3	7.3	3h37m54.0s	57m28.4s	30m25.2s	1.9
8	6,058,860	4	9.1	53m21.2s	7m19.0s	3m44.4s	2.0
9	5,135,720	3	21.9	23.8s	15.6s	8.6s	1.9
10	5,110,976	3	27.7	36.2s	39.9s	8.5s	4.3
11	5,811,310	4	44.2	38.5s	36.9s	23.9s	1.5