# A FAST METHOD OF COMPUTING PERSISTENT HOMOLOGY OF TIME SERIES DATA

Shotaro Tsuji\*

\*Department of Methematical Informatics University of Tokyo Tokyo, Japan

# ABSTRACT

This study proposes a method that speeds up computing persistent homology of time series data. Persistent homology is recently used for clutering time series data and detecting periodicity of them. The proposed method uses line segments to approximate a trajectory in delay-coordinate space. Cubic Bézier curves are fitted to given data and divided into line segments. The distance between line segments is defined and calculated to construct the Vietoris-Rips complex of segments. Exploiting the Vietoris-Rips complex enables us to use fast software like Ripser. In experiments, the performance of the proposed method is compared with that of the ordinary method. The proposed method was 30 times or more faster than the ordinary method. It also smooths noisy data and produces more precise persistent homology.

*Index Terms*— Topological signal analysis, persistent homology, delay-coordinate embedding, Vietoris-Rips complex

# 1. INTRODUCTION

The combination of persistent homology [1, 2] and attractor reconstruction [3, 4] is applied for detecting periodicity from time series data [5, 6] and clustering time series data [7, 8, 9, 10, 11, 12]. Attractor reconstruction is a method that recovers the shape of an attractor from observed time series. Persistent homology analyzes the shape of a reconstructed attractor focusing on holes and the summarized data is used as the feature for machine learning techniques. This combination has difficulty; it requires long computational time and huge memory. For example, it takes several days and comsumes about a half of terabytes of memory for thousands of points.

We propose a method that reduces the computational time and memory by fitting cubic Bézier curves to time series and dividing them into segments; the time series is approximated by line segments. The Vietoris-Rips complex is often used to compute persistent homology of a point cloud. Because it is defined for points in metric space, the Vietoris-Rips complex of line segments can be constructed by defining the distance between segments. Defining and computing the distance between segments are straightforward because they are convex sets and can be parametrized.

In this study, the proposed method is applied to two kinds of data: an irrational flow on a 2-torus and the signals of Japanese vowels. The experiments show that the proposed method is tens or more faster than the ordinary method and it comsumes tens or more less memory. In addition, the proposed method smooths noisy signals and computes the persistent homology nearer to the original signals.

Kazuyuki Aihara\*†

<sup>†</sup>Institute of Industrial Science University of Tokyo Tokyo, Japan

These results make the combination of persistent homology and attractor reconstruction more practical.

There is the Witness complex [13], which reduces the size of simplicial complexes for generic point clouds. The Witness complex chooses landmark points and groups the nearest neighbor points of each landmark point. The proposed method is compared to the Witness complex. In the last experiment, the Witness complex recovered the 1st persistent homology less precisely than our method. Although the Witness complex is not limited for continuous curves, our experiment suggests that our specialized method for curves produces better results. Our method has a merit in this point.

# 2. BACKGROUND

## 2.1. Persistent Homology

Persistent homology [1, 2] is an algebraic tool for counting the number and the widths of holes of a given shape. In mathematics, homology groups count arbitrary dimensional holes in a space. The given space is a topological space or a simplicial complex that approximates the topological space. In real world, however, we cannot obtain a whole topological space or a simplicial complex. Usually, we can only obtain a set of points. Therefore the method to construct a simplicial complex from a set of points is required.

**Vietoris-Rips complex.** The Vietoris-Rips complex [2] is a simplicial complex defined with a set of points A and a threshold r. The Vietoris-Rips complex of A with threshold r is defined by

$$\begin{aligned} &\operatorname{VR}_{r}(A) \\ &= \left\{ |x_{i_{0}}x_{i_{1}}\cdots x_{i_{q}}| : q \text{-simplex} \mid \forall j, k, d(x_{i_{j}}, x_{i_{k}}) \leq r) \right\}, \end{aligned}$$

where  $d(\cdot, \cdot)$  is a distance function.

**Filtration.** Although we got be able to construct a simplicial complex from a set of points, we must choose the threshold of Vietoris-Rips complex. We can avoid determining the threshold by using the filtration of simplicial complex. For a simplicial complex  $K = \{\sigma_i\}_{i=1}^m$ , a filtration of K is a sorted complex  $K = \{\sigma_{\iota(i)}\}_{i=1}^m$ , where  $\iota$  is a total order and each subset  $K^{(l)} = \{\sigma_{\iota(i)}\}_{i=1}^l$  is a simplicial complex and a subcomplex of K. The simplices in a Vietoris-Rips complex  $\operatorname{VR}_r(A)$  are sorted by the value of the filter function f in ascending order. The filter function f is defined as follows:

$$f(\sigma) = \begin{cases} 0, & \text{if } \dim \sigma = 0, \\ d(u, v), & \text{if } \dim \sigma = 1, \\ \max_{\tau \prec \sigma} f(\tau), & \text{otherwise,} \end{cases}$$
(2)

where u and v are the vertexes of 1-simplex  $\sigma$ , and  $\tau \prec \sigma$  means that  $\tau$  is a face of  $\sigma$ .

This research was partially supported by AMED under Grant Number JP18dm0307009 and NEC Corporation.

Practically the threshold of the Vietoris-Rips complex is set to the infinity. We estimate the number of simplexes in  $VR_{\infty}(A)$  to consider the performance of computing persistent homology. Let nbe the number of points in the set A. The complex  $VR_{\infty}(A)$  consists of an n-simplex and its faces because all of vertexes are connected each other. The upper dimension of simplexes is restricted to some value in practice. When the upper dimension is set to q-1, the complex  $VR_{\infty}(A)$  consists of k-simplexes where k is an integer varying from 0 to q. The number of k-simplexes in  $VR_{\infty}(A)$  is  $\binom{n}{k}$ . Thus the number of simplexes in  $VR_{\infty}(A)$  with upper dimension restriction is given by  $\sum_{k=0}^{q} \binom{n}{k}$  and its order is  $O(n^{q})$ . **Persistent homology.** Persistent homology is a variation of homol-

**Persistent homology.** Persistent homology is a variation of homology group and is defined for filtered simplicial complexes. Suppose that a filtered simplicial complex is given:  $\mathcal{K} = \{\sigma_i\}_{i=1}^m$ . We define the boundary matrix of  $\mathcal{K}$  by

$$D_{ij} = \begin{cases} 1, & \text{if } \dim \sigma_i = \dim \sigma_j - 1 \text{ and } \sigma_i \prec \sigma_j, \\ 0, & \text{otherwise.} \end{cases}$$
(3)

The boundary matrix D is a matrix in  $\mathbb{Z}_2^{m \times m}$ . A variation of column or row reduction reveals the image and kernel of the boundary matrix [15, 16]. By pairing the columns of the reduced matrix, we can obtain the generators of persistent homology group. A generator of persistent homology corresponds to a hole in a given shape. A generator is born at some step in the filtration and it will die at other step or will not die.

**Persistence diagram.** Persistence diagram [2] is a way to visualize a persistent homology group. We denote the birth filtration value of a generator  $\alpha$  of q-th persistent homology  $PH_q$  by  $b(\alpha)$  and the death filtration value of  $\alpha$  by  $d(\alpha)$ . The persistence diagram of a persistent homology  $PH_q(A)$  is defined as follows:

$$\operatorname{Dgm}_{q}(A) = \{ (b(\alpha), d(\alpha)) \mid \alpha \in PH_{q}(A) \} \cup \{ (x, x) \mid x \in \mathbb{R} \cup \{\infty\} \}.$$
(4)

The first term of the right-hand side of Eq.(4) is a multiset because the pair of the birth and the death filtration values may be multiple.

There is a distance function between persistence diagrams. The bottleneck distance [17] is defined for two persistence diagrams X and Y:

$$V_{\infty}(X,Y) = \inf_{\eta: X \to Y} \sup_{\alpha \in X} \|\alpha - \eta(\alpha)\|_{\infty},$$
 (5)

where  $\eta$  ranges over all bijections from X to Y and the norm is defined as  $\|\alpha\|_{\infty} = \|(b,d)\|_{\infty} = \max\{|b|, |d|\}.$ 

#### 2.2. Attractor Reconstruction

V

To analyze time series data, we assume that there are differential equations that govern it. We can usually acquire the time series data projected onto the space with the dimension smaller than that of the original phase space. The delay-coordinate embedding can recover the original trajectory from the observed time series [3, 4]. Takens' theorem guarantees that the recovered shape is diffeomorphic to the attractor in the original phase space under some conditions. For a time series  $\{x(t)\}_{t=1}^n$ , the delay-coordinate vector with delay *a* and dimension *d* is defined as

$$y(t) = (x(t), x(t-a), x(t-2a), \dots, x(t-(d-1)a)).$$
(6)

The series of the delay-coordinate vector  $\{y(t)\}_{t=1}^{n}$  is in *d*-dimensional Euclidean space.

We can combine persistent homology and attractor reconstruction. Because the series  $A = \{y(t)\}_{t=1}^{n}$  is in *d*-dimensional Euclidean space, a Vietoris-Rips complex can be constructed from it.



Fig. 1. Computational time of persistent homology of the 2-torus.



Fig. 2. Computational memory of persistent homology of the 2-torus.

This series consists of the sampled points of the recovered trajectory. Then the persistent homology of A can be computed. The persistence diagram of A can be used as a feature for machine learning techniques. Therefore the combination of persistent homology and attractor reconstruction enables us to classify time series data.

#### **3. PROPOSED METHOD**

Although the persistent homology of time series data can be used for clustering, the resources required to compute persistent homology are quite large. However, fortunately, the trajectory of differential equations is a continous curve and we can exploit this fact. First the given series is divided into groups of several tens of points. Second a cubic Bézier curve is fitted to the points of each group. Third the fitted Bézier curves are divided into line segments. Then the Vietoris-Rips complex of line segments is constructed and its persistent homology groups are computed.

#### 3.1. Fit a cubic Bézier curve

In this method, fitting a cubic Bézier curve is implemented with the least squares method. The series of points and their sampling time are given:  $\{(x_i, t_i) \mid x_i \in \mathbb{R}^d\}_{i=1}^l$ , where  $t_0, t_1, \ldots, t_l$  satisfy the inequalities  $0 \le t_0 < t_1 < \cdots < t_l \le 1$ . Let  $p_0, p_1, p_2$  and  $p_3$  be the control points of a cubic Bézier curve. The cubic Bézier curve is



**Fig. 3**. Bottleneck distance between the original data of 2-torus and the noised data of that. The left-side blank bars show the distances between the original and the ordinary Vietoris-Rips complex of the noised data. The right-side hatched bars show the distances between the original and the Vietoris-Rips complex of the segments fitted to the noised data.

parametrized as  $p(t) = \sum_{i=0}^{3} {3 \choose i} (1-t)^{3-i} t^i p_i$  for  $0 \le t \le 1$ . The squared error function is defined as

$$L(p_0, p_1, p_2, p_3) = \sum_{i=1}^{l} ||p(t_i) - x_i||^2,$$
(7)

where  $\|\cdot\|$  means that the Euclidean norm. The cubic Bézier curve is fitted by finding the control points that minimize the squared error function. Solving the equation grad L = 0 achieves the minimization. The equation grad L = 0 is expanded into the equations  $Ap^{(j)} = b^{(j)}$  (j = 1, 2, ..., d), where A is a 4 × 4 matrix,  $p^{(j)}$  and  $b^{(j)}$  are 4-dimensional vector and they are defined as follows:

$$A = \left( \begin{pmatrix} 3\\i-1 \end{pmatrix} \begin{pmatrix} 3\\j-1 \end{pmatrix} \sum_{k=0}^{l} (1-t_k)^{6-(i+j-2)} t_k^{i+j-2} \right)_{\substack{i=1,\dots,4\\j=1,\dots,4}},$$

$$p^{(j)} = \left( \begin{array}{c} p_0^{(j)} & p_1^{(j)} & p_2^{(j)} & p_2^{(j)} \end{array} \right)^{\top}, \quad (9)$$

and

$$b^{(j)} = \left( \binom{3}{i-1} \sum_{k=1}^{l} (1-t_k)^{3-(i-1)} t_k^{(i-1)} x_k^{(j)} \right)_{i=1,\dots,4}^{\top}, \quad (10)$$

where  $p_i^{(j)}$  and  $x_k^{(j)}$  mean that the *j*-th coordinate of  $p_i$  and  $x_k$  respectively. This linear equation is solved with LU decomposition.

## 3.2. Divide into line segments

The fitted cubic Bézier curve is divided into line segments. The number of line segments is set as a fraction of the number of points to which the curve is fitted. Let r be the number of line segments into which the curve is divided. First the unit interval [0, 1] is uniformly divided into r intervals; we obtain the intervals  $[t_0, t_1]$ ,  $[t_1, t_2], \ldots, [t_{r-1}, t_r]$ , where  $t_i = i/r$  for  $i = 0, 1, \ldots, r$ . The left and right boundary of each intervals are mapped into the fitted Bézier curve and the mapped boundaries are turned into the endpoints of line segment. In other words, we obtain the line segments  $p(t_0)p(t_1), p(t_1)p(t_2), \ldots, p(t_{r-1})p(t_r)$ .

#### 3.3. Distance between line segments

In order to construct the Vietoris-Rips complex of line segments, we must calculate the distance between them. The distance between line

segments is defined as a special case of the distance between sets. Let  $\overline{q_0q_1}$  and  $\overline{r_0r_1}$  be line segments. The distance between  $\overline{q_0q_1}$  and  $\overline{r_0r_1}$  is defined as

$$d(\overrightarrow{q_0q_1}, \overrightarrow{r_0r_1}) = \min_{q \in \overrightarrow{q_0q_1}, r \in \overrightarrow{r_0r_1}} d(q, r).$$
(11)

These line segments are parametrized as q(s) and r(t) respectively, where s and t are in the unit interval. The distance is calculated by minimizing the function  $f(s,t) = d(q(s), r(t))^2 = ||q(s) - r(t)||^2$ . Although the equation  $\operatorname{grad} f = 0$  is a linear equation, this equation cannot be solved in the case where the two line segments are parallel. Thus the function f is minimized by the gradient descent method.

# 4. EXPERIMENT

The proposed method was tested with a trajectory of an irrational flow on a 2-torus and Japanese vowels signal data. In the following experiments, the Ripser [14] was used for computing the persistent homology of Vietoris-Rips complex and the computer used for experiments has 2.40GHz Xeon E5-4640 and 1TiB of memory. The upper dimension of the persistent homology was set to 2.

#### 4.1. Irrational flow on the 2-torus

The irrational flow on the 2-torus is defined as

$$\frac{du}{dt} = \alpha \mod 1, \quad \frac{dv}{dt} = \beta \mod 1,$$
 (12)

where  $(u, v) \in [0, 1] \times [0, 1]$  and the ratio  $\alpha/\beta$  is irrational. It is known that the solution of Eq. (12) covers  $[0, 1] \times [0, 1]$  densely after sufficiently long time. In this experiment, we set  $\alpha = 1$  and  $\beta = \sqrt{2}$ . Then the trajectory of the irrational flow is mapped into 3-dimensional Euclidean space. The mapping is defined as

$$\begin{cases} x_1 = R \cos u + r \cos u \cos v, \\ x_2 = R \sin u + r \sin u \cos v, \\ x_3 = r \sin v, \end{cases}$$
(13)

where R and r are positive real values and they satisfy R > r. We set R = 2 and r = 1.

The trajectory was developed from t = 0 to  $t = 50\pi$  and it was sampled in n = 2000, 3000 and 4000 points. Then cubic Bézier curves were fitted to each series of sampled points; the number of points in each group was about 30. Finally each fitted Bézier curve was divided into line segments. The number of line segments was set to r = 3, 6 and 10.

The first experiment was performance comparison. The performance of computing the persistent homology of the ordinary Vietoris-Rips complex was compared with that of computing that of the Vietoris-Rips complex of segments.

Figure 1 shows the plot of computational time versus the number of points of the trajectory. Figure 2 shows the plot of computational memory versus the number of points of the trajectory. The order of time and space complexity is about  $O(n^{3.25})$ . Even the case of r = 10 requires about 45 times shorter and 30 times smaller than the case of the ordinary Vietoris-Rips complex. Because we set  $l \simeq$ 30 and r = 10 in this case, the number of points that constuct the Vietoris-Rips complex was reduced to about 10/30 = 1/3. It is natural that the computational time and memory were reduced to about  $(1/3)^{3.25} \simeq 35$ . The cases of r = 3 and r = 6 require shorter time and smaller memory similarly. The ordinary Vietoris-Rips complex required about 630 GiB of memory for 4000 points. It is too large to use persistent homology in practical use. However the proposed method with r = 3 and that with r = 6 required less than 10 GiB of memory and less than 100 seconds of time. The proposed method enables us to use persistent homology practically.

Second the proposed method was applied to noisy data. The noisy data was generated by adding 10% gaussian noise of the root mean squares of the trajectory of the irrational flow. The number of the points in each group was set to  $l \simeq 30$ . Each fitted Bézier curve was divided into r = 6 segments.

We calculated the bottleneck distance between the persistent homology of the ordinary Vietoris-Rips complex of the original data and that of the Vietoris-Rips complex of the segments fitted to the noisy data. We also calculated the bottleneck distance between that of the original data and that of the noisy data. To simplify the description, let  $T_n$  be the original torus, where n is the number of sampled points. Let  $N_n$  be the noisy torus and  $S_n$  be the segments obtained by fitting. Paraphrasing the first two sentences, we calculated  $W_{\infty}(\mathrm{Dgm}_q(T_n), \mathrm{Dgm}_q(S_n))$  and  $W_{\infty}(\mathrm{Dgm}_q(T_n), \mathrm{Dgm}_q(N_n))$ for q = 0, 1, 2 and n = 2000, 3000, 4000. Let  $d_{\text{segment}}(q, n)$  be the former distance and  $d_{point}(q, n)$  the latter. In Fig.3 the distances  $d_{\text{point}}(q, n)$  were compared with the distances  $d_{\text{segment}}(q, n)$ . The graphs in Fig.3 show the distances of q = 0, q = 1 and q = 2 from left to right. The blank bars show the distances  $d_{point}(q, n)$  and the hatched bars show the distances  $d_{\text{segment}}(q, n)$ . For every value of q and n, the distance  $d_{\text{segment}}(q, n)$  is less than  $d_{\text{point}}(q, n)$ . This fact means that the proposed method has smoothing effect and it can compute more precise persistent homology under noisy observation.

# 4.2. Japanese vowels

The proposed method was also applied to Japanese vowels. The data was obtained from the Vowel Database: Five Japanese Vowels of Males, Females and Children Along With Relevant Physical Data

 Table 1. Performance comparison of computing the persistent homology of speech signals.

	Computational Time	Computational Memory		
Vowel	Ordinary Proposed Witness	Ordinary Proposed Witness		
/a/	290  sec  < 1  sec  < 1  sec	12.5 GiB 0.11 GiB 0.14 GiB		
/e/	$312 \sec < 1 \sec < 1 \sec$	11.5 GiB 0.09 GiB 0.09 GiB		
/i/	268  sec < 1  sec < 1  sec	11.5 GiB 0.09 GiB 0.10 GiB		
/o/	291 sec $< 1 \text{ sec} < 1 \text{ sec}$	11.5 GiB 0.09 GiB 0.09 GiB		
/u/	277 sec $< 1 \text{ sec} < 1 \text{ sec}$	11.5 GiB 0.09 GiB 0.09 GiB		

(JVPD) [18]. The choosed data was the utterance of 27-year-old female speaker. The utterance was recorded in 44.1 kHz, 16 bits PCM format and contains the signals of five Japanese vowels /a/, /e/, /i/, /o/ and /u/. The signals were rescaled to range from -1 to 1. The signals of these vowels were embedded into the delay-coordinate space of 10 dimension with delay of 10 steps. Then the 1100 steps, almost equal to 125 milliseconds, of each embedded signal were extracted. Each embedded signal was divided into the groups of  $l \simeq 10$  points and a cubic Bézier curve was fitted to each group of points, and then each Bézier curve was divided into r = 2 segments. Therefore we get about 220 segments. The Witness complexes [13] of the vowels were also constructed for comparison in the manner explained in Section 2.4 of [13] and their persistent homology was analyzed with the Ripser. The number of the landmarks was 220 and they were chosen by the maxmin selection.

Table 1 shows the comparison between the ordinary method and the proposed method applied to compute the persistent homology of the speech signals. It also shows the comparison with the Witness complex. The proposed method and the Witness complex were about 300 times faster than the ordinary method and took 100 times smaller memory than it. Table 2 shows the bottleneck distance between the persistent homology computed with the ordinary method and that with the proposed method. It also shows the comparison with the Witness complex. The distances of the proposed method took small values except the 1st persistent homology of the vowel /a/. The maximum of filtration value is  $\sqrt{10} \simeq 3.16$  because the space is the ten times product of [-1, 1]. These distances took the value of 0.131 at most. This value is about 4% of the maximum filtration value. However, the distance of the 1st persistent homology of the vowel /a/ was 0.644; this value is about 20% of the maximum filtration value. The reason of this result may be the original data was noisy and the noise was smoothed out but it cannot be proven. In contrast the distances of 1st persistent homology of the Witness complex were much larger than the proposed method.

# 5. CONCLUSION

We proposed a method that makes computing the persistent homology of time series data faster and smooths the observation noise out. This means that the proposed method enables practical use of the combination of persistent homology and attractor reconstruction. Moreover, the proposed method brings noise robustness to it. There is a issue whether the classification performance will be improved with our method. More experiments are necessary to show the advantages.

**Table 2**. The comparison of the Bottleneck distances between the ordinary method and the proposed method (left) and those between the ordinary method and the Witness complex (right).

	Proposed Method			Witness Complex		
Vowel	Oth	1st	2nd	Oth	1st	2nd
/a/	0.064	0.644	0.131	0.126	0.645	0.177
/e/	0.029	0.032	0.047	0.026	0.217	0.038
/i/	0.026	0.027	0.014	0.018	0.292	0.034
/0/	0.021	0.068	0.011	0.022	0.279	0.069
/u/	0.016	0.032	0.020	0.015	0.332	0.029

## 6. REFERENCES

- Herbert Edelsbrunner, David Letscher, and Afra Zomorodian, "Topological persistence and simplification," *Discrete Comput Geom*, vol. 28, pp. 511–533, 2002.
- [2] Herbert Edelsbrunner and John L. Harer, Computational Topology: An Introduction, American Methematical Society, 2010.
- [3] Floris Takens, "Detecting strange attractors in turbulence," *Lecture Notes in Mathematics, Berlin Springer Verlag*, vol. 898, pp. 366, 1981.
- [4] Tim Sauer, James A Yorke, and Martin Casdagli, "Embedology," *Journal of statistical Physics*, vol. 65, no. 3-4, pp. 579– 616, 1991.
- [5] Saba Emrani, Harish Chintakunta, and Hamid Krim, "Real time detection of harmonic structure: A case for topological signal analysis," in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014, pp. 3445–3449.
- [6] Jose A. Perea and John Harer, "Sliding windows and persistence: An application of topological methods to signal analysis," *Found. Comput. Math.*, vol. 15, no. 3, pp. 799–838, June 2015.
- [7] Cássio M.M. Pereira and Rodrigo F. de Mello, "Persistent homology for time series and spatial data clustering," *Expert Systems with Applications*, vol. 42, no. 15, pp. 6026 – 6038, 2015.
- [8] Joshua Garland, Elizabeth Bradley, and James D. Meiss, "Exploring the topology of dynamical reconstructions," *Physica D: Nonlinear Phenomena*, vol. 334, pp. 49 59, 2016, Topology in Dynamics, Differential Equations, and Data.
- [9] Slobodan Maletić, Yi Zhao, and Milan Rajković, "Persistent topological features of dynamical systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, no. 5, pp. 053105, 2016.
- [10] Lee M. Seversky, Shelby Davis, and Matthew Berger, "On time-series topological data analysis: New data and opportunities," in 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2016, pp. 1014– 1022.
- [11] Viney Venkataraman, Karthikeyan N. Ramamurthy, and Pavan Turaga, "Persistent homology of attractors for action recognition," in 2016 IEEE International Conference on Image Processing (ICIP), Sept 2016, pp. 4150–4154.
- [12] Yuhei Umeda, "Time series classification via topological data analysis," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 32, no. 3, pp. D G72 1–12, 2017.
- [13] Vin De Silva and Gunnar Carlsson, "Topological estimation using witness complexes," in *Proceedings of the First Eurographics Conference on Point-Based Graphics*, Aire-la-Ville, Switzerland, Switzerland, 2004, SPBG'04, pp. 157–166, Eurographics Association.
- [14] Ulrich Bauer, "Ripser," https://github.com/ Ripser/ripser, 2016.
- [15] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov, "Vines and vineyards by updating persistence in linear time," in *Proceedings of the twenty-second annual symposium* on Computational geometry. ACM, 2006, pp. 119–126.

- [16] Ulrich Bauer, Michael Kerber, and Jan Reininghaus, "Clear and compress: Computing persistent homology in chunks," in *Topological methods in data analysis and visualization III*, pp. 103–117. Springer, 2014.
- [17] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov, "Geometry helps to compare persistence diagrams," J. Exp. Algorithmics, vol. 22, pp. 1.4:1–1.4:20, Sept. 2017.
- [18] Gen Ohyama, Toshisada Deguchi, and Hideki Kasuya, "Construction of japanese vowel database uttered by native speakers over a wide range of age," 2011, vol. 2-P-15(a).