Audio identification using persistent homology

Wojciech Reise

École Polytéchnique Fédérale de Lausanne,
Spotify *wojciech.reise@alumni.epfl.ch*

4th December 2019

Acknowledgements

- Prof. Kathryn Hess-Bellwald[1],
- Prof. Heather Harrington[2],
- Dr. Mariano Beguerisse-Diaz[2,3],
- Maria Dominguez[3],
- Prof. Ulrike Tillmann[2],
- Dr. Martin Gould[3],
- Laurence Pascal[3].

[1] École Polytéchnique Fédérale de Lausanne,
[2] University of Oxford,
[3] Spotify.

Motivation for audio identification

Improve user experience

Identify the name of the soundtrack from a video or a noisy recording (eg. Shazam).

Prevent copyright infringements

Detect when a track is uploaded, possible with different metadata (f.ex. 'vponline' uploads Eric Claptons' Layla on Soundcloud).

Cluster tracks

Group audio by content (f.ex. covers), or type (music, nature sounds).

Overview

Background on audio identification

Persistent homology on images

Audio identification with persistent homology features

Experiments

Table of Contents

Problem statement

Let $\mathcal{S} \subset \mathbb{R}^N$ be a set of songs, and $\mathcal{T}$ a set of obfuscations.

Find $\phi : \mathbb{R}^N \times \mathbb{R}^N \to \{0, 1\}$, such that

- $\phi(T(s), T'(s)) = 1$ for all songs $s \in \mathcal{S}$ and obfuscations $T, T' \in \mathcal{T}$
- $\phi(s', s) = 0$ otherwise.

In this project, we restrict the problem to $T' = Id$.

Identification a two step method

Fingerprinting

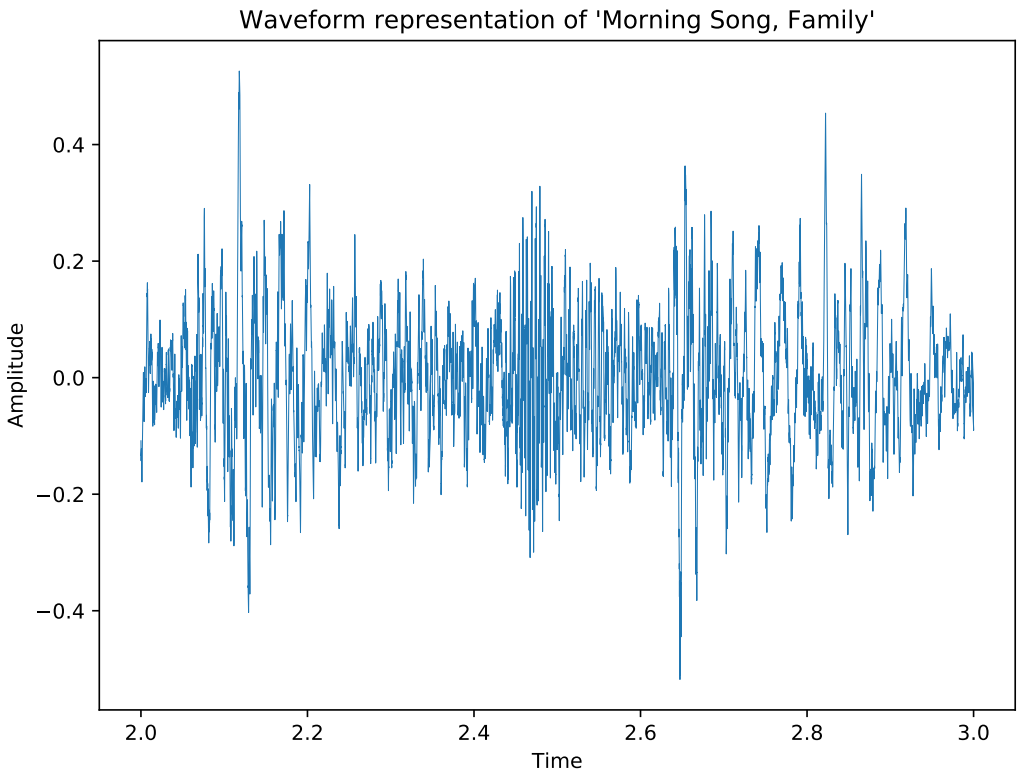
- Robust, and discriminative features, called fingerprints.
- Image analysis techniques on spectral representations of signals.

Querying

- Count the number / percentage of identical fingerprints [YHS05],
- 'Align' a snippet in the other using fingerprints [Wan03].

Audio signal

An audio signal is a collection of samples $s = (s_i)_{i=1}^{N}$, where $N = Tf_s$ and f_s the sampling rate.

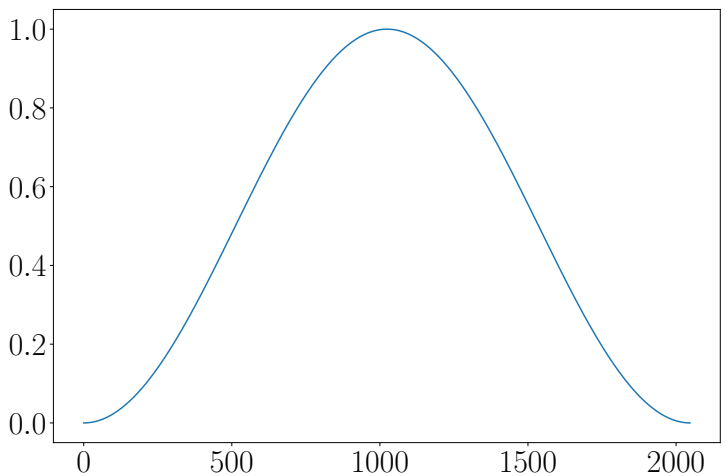


Spectral representations of songs

The signal is often represented in the time-frequency domain, using the discrete short-time Fourier transform (STFT) [Jul11]

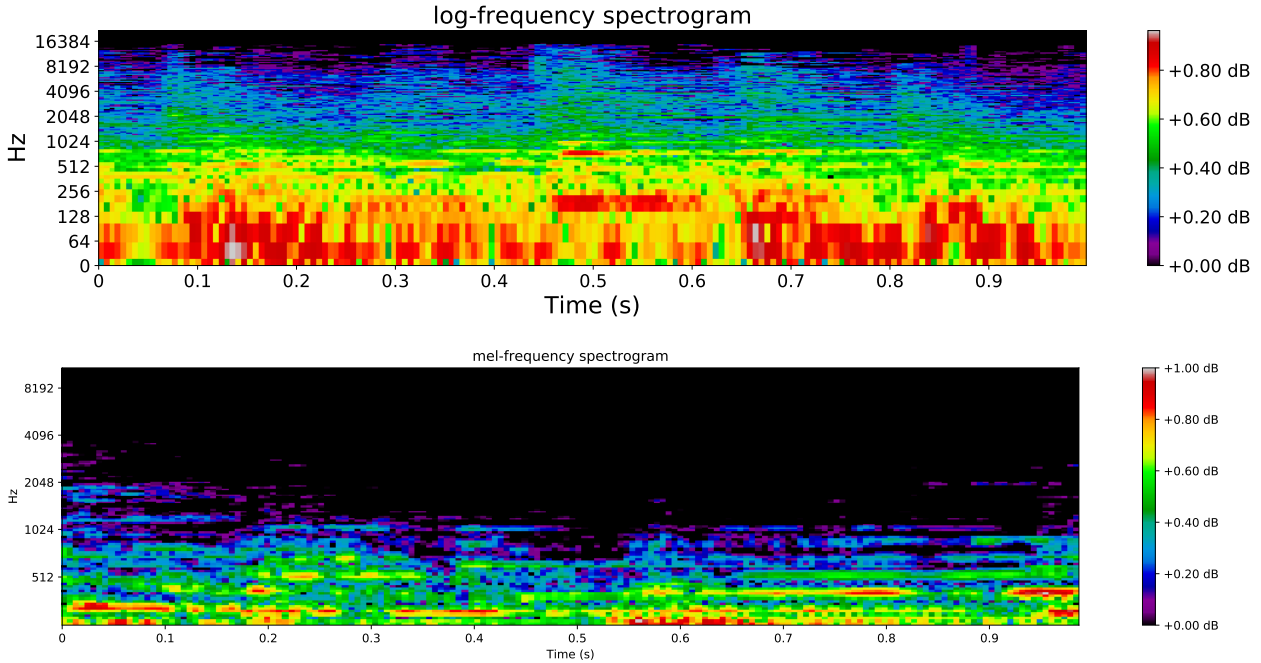
$$\hat{\mathcal{S}}(n, m) = \sum_{k=-\infty}^{\infty} s_k \omega_{k-hn} \exp\left(-\imath k \frac{m f_s}{N_\omega}\right), \tag{1}$$

where h is the hop size, $(\omega_k)_{k=0}^{N_\omega - 1}$ is a 'Hann' window of length N_ω.



Spectrograms - examples

The STFT operator is linear. We change the frequency scale to the mel scale, apply a stable log and an equal loudness filter.



Audio identification on obfuscated tracks

Benchmark

An Industrial-Strength Audio Search Algorithm

Avery Li-Chun Wang
avery@shazamteam.com
Shazam Entertainment, Ltd.

USA:
2925 Ross Road
Palo Alto, CA 94303

United Kingdom:
375 Kensington High Street
4th Floor Block F
London W14 8Q

Fingerprinting

Detect salient points P in S.
Select a subset of pairs $\tilde{P} \subset P \times P$.
Initialize the fingerprint dictionary Φ.
for pairs $(p_1, p_2) \in \tilde{P}$ **do**
    Set $k = (f_1, f_2, t_2 - t_1)$, $p_i = (t_i, f_i)$.
    Add k with value t_1 to Φ.
end for
return Φ

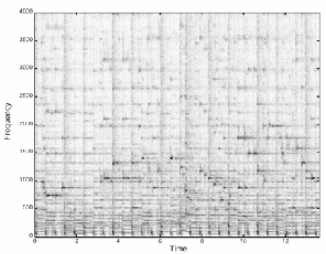


Fig. 1A - Spectrogram

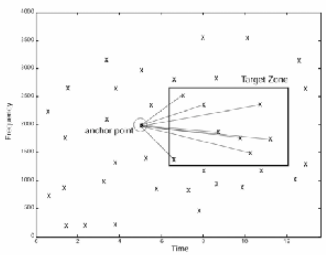


Fig. 1C - Combinatorial Hash Generation

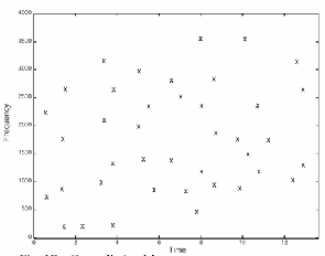


Fig. 1B - Constellation Map

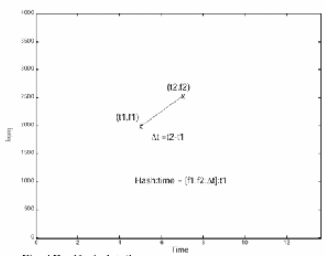


Fig. 1D - Hash details

Audio identification on obfuscated tracks

Benchmark

An Industrial-Strength Audio Search Algorithm

Avery Li-Chun Wang
avery@shazamteam.com
Shazam Entertainment, Ltd.

USA:
2925 Ross Road
Palo Alto, CA 94303

United Kingdom:
375 Kensington High Street
4th Floor Block F
London W14 8Q

Querying

Consider the fingerprints Φ and Φ' of s and s'.
Initialize the alignment list A
for common keys $k \in \Phi \cap \Phi'$ **do**
$t = \Phi(k)$, $t' = \Phi'(k)$.
$A = A \cup \{(t, t')\}$
end for
$q =$ nb of occurrences of the mode of
$\{t - t' \mid (t, t') \in A\}$
return True, if $q > T$, False otherwise.

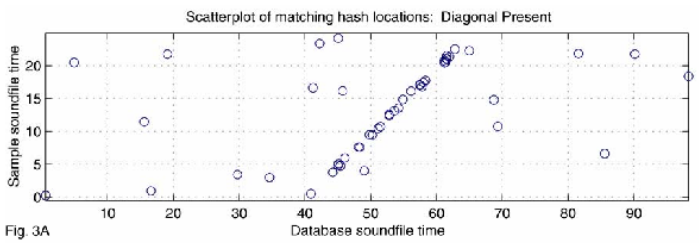


Fig. 3A

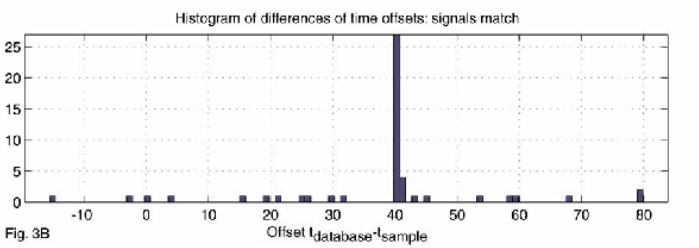


Fig. 3B

Audio identification on obfuscated tracks

The approaches presented in [YHS05, Wan03, BC07] share a common structure:

1. fingerprinting,
2. querying.

The fingerprints should be

- discriminative,
- stable,
- easy to compare.

Table of Contents

Motivation

Obfuscations as tansformations of images

Proximity of coordinates matters

Motivation

Obfuscations as tansformations of images

Obfuscations correspond to operations on spectrograms, like stretching, shifting and adding rows/columns.

Obfuscation	Operation
Pitch-shifting	Vertical translation
Tempo-shift	Horizontal stretching
High-/Low-pass filter	Attenuation of values in rows below/above a threshold
Noise (pink or white)	Adding noise with a specific spectrum to the image
Leading/trailing silence	Adding columns with zeros to the left /right.

Proximity of coordinates matters

Motivation

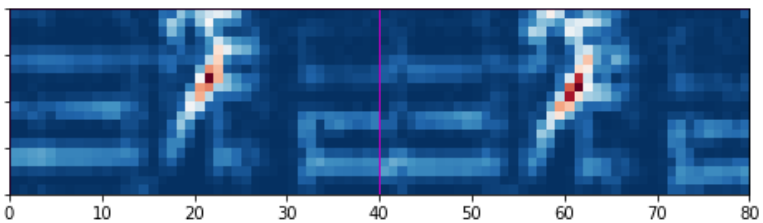
Obfuscations as tansformations of images

Proximity of coordinates matters

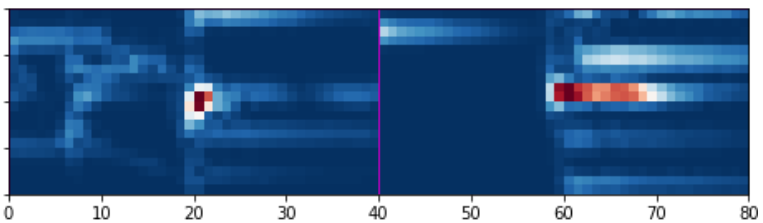
In a spectrogram, there is a notion of proximity between coordinates as neighbouring pixels share information.

- Columns $S_{n,:}$ and $S_{n+1,:}$ encode spectral decompositions of overlapping signals.
- The frequency responses of filters overlap, so that $S_{:,m}$ and $S_{:,m+1}$ are related.

Examples - positive and negative

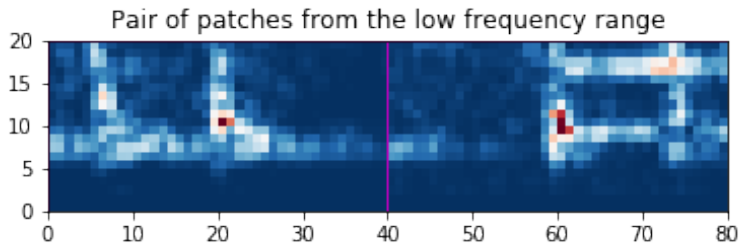


'Matching' pair

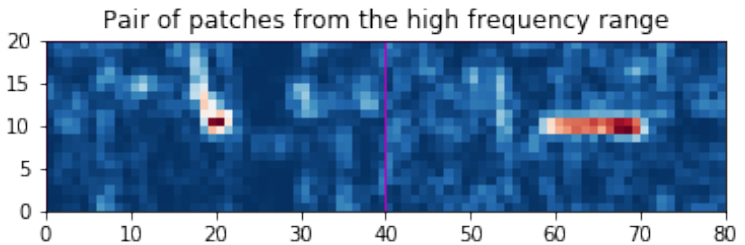


Random pair

Examples - by frequency region

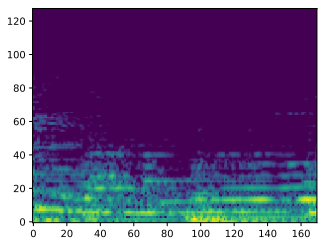


‘Low’

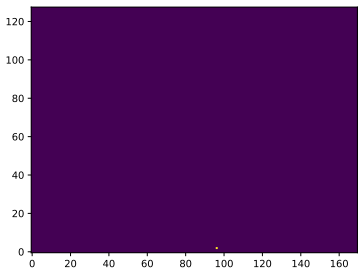


‘High’

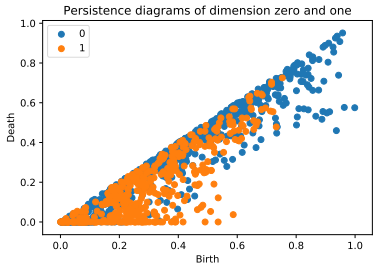
Fingerprinting an image



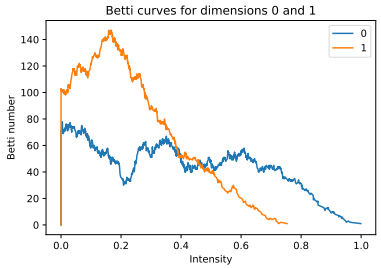
Image



Filtration



Persistence diagrams



Betti curves

Cubical complex

Definition (following [KMM11])

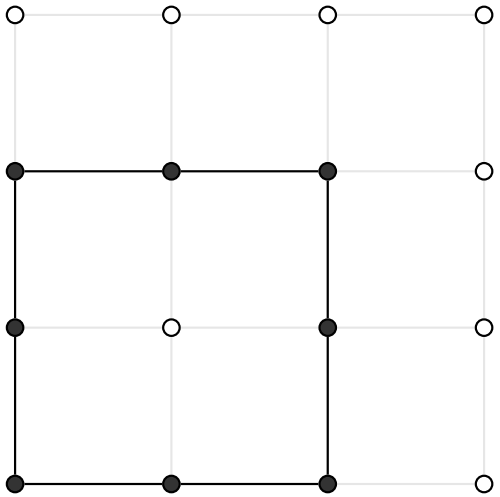
A closed interval $I = [a, b] \subset \mathbb{R}$ is called elementary if $b \in \{a, a+1\} \subset \mathbb{Z}$. In the case $a = b$, the interval is called degenerate.
An elementary cube $Q = I_1 \times \ldots \times I_d \subset \mathbb{R}^d$ is a product of d elementary intervals.

Example

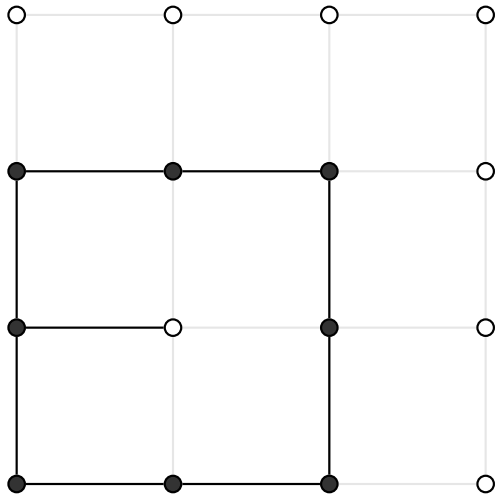
A vertex (0-cube) is a product of degenerate intervals. An edge (1-cube) is a product of intervals where all but one are degenerate.

Cubical complex - example

Example



A cubical complex



Not a cubical complex.

Homology groups of cubical complexes

Definition

We call k-chains $C_k(K)$ the vector space on $\mathbb{F}$, generated by elementary cubes of dimension k, namely

$$C_k(K) = \left\{ \sum_{Q \ k-\text{cube}} a_Q \widehat{Q} \mid a_Q \in \mathbb{F}_2 \right\}.$$

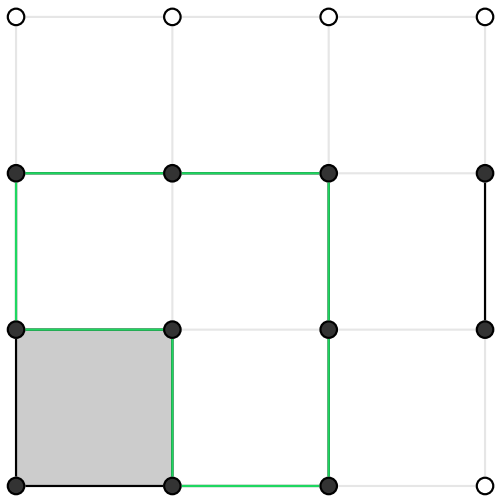
Homology groups H_k are defined as

$$H_k(K) = \ker(\partial_k)/\mathrm{im}(\partial_{k+1}),$$

where $\partial_k : C_k(K) \to C_{k-1}(K)$ is the boundary operator on cubes.

Homology groups of cubical complexes

Example



$$H_0(K) = \mathbb{F}_2^2,$$
$$H_1(K) = \mathbb{F}_2.$$

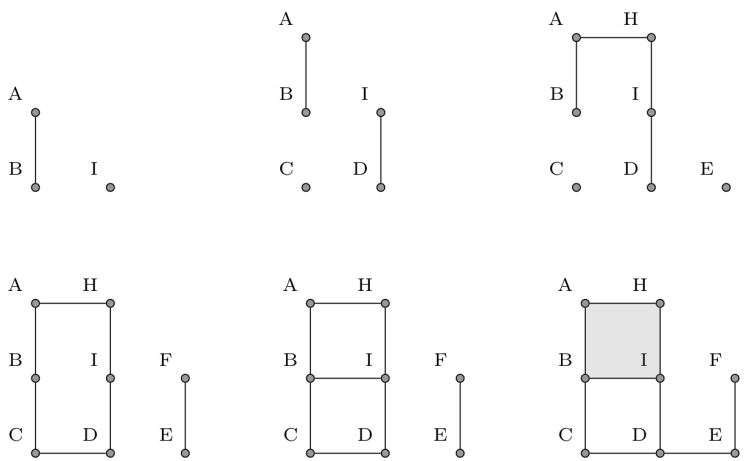
Persistent homology of a cubical complex

We call a filtration a growing family of cubical complexes (2) $(K_r)_{r\in\mathbb{N}}$.

$$K_0 \subset K_1 \subset \dots K_N. \tag{2}$$

We call persistent homology groups $(H_k(K_r), \iota_r^s)_{r\leq s\in\mathbb{N}}$, where $\iota_r^s : H_k(K_r) \to H_k(K_s)$.

Example



$$\begin{aligned} D_0 = &\ \{(0,\infty),(0,2),\\ &\ (1,3),(2,5)\}, \\ D_1 = &\ \{(3,\infty),(4,5)\}. \end{aligned} \tag{3}$$

Upper-star filtration from vertex-defined functions

An image S is a function, defined on a set of vertices $f : V(K) \to \mathbb{R}$.
We extend it (4) to $\tilde{f} : K \to \mathbb{R}$,

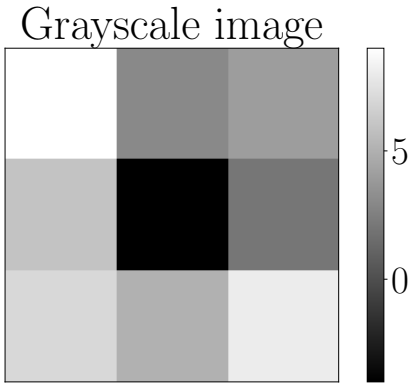
$$\tilde{f}(Q) = \min\{f(v) \mid v \in Q \cap V(K)\}, \tag{4}$$

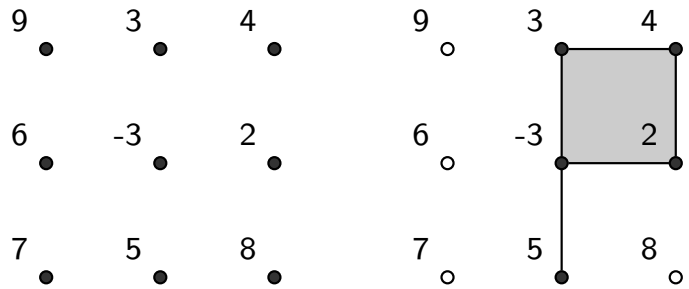
what defines an ordering on all the cubes.
Consider $f : V(K) \to \mathbb{R}$. Then, $(K_{-r})_{r\in\mathbb{R}}$ is a filtration, where

$$K_r = \tilde{f}^{-1}(]r, \infty[). \tag{5}$$

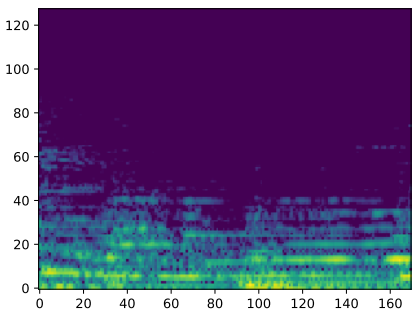
By abuse of terminology, we call the family $(K_r)_{r\in\mathbb{R}}$ a filtration, even though the inclusions are inverted.

Upper-star filtration

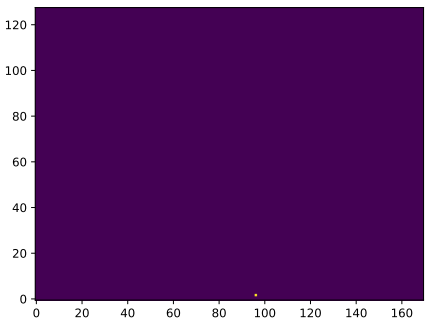




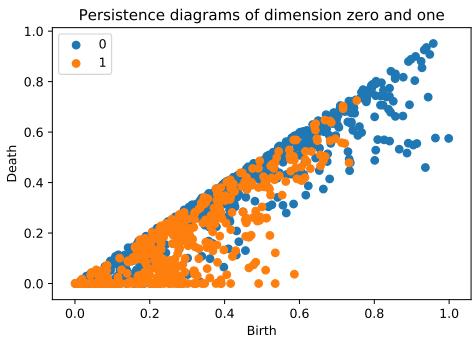
Filtration from a vertex-defined function - example



Image



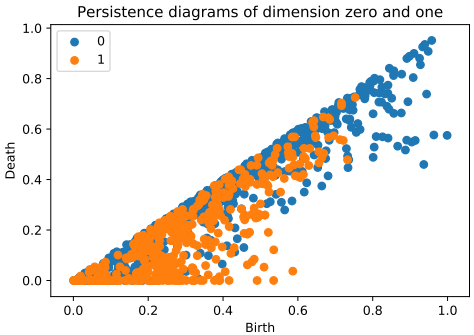
Filtration



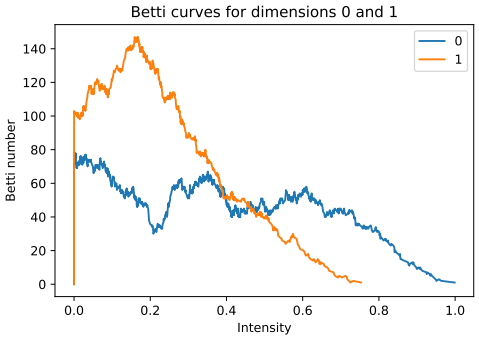
Persistence diagrams

Betti curves encode information from persistent diagrams

$$\begin{array}{rccc} \beta_k : & \mathbb{R} & \to & \mathbb{N} \\ & x & \mapsto & \sum_{(d,b)\in D_k} 1_{]d,b]}(x), \end{array} \tag{6}$$



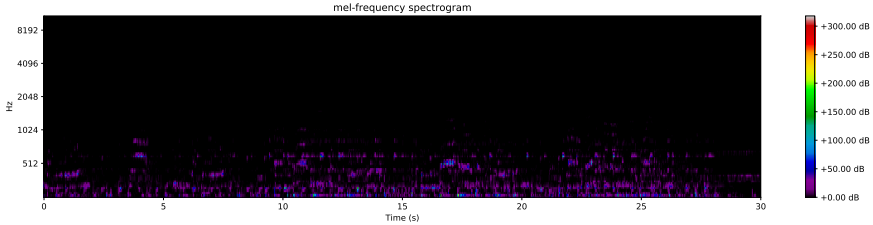
Persistence Diagrams

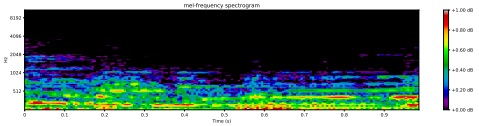


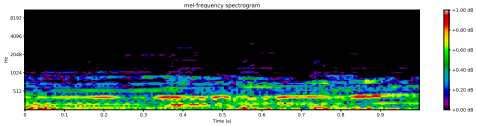
Betti curves

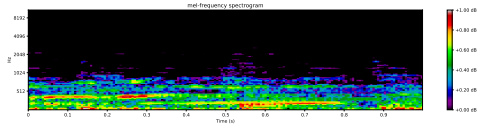
Table of Contents

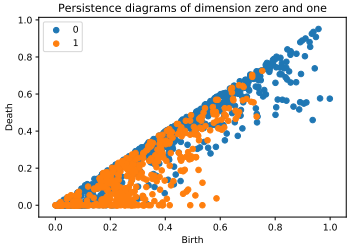
Fingerprinting a song using time-windows

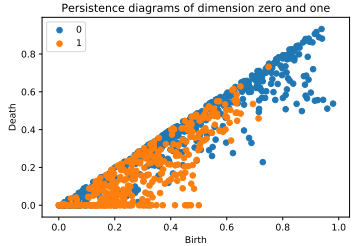


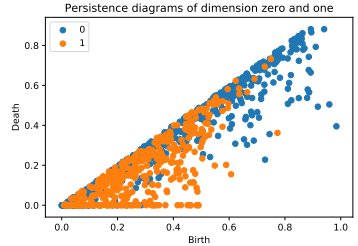


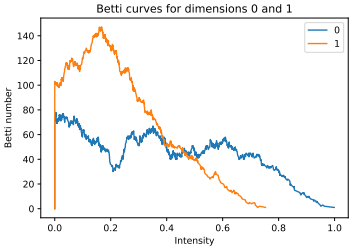


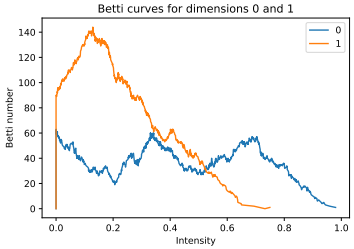


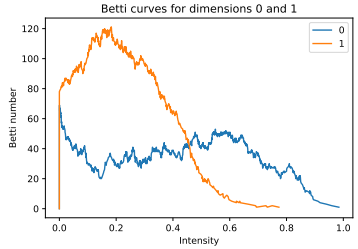












Fingerprinting a song using time-windows

Compute the mel-spectrogram S;
Extract I windows $(S_i)_{i=1}^{N_s}$, where $S_i = S_{:,t_1^i:t_2^i}$;
Initialize fingerprint $\Phi(s) = []$;
for all $\tilde{S} \in (S_i)_{k=1}^{N_s}$ **do**
    Calculate the filter function $f_{\tilde{S}}$;
    Build the complex $K_{\tilde{S}}$;
    Calculate the ph groups $H_k(K_{\tilde{S}}, f_{\tilde{S}})$;
    Calculate Betti curves β_0, β_1;
    $\Phi(s) = \Phi(s) \cup \{(\beta_0, \beta_1)\}$;
end for
return $\Phi(s)$;

Querying

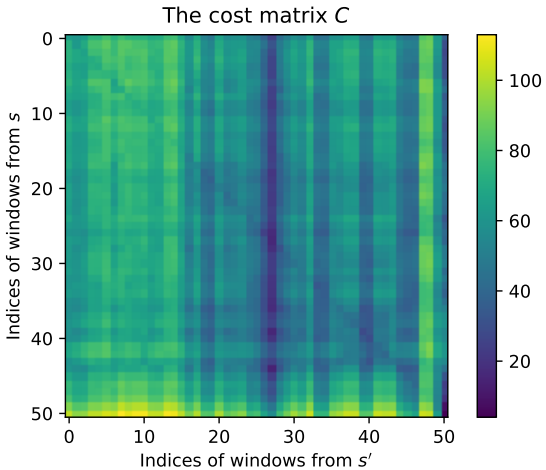
General idea: estimate an alignment (time shift) of s in s'.

Matching algorithm

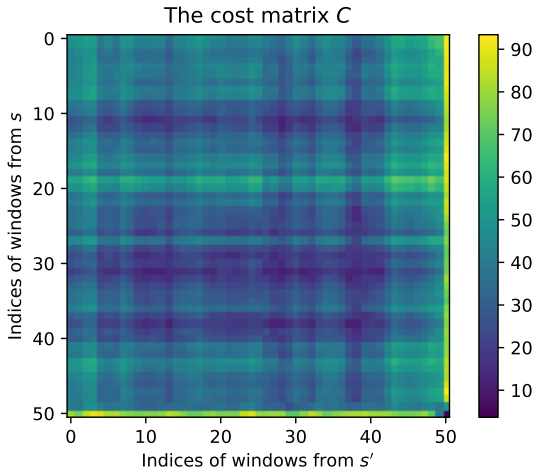
Compute $(\beta_{i,k})_{k=0,1,i=1,\ldots,N_s}$, $(\beta_{j,k})_{k=0,1,j=1,\ldots N_{s'}}$ fingerprints of s, s';
Compute the distance matrices $(M_k)_{i,j} = \|\beta_{i,k} - \beta'_{j,k}\|_{L^1}$, for dim $k = 0, 1$;
Set $C = \lambda_0 M_0 + \lambda_1 M_1$;
Find a minimal cost matching $\gamma : \{1, \ldots, N_s\} \to \{1, \ldots, N_{s'}\}$ in C;
Model $\{(t^i, t^{j'}) \mid j = \gamma(i)\}$ with a linear regression, estimate a, t_{offset};
Calculate the median error of the model $r = \operatorname{median}_{i=1}^{N_s} |t^{\gamma(i)'} - (at^i + t_{offset})|$;
return True, if $r < T_r$;

The threshold T_r is estimated empirically, to achieve desired precision.

Example



Matching (noise obfuscations)



Not matching.

Table of Contents

Dataset

A subset of 35393 from the Million Song Dataset [BMEWL11].
A set of pairs of songs, with a ground truth (match or not).

Type	Degree					
Low-pass filter	1000,	1500,	2000,	3000,		
High-pass filter	1500,	2000,	2500,	3000,		
Pitch shift	0.8	0.85	0.9	1.05	1.15	1.2
Tempo shift	0.8	0.85	0.9	1.05	1.15	1.2
White-noise	0.05	0.1	0.2	0.4		
Pink-noise	0.05	0.1	0.2	0.4		
Reverb	40	70	100.			

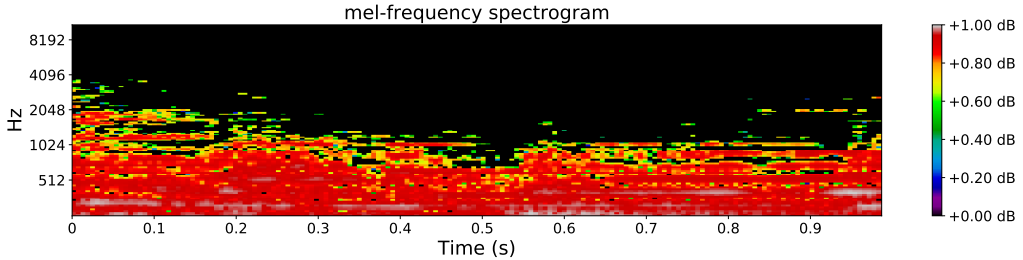
Dataset details

The dataset the experiments were run on.

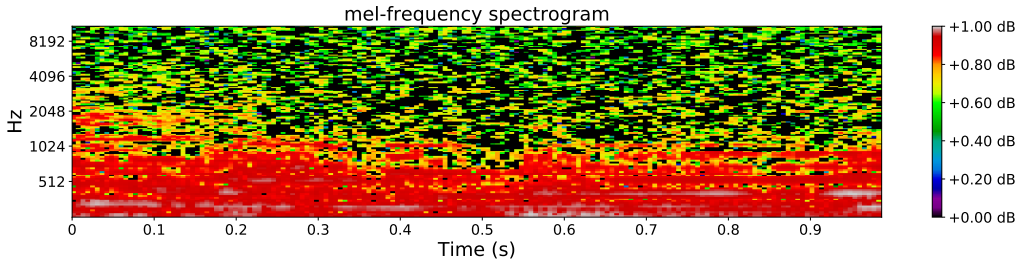
Obfuscation	Degree					
	0	1	2	3	4	5
None	784	-	-	-	-	-
High-pass	27	22	21	30	-	-
Low-pass	67	25	21	23	-	-
White-noise	26	30	18	20	-	-
Pink-noise	26	33	29	21	-	-
Pitch shift	28	19	29	27	18	27
Tempo shift	25	20	22	21	21	27
Reverb	25	33	19	-	-	-

Example of an obfuscated song

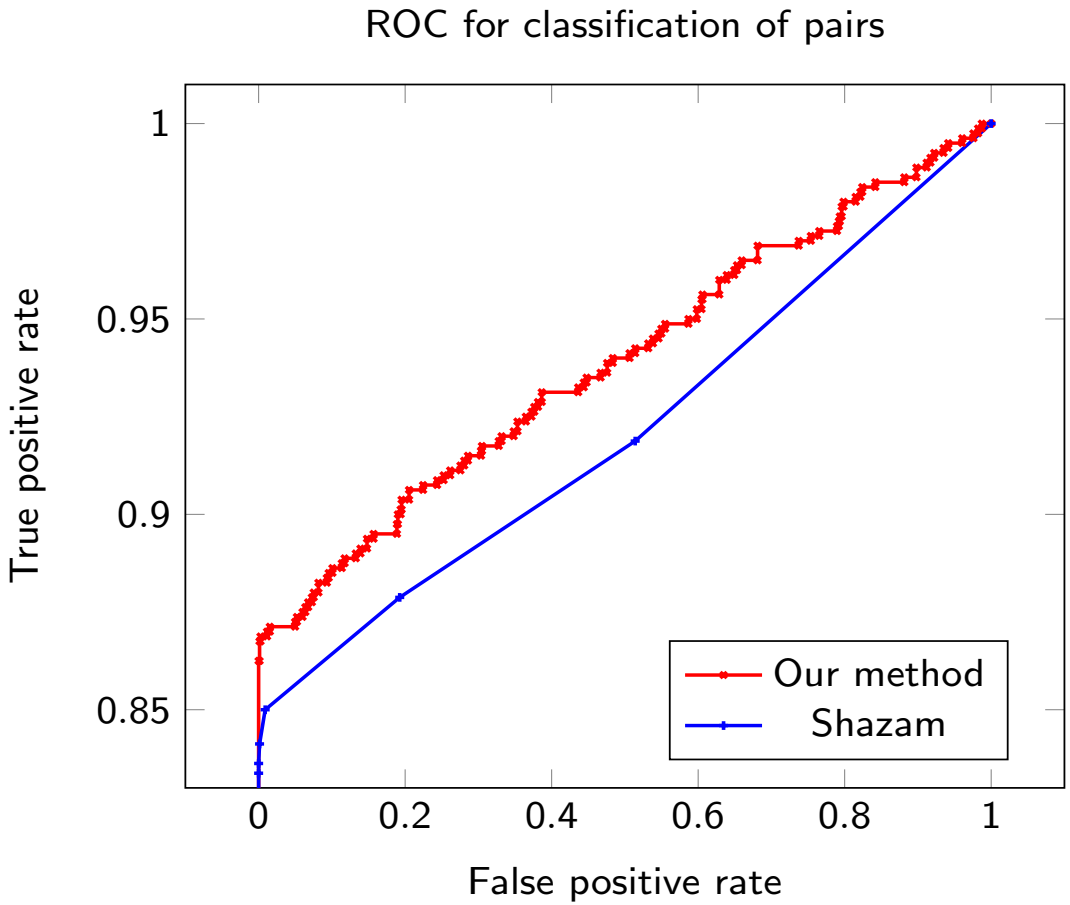
Original:



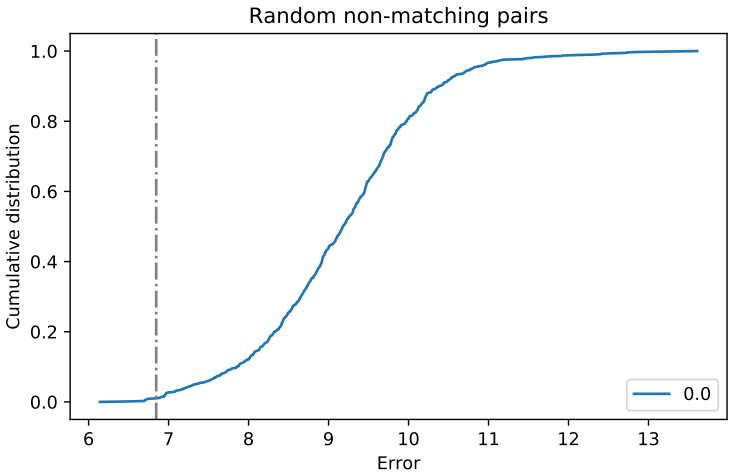
White noise (0.1):

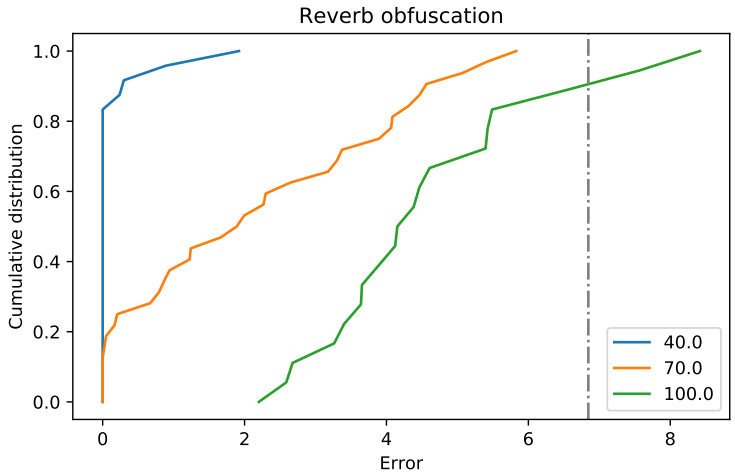


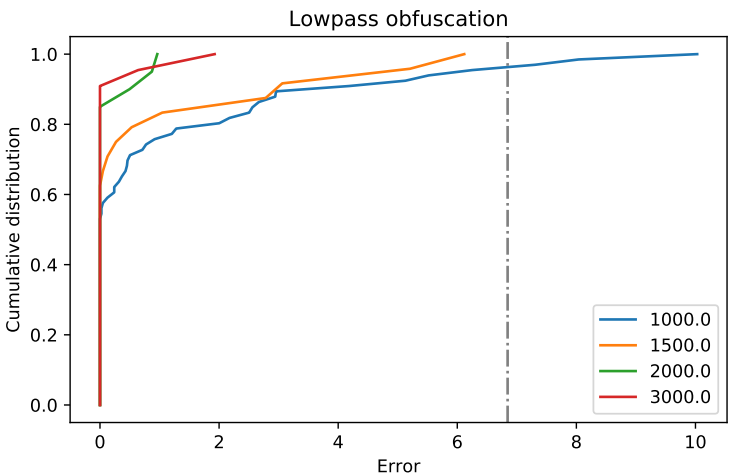
Results

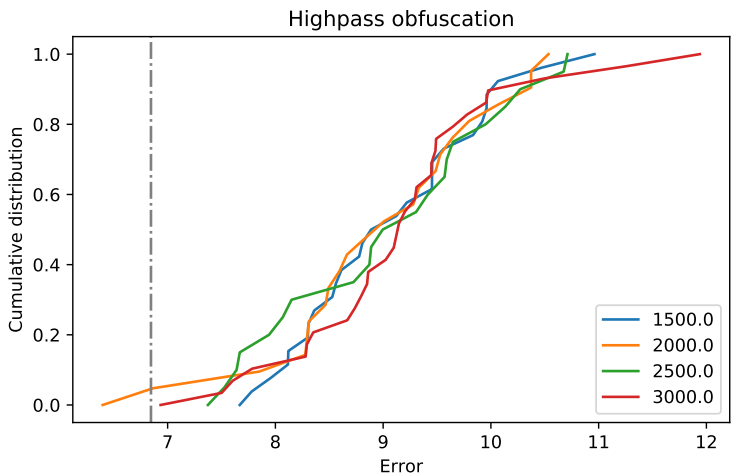

ROC for classification of pairs
True positive rate
False positive rate
1
0.95
0.9
0.85
0
0.2
0.4
0.6
0.8
1
Our method
Shazam

Results by obfuscation

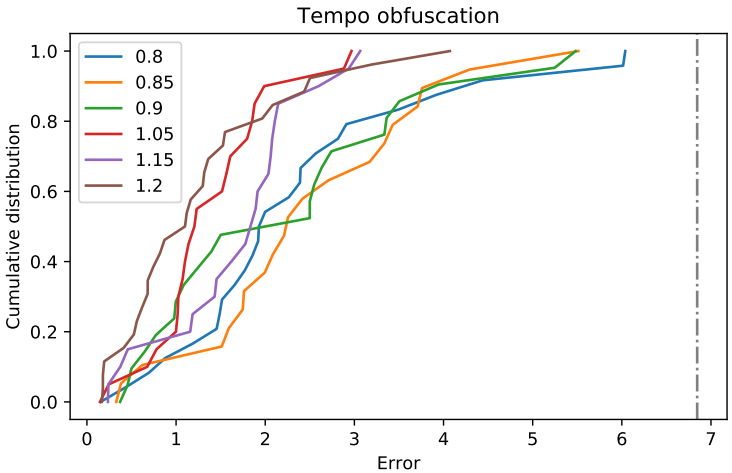


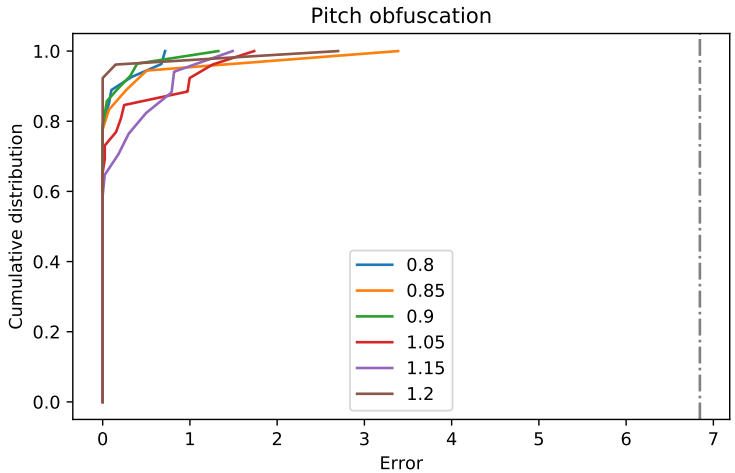


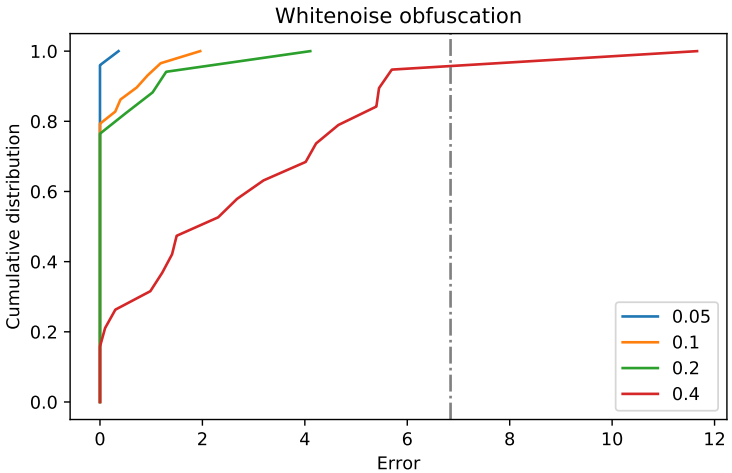


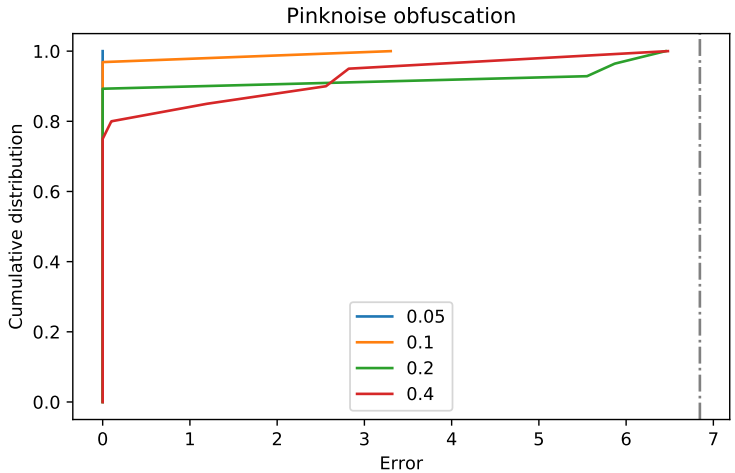


Results by obfuscation

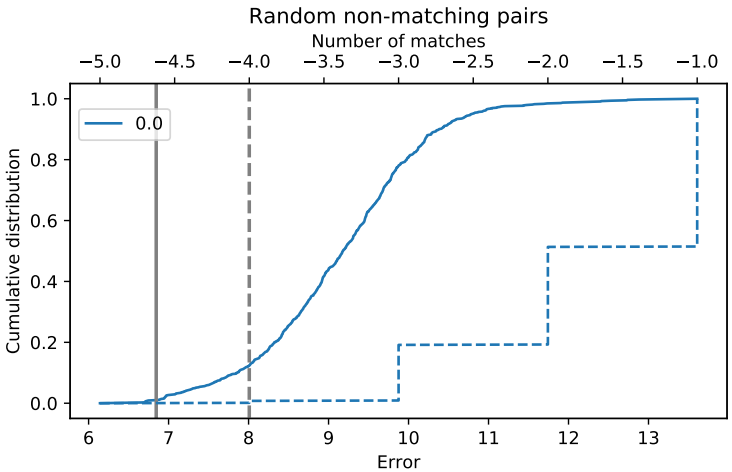


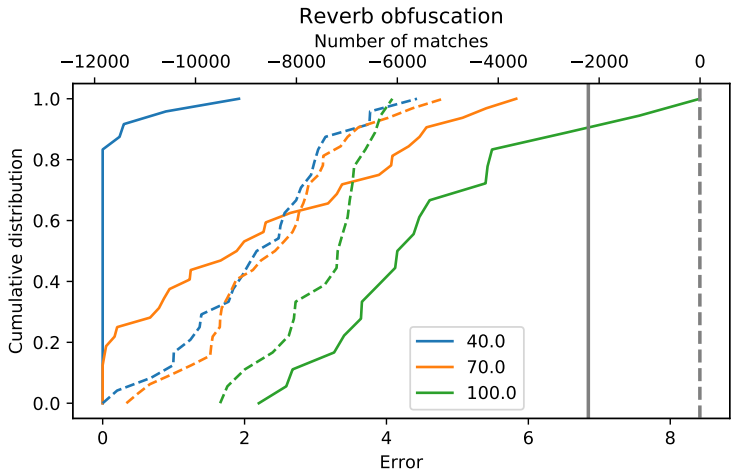


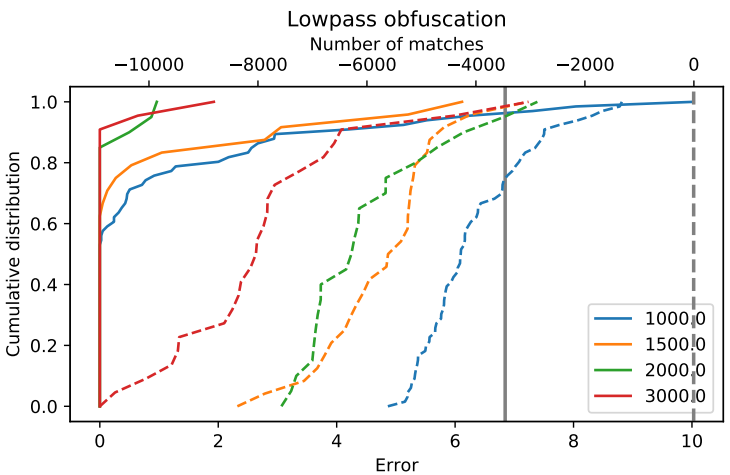


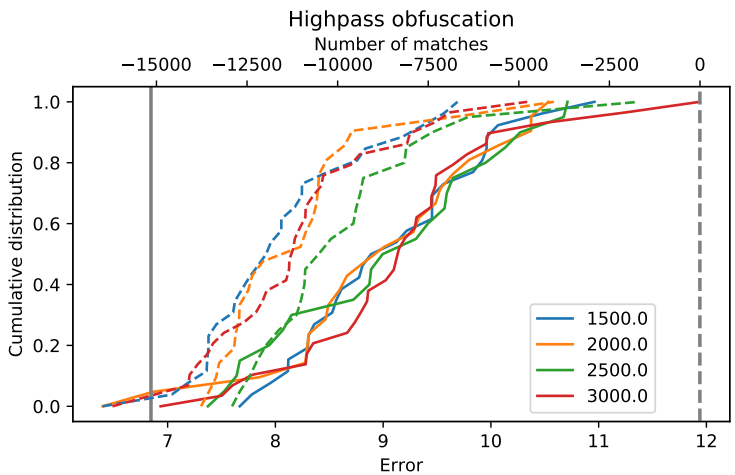


Results by obfuscation against benchmark

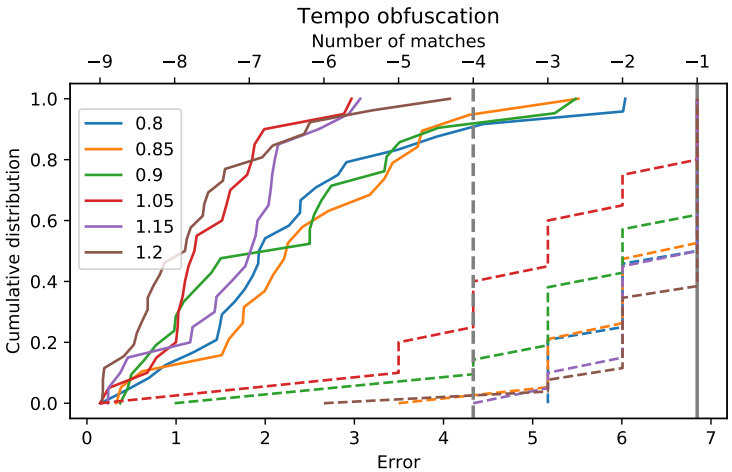


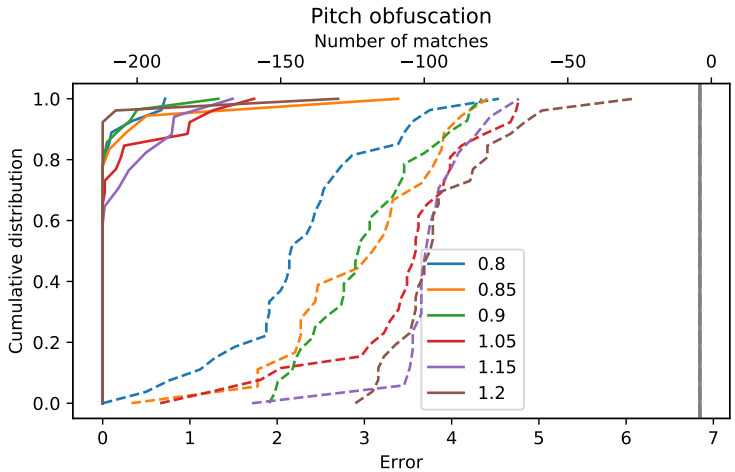


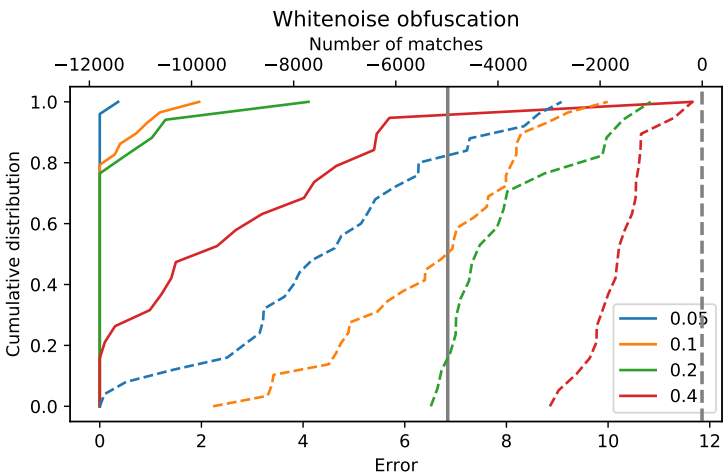


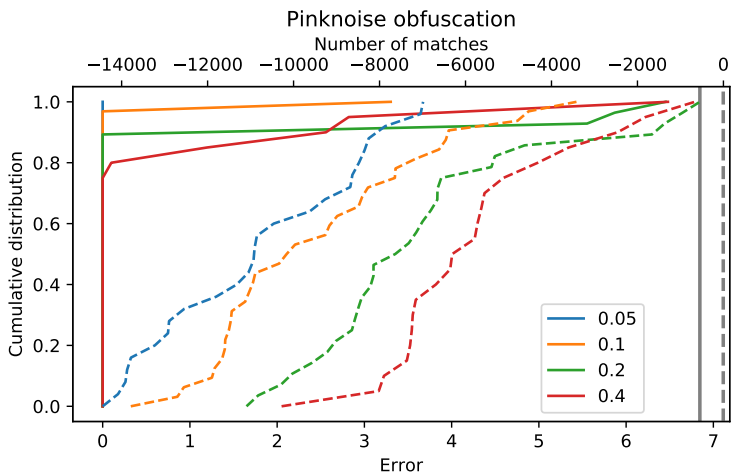


Results by obfuscation against benchmark









Comments

- The proposed method performs well on all obfuscations but low and high pass filters.
- Betti curves are sufficient to measure the discrepancy between topological features of spaces.
- An ‘alignment’ algorithm is necessary.

To improve

1. The performance is linked to the matching algorithm- some windows are close to all windows from the other snippet.

2. High computational complexity - fingerprinting and comparing two songs takes 40s with our method, against 10s with Shazam.

3. Generalize to one-vs-N.

4. Low performance on low- and high-pass filters - distribution of intensity values encoded in the features.

To improve and possible solutions

1. The performance is linked to the matching algorithm- some windows are close to all windows from the other snippet.
   - Normalize features.
2. High computational complexity - fingerprinting and comparing two songs takes 40s with our method, against 10s with Shazam.
   - Re-use computations from different windows - for example, using vineyards.
3. Generalize to one-vs-N.
   - Use a hash function, and a nearest neighbour search (linked to 1.)
4. Low performance on low- and high-pass filters - distribution of intensity values encoded in the features.
   - Consider several frequency ranges: separately, or using vineyards).


Shumeet Baluja and Michele Covell.
Audio Fingerprinting: Combining Computer Vision & Data Stream Processing.
In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, pages II–213–II–216, Honolulu, HI, USA, 2007. IEEE.


Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere.
The million song dataset.
In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.


Frdric Chazal and Bertrand Michel.
An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists.
arXiv:1710.04019 [cs, math, stat], October 2017.
arXiv: 1710.04019.


Vin de Silva, Primoz Skraba, and Mikael Vejdemo-Johansson.
Topological Analysis of Recurrent Systems.
page 5, 2012.


Julius O. Smith.
Spectral Audio Signal Processing.
http://ccrma.stanford.edu/jos/sasp/, 2011.
online book, 2011 edition.


Tomasz Kaczynski, Konstantin Michael Mischaikow, and Marian Mrozek.
Computational homology.
Springer, New York; London, 2011.
OCLC: 1063425526.


Nicole Sanderson, Elliott Shugerman, Samantha Molnar, James D. Meiss, and Elizabeth Bradley.
Computational Topology Techniques for Characterizing Time-Series Data.
arXiv:1708.09359 [cs], August 2017.
arXiv: 1708.09359.


Avery Li-Chun Wang.
An Industrial-Strength Audio Search Algorithm.
page 7, 2003.


Yan Ke, D. Hoiem, and R. Sukthankar.
Computer Vision for Music Identification.
In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 597–604, San Diego, CA, USA, 2005. IEEE.


Afra Zomorodian.
Computing Persistent Homology.
Discrete & Computational Geometry, 33(2):15, February 2005.

Q&A

Audio identification on obfuscated tracks

Previous work

AUDIO FINGERPRINTING:
COMBINING COMPUTER VISION & DATA STREAM PROCESSING

Shumeet Baluja & Michele Covell

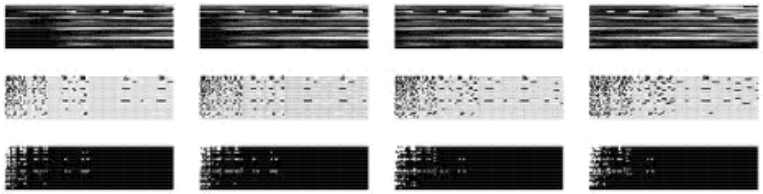
Google, Inc.
1600 Amphitheatre Parkway, Mountain View, CA. 94043
{shumeet,covell}@google.com

Fingerprinting

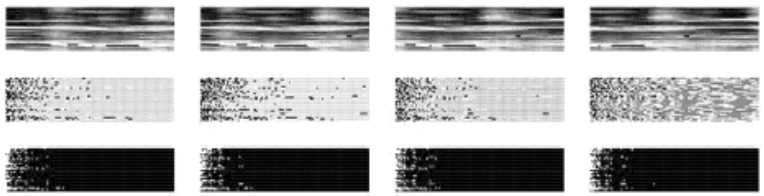
Create spectral representations, with random spacing.
for all spectral images S **do**
Compute a wavelet decomposition of S.
Extract top wavelets.
Create a binary vector of top wavelets.
end for
return Binary vectors.

Querying - count the number of similar fingerprints.

*The Dave Matthews Band – Lie in Our Graves (album **Crash**)*



*Enya – Shepherd Moons (album **Shepherd Moons**)*



Vines

Definition

Define auxiliary functions $(f_{r,\tilde{S}})_{r \in R}$, where r denotes the index of the lowest row in the filtration.

$$\begin{aligned} f_{r,\tilde{S}} : \quad V(K_{\tilde{S}}) &\to \mathbb{R} \cup \{-\infty\} \\ (i,j) &\mapsto \begin{cases} -\infty & \text{, if } r > i, \\ \tilde{S}_{i,j} & \text{, otherwise.} \end{cases} \end{aligned} \tag{7}$$

Vines

Example

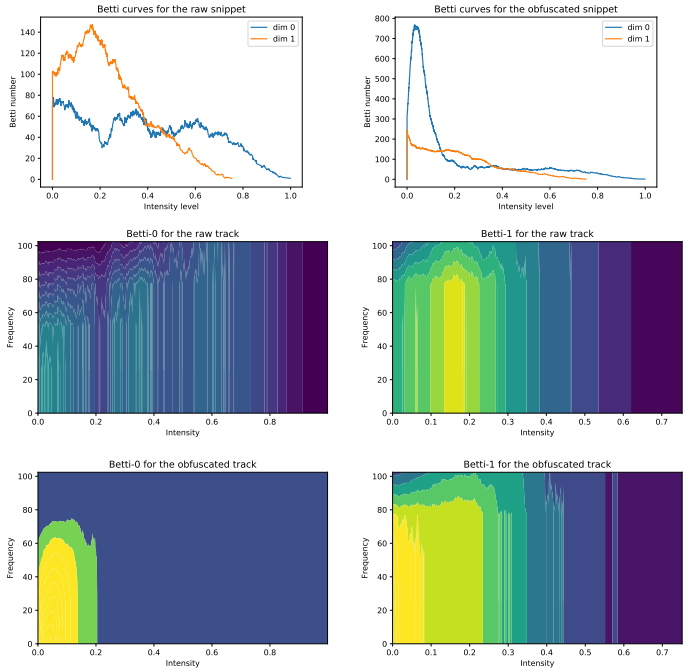


Figure: The stacked Betti curves of dimensions zero and one, on the left and right respectively. Each row represents a Betti curve, for a fixed value of $r \in R$. Here, we have used $R = \{0, 25, 51, 76, 102\}$. The bottom row shows the features for the obfuscated version.

Homology groups of cubical complexes

Definition

We call k-chains $C_k(K)$ the vector space on $\mathbb{F}$, generated by elementary cubes of dimension k, namely

$$C_k(K) = \left\{ \sum_{Q \ k-\text{cube}} a_Q \widehat{Q} \mid a_Q \in \mathbb{F}_2 \right\}.$$

Definition

For cubes of dimensions zero, one and two, the boundaries

$$\partial_k : C_k(K) \to C_{k-1}(K)$$

can be defined as

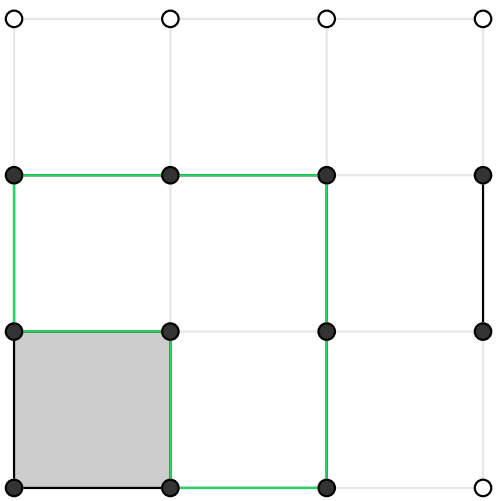
$$\begin{aligned} \partial_2 ([a,b] \times [c,d]) &= \partial_1([a,b]) \times [c,d] - [a,b] \times \partial_1([c,d]), \\ \partial_1 ([a,b]) &= b - a, \\ \partial_0 ([a,a]) &= 0. \end{aligned}$$

Homology groups of cubical complexes

Homology groups H_k are defined as

$$H_k(K) = \ker(\partial_k)/\mathrm{im}(\partial_{k+1}).$$

Example



$$H_0(K) = \mathbb{F}_2^2,$$
$$H_1(K) = \mathbb{F}_2.$$

Cubical complex

Definition (following [KMM11])

A closed interval $I = [a, b] \subset \mathbb{R}$ is called elementary if $b \in \{a, a+1\} \subset \mathbb{Z}$. In the case $a = b$, the interval is called degenerate.
An elementary cube $Q = I_1 \times \ldots \times I_d \subset \mathbb{R}^d$ is a product of d elementary intervals.

Example

A vertex (0-cube) is a product of degenerate intervals. An edge (1-cube) is a product of intervals where all but one are degenerate.

Definition

Let K be a collection of cubes of the same embedding dimension. Then, K is a cubical complex if

- for any cube $Q \in K$, its faces are also in K,
- for all cubes $Q_1, Q_2 \in K$, the intersection $Q_1 \cap Q_2 \in K$ is either empty or a face of Q_1 and Q_2,

Persistent homology for signal processing

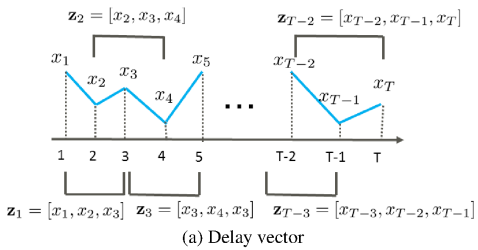
Delay embeddings allows us to describe the recurrent nature of signals

Definition

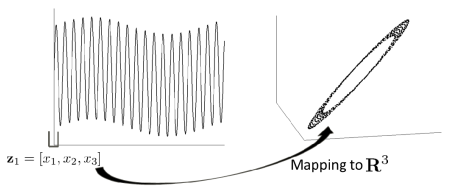
Let n_τ be a delay and τ a period. Then, we call the point cloud $\{(s_k, s_{k+\tau}, \ldots, s_{k+\tau n_\tau})\}_{k=1}^{N-\tau n_\tau}$ a delay embedding of the signal $(s_k)_{k=1}^{N}$.

Persistent homology on delay embeddings has been used to

- characterise dynamical systems [dSSVJ12],
- time series classification [**?**],
- instrument identification [SSM+17],
- visualising the structure of songs [**?**].



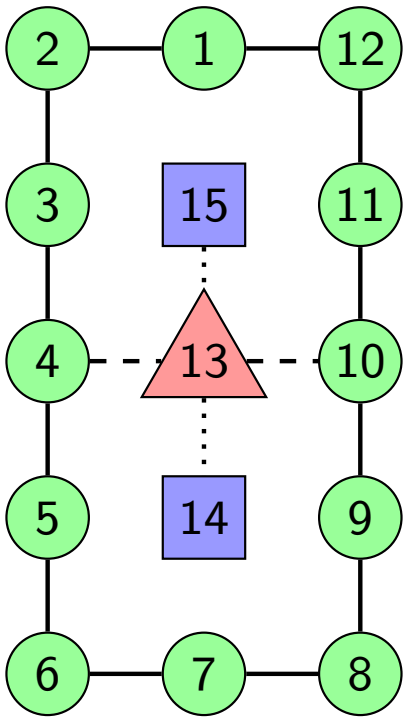
(a) Delay vector

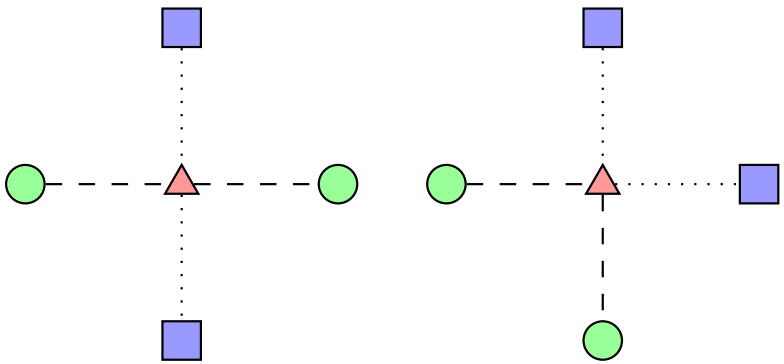


(b)Quasi-attractor

Fig. 2 (a)Delay vectors generated from observed time series. (b) Delay vectors map to three-dimensional Euclidean space. The set of points mapped from the delay vectors is the quasi-attractor.

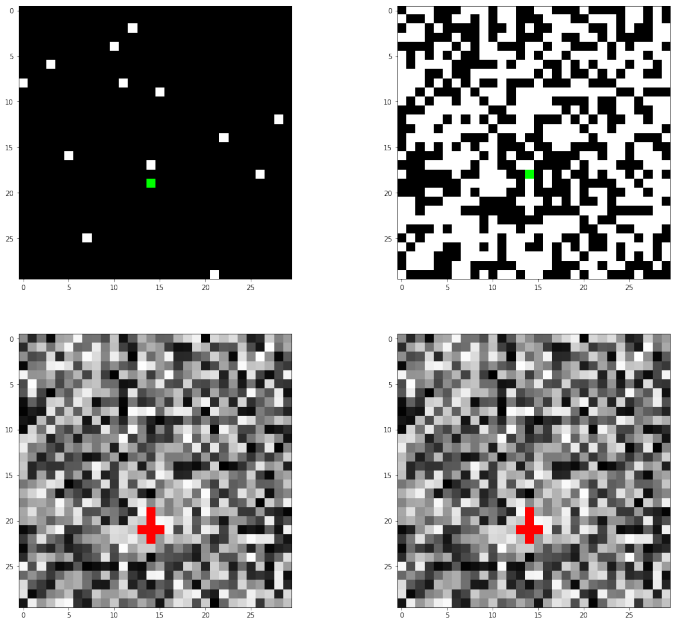
Duality in sup- and sub-level filtrations





The point $(13, 14)_1$ is in the persistence diagram of the sub-level filtration, while $(13, 14)_0$ appears in the persistence diagram of the super-level set filtration.

Example



We look at the birth death pair $(13, 490)$ from the sub-level filtration, which has a corresponding one $(490, 13)$ in the super-level set filtration. It is worth noting that the filter function is comprised of unique integers in $[0, 900]$. In the top row, we can see the sub-level filtration of a random image, with the birth (left) and death (right) pixels marked in green. In the bottom row, we use the Volume-Optimal Cycle (VOC) approach to find a geometric realization of a persistent homology generator - VOC cycles for the sub- and super-level filtrations are shown on the left and right pictures respectively.