# Audio Bigrams as a Unifying Model of Pitch-based Song Description

Jan Van Balen, Frans Wiering, and Remco Veltkamp

Department of Information and Computing Sciences,
Utrecht University, The Netherlands

**Abstract.** In this paper we provide a novel perspective on a family of music description algorithms that perform what could be referred to as 'soft' audio fingerprinting. These algorithms convert fragments of musical audio to one or more fixed-size vectors that can be used in distance computation and indexing, not just for traditional audio fingerprinting applications, but also for retrieval of cover songs from a large collection, and corpus-level description of music. We begin with a high-level overview of the algorithms. Next, we identify and formalize an underlying paradigm that allows us to see them as variations of the same model. Finally, we present PYTCH, a Python implementation of the model that accommodates several of the reviewed algorithms and allows for a variety of applications. The implementation is available online and open to extensions and contributions.

**Keywords:** Audio fingerprinting, Cover detection, Convolutional neural networks

## 1 Introduction

### 1.1 Song Description and Audio Fingerprinting

Robust, large-scale audio fingerprinting was one of the first achievements in music information retrieval to cause a ripple outside the field, with the first effective algorithms being developed in the early 2000's by Haitsma and Kalker at Philips, and Wang and Smith for Shazam [5, 14]. Twelve years later, these same audio fingerprinting algorithms still form the backbone of music search services across industry, reliably identifying a single exact music fragment in a collection of millions of songs. Yet, crucially, they are unable to identify covers, live renditions, hummed versions, or other variations of a piece.

Several attempts have been made to adapt the concept of fingerprinting to such use cases, which require invariance to intentional and performance-related changes to the song. We refer to such systems as 'soft audio fingerprinting' systems.

In this broadened interpretation, fingerprinting can be defined as any reduction of a large audio object to a compact, representative digest. Classic fingerprinting systems like Shazam[1] and Soundhound[2] use such representations to identify short musical fragments, by matching the fingerprint of the unlabeled fragment to a large reference database. State-of-the-art algorithms for audio fingerprinting produce fingerprints with a high degree of robustness to noise, compression and interference of multiple signals [4] and perform matching of fingerprints very efficiently.

An important distinction between this notion of fingerprinting and other kinds of document retrieval, is the fixed size of the representations and the use of an index to store them. Consider for example what is seen as the most challenging of song retrieval applications: cover song detection. Several strategies for the retrieval of cover songs exist, yet only a few of these strategies use indexing. Most, including the best-performing systems, rely on alignment to assess the similiarity for every pair of songs. Since alignment is expensive, and direct comparison of songs results in search times linear with the size of the dataset, alignment-based algorithms are not a good solution for large-scale cover song retrieval [6]. Unfortunately, of all index-based cover song retrieval algorithms proposed so far, none have reached performance numbers close to those of the alignment-based systems.

Admittedly, the situation for some related tasks, such as query by humming, is better. There is little information about the exact workings of commercial services such as Soundhound's MIDOMI[3], but they work. However, they are generally understood to rely on matching (alignment or otherwise) of simplified contours of melodies sung by volunteers, rather than matching hummed melodies with a song's original audio, which remains an unsolved problem. By and large, the continuing prevalence of alignment-based methods illustrates the need for a fresh perspective on fingerprinting.

### 1.2 Contribution and Outline

In this paper, we introduce a unifying perspective on current song description strategies by pointing out some essential commonalities. We then propose a general model of song-level music description, that accomodates a large family of approaches, including some of the existing ones. We argue that the fingerprinting architecture proposed in this paper has the potential to produce both very complex and very interpretable musical representations. Finally, we present an implementation of the proposed computational pipeline that can be used for the comparison of fingerprints, and for song and corpus description in general, alongside an example experiment that illustrates its use.

[1] `http://www.shazam.com/`
[2] `http://www.soundhound.com/`
[3] `http://www.midomi.com/`

First, however, we succinctly review a number of fingerprinting and soft fingerprinting techniques, beginning with the seminal 'landmark-based' strategy originally proposed by Wang.

## 2 Overview of Audio Fingerprinting and Soft Audio Fingerprinting Algorithms

### 2.1 Landmark-based fingerprinting

Like most fingerprinting systems, Wang's system includes an extraction and a matching component [14]. In the extraction component, a piece of audio is first converted to a spectrogram. The most prominent peaks in the spectrogram are detected and paired based on proximity. Pairs of peaks are called landmarks, and can be fully described by 4 parameters: the frequencies of the peaks, a start time, and the time interval between the peaks. In a last step, the two peaks frequencies and time interval are combined into a hash code for efficient look-up. The reference database is constructed by storing all hashes for a collection of songs into an index, where each hash points to the landmark start time and a song ID.

In the matching stage, a query is passed to the system. Its landmarks and corresponding hashes are computed. Any matching landmarks from other songs are then retrieved from the reference database, with their corresponding start time and song ID. Note that this can be done in constant time. In the last step, the systems determines for which landmarks the start times are consistent between query and candidate, and the song with most consistently matching landmarks is returned as the result.

### 2.2 Constant-Q-based Fingerprinting

At least three systems in the literature have succesfully applied Wang's ideas to spectrograms for which the frequency axis is divided into logarithmic rather than linearly spaced bins [3,10,13]. Such spectral representations are generally referred to as constant-Q transforms. All three systems aim to produce fingerprints that are robust to pitch shifting, a tranformation of musical audio that is often applied by DJ's at radio stations and in clubs. The system by Van Balen specifically aims to identify cases of sampling, a compositional practice in which pieces of recorded music are transformed and reused in a new work. In each case, the idea is that the constant-Q spectrogram preserves relative pitch.

### 2.3 Chroma- and Melody-based Fingerprints

Other soft audio fingerprintinging systems are simpler in concept and resemble Wang's original strategy a little less, among them a procedure proposed by Kim et al. [7]. This system takes 12-dimensional chroma and delta chroma features and computes their $12 \times 12$ covariance matrices to obtain a global fingerprint for each song. This relatively simple strategy achieves good results on a test set of classical music pieces. A later extension by the same authors introduces the more sophisticated 'dynamic chroma feature vectors', which roughly describe pitch intervals [8].

A similar family of fingerprints was proposed by Van Balen in [12]. Instead of the covariance matrix, the correlation matrix of the chroma features is used. The feature is also $12 \times 12$ in size. Compared to the covariance, correlation coefficients introduce more invariance to the overall ubiquity of a pitch; it picks up less occurence and more actual concurrence. A second fingerprint is based on the melody as extracted by a melody estimation algorithm, and aims to measure transitions in melodic pitch. The fingerprint counts co-occurrence of pitches given a certain maximum offset in time, respecting order.

The above fingerprints proposed in [7] and [12] can be seen as a continuous-time, audio-domain analogues of bigram representations often used in symbolic music research. Bigrams are pairs of objects, typically letters or words in a text or notes in a score. Distributions of bigrams have been succesfully used to model pitch-related expectations [11] and the evolution of style in Western classical music [9].

### 2.4 Jumpcodes and the 2D Discrete Fourier Transform

Two studies by Bertin-Mahieux and others have taken the idea of fingerprinting and applied it to truly large-scale retrieval of cover songs. The first study aims to discover to which extent the landmark-based approach can be used with 12-dimensional chroma instead of the spectrogram. In an evaluation using the very large Million Song Dataset (MSD), the approach was found to be relatively unsuccesful [1, 2].

The second study therefore follows an entirely different approach. Beat-aligned chroma features are transformed using the two-dimensional Discrete Fourier Transform (DFT), to obtain a compact summary of the song that is somewhat hard to interpret, but roughly encodes recurrence of pitch intervals in the time domain [2]. Since only the magnitudes of the Fourier coefficients are used, this '2DFTM' approach is robust to both pitch shifting and tempo changes by design. Results are modest with a mean average precision in the MSD in the order of 0.03, but formed the state-of-the art in scalable cover song retrieval at the time.

Humphrey et al. have taken this idea further by applying a number of feature learning and dimensionality reduction techniques to the above descriptor, with the aim to construct a sparse geometric representation that is more robust against the typical variations found in cover songs [6]. The method performs an initial dimensionality reduction on the 2DFTM features, and then uses the resulting vectors to learn a large dictionary (using $k$-means clustering), to be used as a basis for a higher-dimensional but sparser representation. Finally, supervised Linear Discriminant Analysis (LDA) is used to learn a reduced embedding optimized for cover song similarity. Their method achieves an increase in precision, but not in recall.

## 3 Unifying Model

### 3.1 Fingerprints as Audio Bigrams

The song description model we propose in this paper builds on the following observation: many of the fingerprinting methods listed in the previous section can be reduced to a combination of detecting salient events in a time series, and pairing these over different time scales to obtain a set of bigram distributions.

The paradigm identified here will be referred to as the *audio bigram* paradigm of fingerprinting. We propose the following formalisation:

The audio bigram model is a fingerprinting paradigm in which a system

1. extracts salient events from a multidimensional time series $M$, to produce a time series $P$ of lower density,
2. computes co-occurrences of the events in $P$ over $K$ different timescales, to produce a fixed-size fingerprint $F$.

We proceed to show how the example systems from section 2 can easily be mapped to the above formulation of the model.

For the case of the Wang's landmark approach, the salient events are peaks in the linear spectrum ($M =$ DFT magnitudes), and the time scales for pairing are a set of fixed, linearly spaced offsets up to a maximum horizon $\Delta t$.

$$\tau = 1, 2 \ldots \Delta t,$$

yielding a set of $i$ peak bigram distributions for the total of the fragment.

The constant-Q and chroma landmark-based systems reviewed above are essentially analoguous, with salient events as peaks in the logarithmic spectrum ($A =$ CQT) and beat-aligned chroma features, respectively.

For the case of the melody bigrams, the salient events are pitch activations pertaining to the melody and the time scale for pairing is a single range of offsets

$$\tau \in (1, \Delta t).$$

Chroma (and delta chroma) covariance and correlation matrix features are even simpler under this paradigm: pitch activations are only paired if they are simultaneous, i.e. $\tau = 0$.

The only approaches that don't seem to be accommodated at first sight, are Bertin-Mahieux and Humphrey's algorithms based on the 2D Fourier transform. In the remainder of this section, we will show that:

- a formulation of the audio bigram model exists that has the additional advantage of easily being vectorized (expressed as vector and matrix operations, for efficient computation),
- the vectorized model is conceptually similar to the 2D Fourier transform approach to cover song fingerprinting,
- the model is closely related to convolutional neural network architectures and can be used for feature learning.

It is good to point out that the model will not accommodate all of the algorithms completely. Notably, in the adaptation of landmark-based fingerprinting as described here, some of the time information of the landmarks is lost, namely, the start time of the landmarks. We believe this can ultimately be addressed,[4] but currently don't foresee any such adaptation, as the primary aim at this stage is to explore and evaluate the commonalities between the algorithms.

### 3.2 Efficient Computation of Audio Bigrams

In this section, we further formalize the model and characterize its computational properties by proposing a vectorized reformulation. Vectorized computations are expressed in terms of matrices and vectors, and are crucial in optimizing computational performance. The first step to be examined is the detection of salient events.

**Salient Event Detection** In its simplest form, we may see this component as a peak detection operation, where we define peak detection as the transformation that sets a matrix cell to 1 if its value is greater than any of the values in its immediate surroundings, and 0 otherwise.

Peak detection may be vectorized using dilation. Dilation, denoted with $\oplus$, is an operation from image processing, in which the value of a pixel in an image or cell in a matrix gets set to the maximum of its surrounding values. Which cells or pixels constitute the surroundings is specified by a small binary mask or 'structuring element'.

Given a masking structure $S_m$, a complete mask for $X$ is given by $S_m \oplus X$ and peaks $P$ are those positions in $X$ where $X = S_m \oplus X$. More precisely,

$$P = h(X - S_m \oplus X)$$

[4] e.g. by not extracting one global fingerprint, but fingerprinting several overlapping segments and pooling the result, cfr. [2, 6]

where

$$h(x) = \begin{cases} 1 & \text{if } 1 \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

the heaviside (step) function.

As often in image processing, convolution, denoted with $\otimes$, can be used as well. We get:

$$P = h(X - S_m \otimes X)$$

where $h(x)$ as above, or if we wish to retain the peak intensities,

$$h(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

the rectification function.

Equivalently, we may write

$$P = h(S \otimes X)$$

where $S$ is a negative kernel with center 1 and all other values equal to $-S_m$, similar to kernels used for edge detection in images (top left in Figure 1).

The latter approach, based on convolution, directly allows for the detection of salient events beyond simple peaks in the time series. Indeed, as in image processing and pattern detection elsewhere, convolutional kernels can be used to detect a vast array of very specific structures ranging, for this model, from specific intervals (e.g. fifths or sevenths) over major and minor triads to interval jumps. See Figure 1 for simplified examples.

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 17 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 8 & -1 & 8 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 5 & -1 & 5 & -1 & 5 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 5 & -1 & -1 & -1 & -1 & -1 \\ 2 & -1 & 2 & -1 & -1 & -1 \\ -1 & -1 & 5 & -1 & -1 & -1 \end{bmatrix}$$

**Fig. 1.** Examples of event-detecting kernels. Rows are time frames, columns can be thought of as pitch classes or frequency bins. Clockwise, they would roughly detect edges or single peaks, a two-semitone interval sounding together, a two-semitone jump, and a triad.

**Co-occurrence Detection** Co-occurrence computations can be vectorized just as easily. Consider that the correlation matrix of a multidimensional feature can be written as a matrix product:

$$F = P^T P$$

where $P$ has been normalized by subtraction of the mean and dividing by the standard deviation (for each dimension). The resulting fingerprint measures the co-occurrence of harmonic pitch classes.

When a certain time window for pairing needs to be allowed, one efficient approach is to apply dilation or convolution prior to the matrix multiplication.

In this case, the structuring element we need is a binary column matrix (size along the pitch dimension is one) of length $\Delta t$, denoted $T$. The melody co-occurrence feature can be defined as

$$F = P^T (T \oplus P)$$

where $P$ is a chroma-like matrix containing 1 when a pitch class is present in the melody $m(t)$ and 0 everywhere else:

$$P_{t,i} = \begin{cases} 1 & \text{if } m(t) = i, \\ 0 & \text{otherwise.} \end{cases}$$

To see how the above $F$ is mathematically equivalent to the proposed co-occurrence matrix, consider that by definition of dilation,

$$(T \oplus P)_{t,i} = \max_{\tau \in T}(P_{t,i-\tau})$$

so that

$$(P^T (T \oplus P))_{i,j} = \sum_t (P_{t,i} * \max_{\tau \in T}(P_{t,j-\tau}))$$

which, for a binary melody matrix $P$ based on the melody $m(t)$, translates to

$$F_{i,j} = \sum_t \max_{\tau \in T} \begin{cases} 1 & \text{if } m(t) = i \text{ and } m(t+\tau) = j, \\ 0 & \text{otherwise,} \end{cases}$$

i.e. the standard definition of the co-occurrence matrix over discrete one-dimensional data.

Alternatively, convolution can be applied, and we get

$$F = P^T (T \otimes P)$$

or

$$F_{i,j} = \sum_t \sum_{\tau \in T} \begin{cases} 1 & \text{if } m(t) = i \text{ and } m(t+\tau) = j, \\ 0 & \text{otherwise,} \end{cases}$$

provided $S$ is again binary.

The difference between these two types of co-occurrence matrix is small for sufficiently sparse $M$, in which case the maximum and the sum over $\tau \in T$ are on the order of 1. This is generally true for natural language data, the context in which co-occurrence matrices were first used. It may also hold for the peak constellations used in classic landmark-based audio fingerprinting. For more general, dense matrices, the convolution-based $F$ will scale with the density of $M$ while the dilation-based $F$ will not. This underlines the advantage of sparsity in the extraction of $P$.

We conclude that the pairing of salient events over different time scales can be completely vectorized for efficient computation using image processing techniques such as dilation, convolution or both.

**Summary** Given an input times series $X$ (time frames as rows), a set of $n$ masking structures $\{S_i\}$ and a set of $K$ structural elements $\{T_k\}$ specifying the time scale for co-occurrence, we apply

1. **salient event detection** using
   - convolution with $S_i$:
   $$X'(t) = [S_i \otimes X](t) \tag{1}$$
   - rectification:
   $$P(i,t) = h([S_i \otimes X](t)]) \tag{2}$$
   $i = 1 \ldots n$.
2. **co-occurrence detection** using
   - convolution with $T_k$:
   $$F(k,i,j) = [P^T \cdot (T_k \otimes P)](i,j) \tag{3}$$
   $i,j = 1 \ldots n$ and $k = 1 \ldots K$.
   - optional normalization.

so that $F(k,i,j)$ in fingerprint $F$ encodes the total amount of co-occurrences of $S_i$ and $S_j$ over time scale $T_k$.[5]

### 3.3 Audio Bigrams, 2DFTM and Feature Learning

Reviewing the 2D discrete Fourier transform-based approaches we noted how, of all song description features, they aren't the most open to interpretation.

[5] For completeness we point out that the above example only covers cases in which all $S_i$ have a number of rows equal to that of $X$, so that each convolution yields a one-dimensional result. In cases where the number of rows in S is smaller, an additional index must be introduced, iterating over the rows of the resulting convolution.

The most straightforward intuition behind the output of the 2D Fourier magnitude coefficients over a patch of chroma features, is that it encodes periodic recurrences in both time and pitch class.

The audio bigram model proposed in this paper measures co-occurrences of events given a set of timescales. In other words, its aspires to do just the same as the 2DFTM-based systems, but dropping the periodicity requirement, pinning down the specifics of what kinds of recurrence in time and pitch space are allowed, and linking it to the bigram paradigm that has been succesful in other strands of audio fingerprinting.

**Audio Bigrams and Convolutional Neural Networks** Last but not least, we demonstrate the most promising aspect of this model, which is its relationship to feature learning systems. Indeed, the above set of transforms is very similar to the architecture of convolutional neural networks as used in computer vision and artificial intelligence.

Convolutional neural networks are a currently popular type of artifical neural networks (ANN) in which a cascade of convolutional filters and non-linear activation functions is applied to an input vector or matrix (e.g. an image). Common non-linear functions include sigmoid functions (e.g. tanh) and the rectification function, used in so-called Rectified Linear Units or ReLU's.

Convolutional neural networks are much like other ANN, in that most of the layers can be expressed as either a linear operation on the previous layer's output or a simple non-linear scaling of the result. The coefficients or 'weights' of these linear operations can be seen as connections between neurons, and make up the majority of the network's parameters. Crucially in ANN, these parameters can be learned given a large dataset of examples.

Learning parameters in the context of variable-length time series presents an extra challenge, since either the output or the number of weights will not be constant. This system circumvents that issue by exploiting the fixed size of the dot product in Equation 3. An important additional advantage of convolutional neural networks over other ANN is that the connections are relatively sparse, and many of the weights are shared between connections, both of which make learning easier.

As it is summarized in section 3.2, the bigram model only consists of convolutions, one non-linear activation function $h$ and a dot product, making it a a rather simple convolutional network with relatively few parameters.

**Audio Bigrams and 2DFTM** Finally, because of the convolution-multiplication duality of the DFT, the audio bigram model can be considered the non-Fourier domain analogue of the k-NN-based system proposed by Humphrey, who describes their system as 'effectively performing convolutional sparse coding' [6].

Effectively, the differences come down to the use of two complementary smaller kernels instead of one large one, saving in the number of weights.

Future tests will determine whether standard back-propagation-style learning can be used directly for this kind of convolutional architecture. Yet, whatever the outcome, we believe the model is an important first step towards learning convolutional kernels over variabe-length time-series, and learning powerful song descriptors specifically.

## 4 Implementation

### 4.1 PYTCH

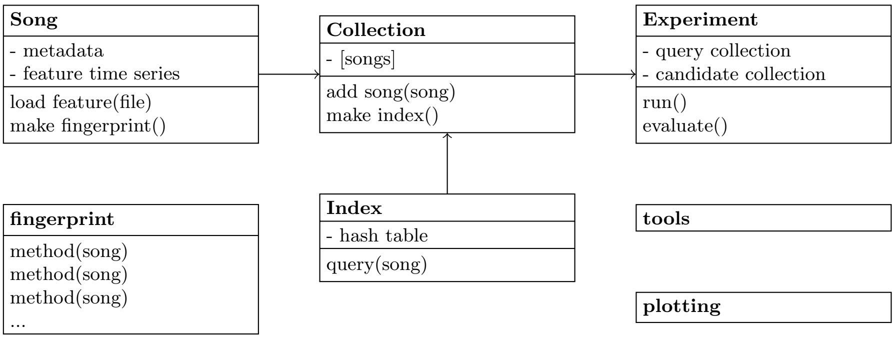

**Fig. 2.** Class and module structure of the soft audio fingerprinting toolbox PYTCH.

We provide an implementation of the above ideas in the form of PYTCH, Python toolbox for pitch-based song description available at `www.github.com/jvbalen/pytch`.

The toolbox builds on three primary classes. A class *Song* returns several representations including raw features, a fingerprint, or a set of subfingerprints. The *Collection* class contains a list of songs and returns an instance of *Index* which contains fingerprints and metadata for a collection in a datastructure that allows for efficient look-up.

Centrally in the toolbox is the *fingerprint* module, containing the fingerprinting transforms. It contains the feature transforms as layed out in section 3.2, and a set of fingerprinting methods building on these transforms. New transforms and new configurations of the existing architecture can be added here.

On top of this, there is a class *Experiment* that can be used to evaluate fingerpinting methods, a module for supplementary methods *tools*, and a *plotting* module. Figure 2 illustrates the class and module structure of the toolbox.

### 4.2 Code Example

In the following (Python) example, an experiment is run on a set of 100 candidates and queries for which the *Song* class has access to a file with chroma base features.

- Songs and their features are loaded upon construction of the *Collection* objects `queries` and `candidates`.

```
import collection as cn
import fingerprint as fp
import experiment as xp

query_ids = range(100)
candidate_ids = range(100,200)

queries = cn.Collection(query_ids)
candidates = cn.Collection(candidate_ids)
groundtruth = cn.groundtruth(queries.set, candidates.set)
```

- A fingerprinting function and its parameters are chosen and passed to the object `my_experiment` of the *Experiment* class.

```
fingerprint_type = fp.chromafp
params = {‘win’: 0.5,
 ‘normfunction’: ‘whiten’,
 ‘keyhandling’: ‘transpose’}
my_experiment = xp.Experiment(queries, candidates,
 groundtruth, fingerprint_type, params)
```

- Finally, the experiment is run. The index class is not used here as the experiment is small enough for the system to compute all pairwise (cosine) distances between the fingerprints.

```
my_experiment.run(distmetric=‘cosine’, evalmetrics=[‘map’])
print my_experiment.results
```

In most practical uses of this toolbox, it is adviced to set up a new module for one’s dataset, overriding the *Song*, *Collection* and *Experiment* constructors to point to the right files.

### 4.3 Example Experiment

We now demonstrate in an example experiment how bigram-based fingerprints can be compared, by testing a number of configurations of the system in a cover song retrieval experiment.

As a dataset, we use a subset of the *Second Hand Song* dataset[6] of 1470 cover songs. The subset contains 412 'cover groups' or *cliques*, and for each of these we have arbitrarily selected a song for the query collection. The other 1058 songs constitute the candidate collection. Since the subset is based on the second hand song dataset, we have access to pre-extracted chroma features provided by the Echo Nest. Though not ideal, as we don't know exactly how these features were computed, they make a rather practical test bed for higher-level feature development.

We implemented four bigram-based fingerprints: three kinds of chroma co-occurrence matrices (correlation, covariance, and chroma difference covariance following [7,8,12]), and one chroma landmark system, roughly following [1]. The results, with a description of the kernels $S$ and $T$, are given in Table 1.

| System | Mean average precision | Precision at 1 | Recall at 5 |
|---|---|---|---|
| Random baseline | .012 | .002 | .001 |
| Chroma correlation<br>no $S$, no $T$ | .181 | .155 | .097 |
| Chroma covariance<br>no $S$, no $T$ | .223 | .194 | .112 |
| Chroma difference covariance<br>$S$ contains $[-1, 1]^T$ | .114 | .107 | .051 |
| Chroma landmarks<br>$S$ simple peak detection<br>$T_k$ of form $[\ldots 0, 1]$ | .367 | .340 | .189 |

**Table 1.** Results table for the example cover song experiment. Chroma landmarks outperform other bigram-type fingerprints.

The chroma landmark strategy was optimized over a small number of parameters: $T_k$ was settled on a set of length-$k$ arrays where $k = 1 \ldots 16$. The best length of the peak-detecting matrix $S$ for the system was found to be 32. Only one peak detection matrix was used.

As can be seen from the table, the chroma landmark system outperforms the other systems. We believe this supports the hypothesis that, starting from the kernels $S$ and $T$ that describe this transform, a more powerful representation can be learned. In future work, the above table will be extended with the results for more bigram configurations, and with an evaluation of the learned system proposed in Section 3.3.

[6] `http://labrosa.ee.columbia.edu/millionsong/secondhand`

## 5 Conclusions and Future Work

We have reviewed a selection of 'soft' audio fingerprinting methods, and described a fingerprinting model that allows to see these methods as variations of the *audio bigram* paradigm. The audio bigram model measures co-occurrence of prespecified salient events in a multidimensional time series. We have presented an exploration of the computational architecture of the model and showed that can essentially be implemented as a particular type of convolutional neural network. The model can therefore be optimised for specific retrieval tasks using supervised learning. Finally, we have introduced our implementation of the model, PYTCH.

As future work, we plan a more extensive evaluation of some of the existing algorithms the system is capable of approximating. Standard datasets like the *covers80* dataset can be used to compare results to the existing benchmarks. If the results are close to what the original authors have found, PYTCH may be used to do a comparative evaluation that may include some variants of the model that have not previously been proposed.

We also intend to study the extent to which the convolutional network implementation of the model can be trained, and what kind of variants of the models this would produce. This can be done most easily using the *Second Hand Song* dataset, because a rather large number of train and test data will be required.

Finally, we would like to explore whether higher-order n-grams can be constructed using the same method. A rather straightforward option is to replace the matrix product in Equation 3 with a novel matrix product that takes three or four or more matrices and computes a single result, e.g. for trigrams:

$$P(X, Y, Z)(i, j, k) = \sum_n X(n, i)\, Y(n, j)\, Z(n, k).$$

We believe n-grams could prove useful in adapting more techniques from natural language processing and text retrieval, such as document frequency weighting and topic modeling, to audio data.

## References

1. Bertin-Mahieux, T., Ellis, D.P.W.: Large-scale cover song recognition using hashed chroma landmarks. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics pp. 10–13 (2011)
2. Bertin-Mahieux, T., Ellis, D.P.W.: Large-Scale Cover Song Recognition Using The 2d Fourier Transform Magnitude. In: Proc. International Society for Music Information Retrieval Conference (ISMIR). pp. 2–7 (2012)
3. Fenet, S., Gaël, R., Grenier, Y.: A Scalable Audio Fingerprint Method with Robustness to Pitch-shifting. In: Proceedings of the 12th International Conference on Music Information Retrieval Conference (ISMIR) (2011)

4. Grosche, P., Müller, M., Serrà, J.: Audio Content-based Music Retrieval. In: Müller, M., Goto, M., Schedl, M. (eds.) Multimodal Music Processing. Dagstuhl Publishing (2012)
5. Haitsma, J., Kalker, T., Oostveen, J.: Robust audio hashing for content identification. In: International Workshop on Content-Based Multimedia Indexing. pp. 117–124 (2001)
6. Humphrey, E., Nieto, O., Bello, J.: Data Driven and Discriminative Projections for Large-Scale Cover Song Identification. In: Internation Society for Music Information Retrieval Conference (ISMIR) (2013)
7. Kim, S., Narayanan, S.: Dynamic chroma feature vectors with applications to cover song identification. In: 2008 IEEE 10th Workshop on Multimedia Signal Processing. pp. 984–987 (2008)
8. Kim, S., Unal, E., Narayanan, S.: Music fingerprint extraction for classical music cover song identification (2008)
9. Rodriguez Zivic, P.H., Shifres, F., Cecchi, G.a.: Perceptual basis of evolving Western musical styles. Proceedings of the National Academy of Sciences of the United States of America 110(24), 10034–8 (Jun 2013)
10. Six, J., Leman, M.: Panako: a scalable acoustic fingerprinting system handling time-scale and pitch modification. In: Proc. International Society for Music Information Retrieval Conference (ISMIR) (2014)
11. Temperley, D., Marvin, E.W.: Pitch Class Distribution and the Identification of Key. Music Perception pp. 193–212 (2008)
12. Van Balen, J., Bountouridis, D., Wiering, F., Veltkamp, R.: Cognition-inspired Descriptors for Scalable Cover Song Retrieval. International Society for Music Information Retrieval Conference (ISMIR) pp. 379–384 (2014)
13. Van Balen, J., Serrà, J., Haro, M.: Automatic Identification of Samples in Hip Hop Music. In: Int. Symp. on Computer Music Modeling and Retrieval (CMMR) (2012)
14. Wang, A.L.C.: An industrial strength audio search algorithm. In: Proc Int Society Music Information Retrieval Conf (ISMIR) (2003)