

# AUDIOPRINT: AN EFFICIENT AUDIO FINGERPRINT SYSTEM BASED ON A NOVEL COST-LESS SYNCHRONIZATION SCHEME

*Mathieu Ramona, Geoffroy Peeters*

Ircam (Sound Analysis/Synthesis Team) - CNRS  
1, pl. Igor Stravinsky - 75004 Paris - France  
mathieu.ramona, geoffroy.peeters@ircam.fr

## ABSTRACT

This paper presents the latest improvements on AudioPrint: the IRCAM audio fingerprint system. Cosine filters are introduced in the short-term spectral analysis, in order to compensate the effect of pitch shifting, and a simple solution is proposed for the determination of the frame positions, robust to audio degradations, with nearly no additional cost. We then show that both contributions significantly improve the AudioPrint system, with evaluations both on a free corpus, made publicly available, and a real-world corpus of broadcast radio streams.

**Index Terms**— AudioPrint, Audio fingerprinting, Pitch shifting, Synchronization.

## 1. INTRODUCTION

The IRCAM AudioPrint technology is an audio fingerprint system that aims at detecting occurrences of known audio samples (or items) in an unknown audio sample or stream. Audio fingerprint systems are generally composed of two parts : the design of compact codes (the so-called *fingerprints*) that describe the acoustical properties of the signal, while being robust to common audio alterations, and a process to search the codes extracted from a signal, in a database.

The field is covered by many industrial actors, among which Philips [1] proposes a very compact representation (32 bits) of sub-band energy differences, combined with an exact match search in a hash table. The Shazam system [2] is based on numerous compact key signatures representing peak pairs in the spectrogram. The *AudioID* technology developed by Fraunhofer [3], is built on a classical pattern classification framework, using a standard Nearest Neighbor rule on MPEG-7 descriptors, coded through Vector Quantization.

More recent academic contributions include the improvement of the Shazam method with the use of the constant Q transform [4], a computer-vision approach on the spectrogram [5], particularly robust to time scale modification, and a study on search strategies for binary audio fingerprints [6].

We provide here two improvements on the fingerprint design of AudioPrint. The first one is the introduction of cosine filters to increase robustness to pitch shifting. The second one is a synchronization scheme that implies almost no additional cost, and allows to determine frame positions that are robust to audio alterations. A free corpus, called *Sync-Occur*, containing pairs of real-world aligned audio samples is also introduced here, and used for the evaluation of the fingerprint code. This study is concluded by an evaluation on a real-world broadcast corpus that demonstrates the reliability of the updated system.

The article is structured as follows : the base-line AudioPrint system will be presented in section 2, which also includes a brief presentation of the search strategy in section 2.2. The cosine filters and the synchronization scheme will then be exposed in section 3, followed by an evaluation, both on the new corpus and a real-world corpus, in section 4. A brief conclusion in section 5 will sum up our contributions and perspectives for future work.

## 2. THE AUDIOPRINT SYSTEM

The AudioPrint system is based on a fingerprint design and a code search process, that are briefly presented here. More details can be found in [7].

### 2.1. Double-nested Fourier Transform

The key idea that lies behind the design of the Ircam fingerprint is to directly model in a single code the evolution of the audio characteristics over time. While most technologies, such as Shazam [2] or Philips [1], rely on a large amount of codes, modeling instantaneous audio properties over time, AudioPrint uses a rather large temporal scope (a few seconds) for computing each fingerprint. The double-nested FT consists in estimating the evolution of spectral band energies.

Let us consider an audio signal  $x(n)$ , sampled at  $f_s$ . The signal is first analyzed through a Short Time Fourier Transform with a Blackman window, on  $L_S = 100$  ms frames with a  $H_S = 25$  ms hop size. A set of  $D_S = 6$  short-term spectral bands are defined (following the Bark scale [8]), around 1000

---

This work was partly supported by the Quaero Program funded by Oseo French State agency for innovation.

Hz. The energy of each spectral band is then computed for each frame and expressed in some scale, in order to reproduce the human loudness perception. This is also justified by the fact that the sone scale is close to a log scale. A log-scaled spectral energy separates any filter contribution into a constant offset, which makes our fingerprint robust to common linear filters, such as gain or equalization.  $B_k(m)$  denotes the value of the  $k^{\text{th}}$  spectral band sone-scaled energy for the short-term frame  $m$ .

A second STFT is then performed over time  $m$  for each short-term band  $k$ , with a rectangular window, on  $L_L = 2$  s frames with a  $H_L = 0.5$  s hop size. Each long-term frame uses about 80 short-term frames for the spectra estimation. The set of short-term values is normalized for each long-term FFT computation. Another set of  $D_L = 6$  long-term spectral bands are defined on the long-term spectra, linearly spaced around 2 Hz. The energy of each long-term band  $\ell$  is computed, for all short-term bands  $k$ . The set of  $D_S \times D_L = 36$  spectral energies defines the AudioPrint code.

## 2.2. Code search

The code search in AudioPrint is straightforward. First step consists in retrieving, for each processed code, the  $k$  nearest neighbors in the database (efficient nearest neighbor search is considered an auxiliary problem, an exact search is used here). Each neighbor corresponds to a {item index, time stamp} pair. Items that appear several times in a sliding window of  $P$  codes are candidates.

A post-processing step then validates the candidate items if at least  $Q$  frames (with  $Q \leq P$ ) from the window contain nearest neighbors that have their item timestamp differences correlated with the frame timestamp differences. Figure 1 illustrates an example of (a) validated detection and (b) discarded candidate (where the colored squares represent the nearest neighbors pointing to a same item).

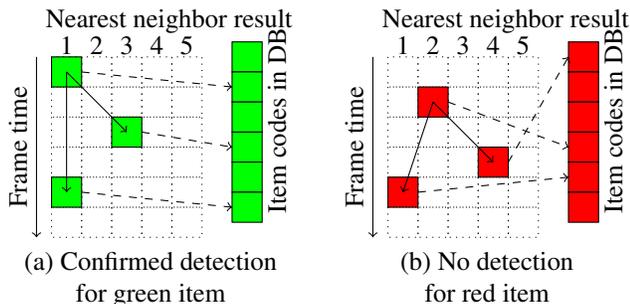


Fig. 1. Illustration of the search post-processing.

## 3. IMPROVEMENTS

### 3.1. Cosine filters

In the baseline version of AudioPrint, short-term band energies are computed as sums of squared amplitudes of the spec-

trum. This is equivalent to applying rectangular filters. However, in the presence of pitch shifting (a common alteration in radio broadcast), a spectral peak originally located in a band might end up in a neighbor one, and then significantly alter both band energies. This is partly corrected by using cosine filters instead of rectangular. Indeed, the smooth shape allows for a continuous increase/decrease of the filter energy, instead of the staircase discontinuity induced by the original filters. The cosine filters are designed to reach 0 at the middle of neighboring bands. Figure 2 shows the original and cosine filters respectively in red and dashed blue.

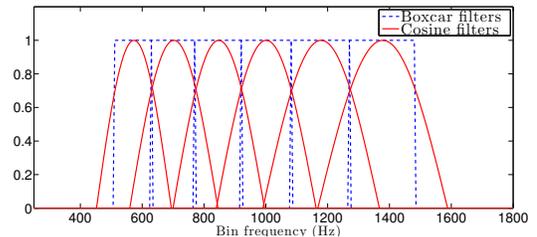


Fig. 2. Comparison of the short-term bank filter profiles.

### 3.2. New frame synchronization scheme

**Spectral energy** The second improvement provided here concerns the determination of the long frame positions in the audio stream. The baseline system is based on regular frames, with a step size of  $H_L = 0.5$  s. However, considering that the occurrence of an item in the stream is uniformly distributed, the average difference between a frame time and the closest code in the database is  $H_L/4 = 125$  ms.

A previous article [7] proposed a scheme to reach a better synchronization between the codes in the database and in the stream. However, the determination of the frame position was based on an onset detection process undertaken by an external tool that implies an added computational cost (the audio analysis/transformation tool SuperVP [9] was used in our case).

In order to reach synchronization, frame positions must be estimated from the audio signal and robust to noise and alterations. Detecting onsets can be a complex and costly process, covered by many contributions in the literature [10]. However, we propose here to use only the amplitude energy of the spectrum on the AudioPrint short-term band, which is basically the sum (or, equivalently, the mean) of the short-term filter band energies introduced in Section 2.1 :

$$E(m) = \frac{1}{D_S} \sum_{k=1}^{D_S} B_k(m) \quad (1)$$

**Energy maxima detection** The maxima peak picking goes as follow. First,  $E(m)$  is normalized (subtraction of median offset and division by standard deviation) on a sliding window of 20 frames, in order to have homogeneous statistical

properties. Then the “plateau” signal  $P(m)$  is defined, such that each value is the maximum value of  $E(m)$  in a window of 7 frames around  $m$ . Values  $m$  for which  $P$  and  $E$  are equal are kept as local maxima, as shown in figure 3.

**Anchorization of regular frames** In our previous contribution [7], the detected onsets were all kept as frame positions. However, using all detections implies an irregular distribution of the frames, possibly with areas not covered by any frame, and others with many frames overlapping.

Detected peaks are rather used here as anchors to tune the regular offsets. All peaks are sequentially examined by descending order of amplitude. If the nearest regular frame offset hasn’t been updated yet, it is assigned with the peak position. This way, predominant peaks have priority over neighboring peaks. Figure 3 illustrates this process.

This “anchoring” process has the benefit of keeping the same number of frames, and of guaranteeing a global homogeneity of their distribution over time.

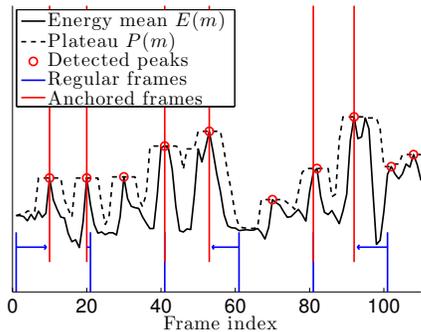


Fig. 3. Regular frames anchoring process on energy peaks.

**Computational cost** Table 1 shows the compared computation time (in milliseconds) of the method with and without synchronization, on an audio sample of 5 minutes. It appears clearly that the energy-based synchronization (**Energy** in the table) brings almost no overload, when compared to the regular frame baseline (**Regular**), while the onset-based (**Onset**) needs twice as much time.

Regular	Energy	Onset
1438 ms	1497 ms	3297 ms

Table 1. Compared computational costs.

## 4. EVALUATION

### 4.1. Code search evaluation

#### 4.1.1. The SyncOccur corpus

We provide along with this paper a corpus automatically generated from broadcast radio streams. A method was previously presented [11] to check audio fingerprint ground-truth

and automatically align with great precision the original item and its annotated occurrence. Since music tracks are usually time-stretched when broadcasted, the alignment implies the estimation of the time-scale factor between item and occurrence, and of the exact position in the stream.

This method was used to generate a collection of 10 000 item/occurrence audio pairs of 20 seconds that make the so-called *SyncOccur* corpus. All occurrences were cut and “unstretched” in order to be perfectly synchronized with the original items. The synchronicity can be manually verified by playing both audio samples simultaneously. The full corpus is detailed and available at <http://www.mathieuramona.com/wp/data/syncoccur-corpus>.

We hope the *SyncOccur* corpus might become a useful tool for the audio fingerprint community, for two reasons. First, both the item and the occurrence come from radio broadcast, hence they show typical real-world distortions, whereas most corpuses used in the literature are distorted with artificial alterations, as discussed in [12]. Second, the alignment allows to study and quantify the effect of additional alterations (additive noise, time stretching, time shifting, ...) on the fingerprint codes, as shown in the next section.

#### 4.1.2. Cosine filters for pitch shifting

In order to evaluate the performance of a fingerprint design, 5 000 synchronized code pairs were extracted from the *SyncOccur* corpus. The codes were computed under different levels of added pitch shift on one of the audio samples. Then the pitch-shifted code is searched among a database containing the corresponding one and  $n = 50000$  other random codes (computed from random audio excerpts). Figure 4 shows how often the expected code is not found among the  $k$  nearest neighbors in the database.

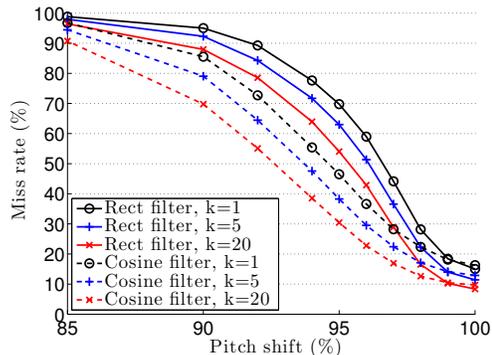


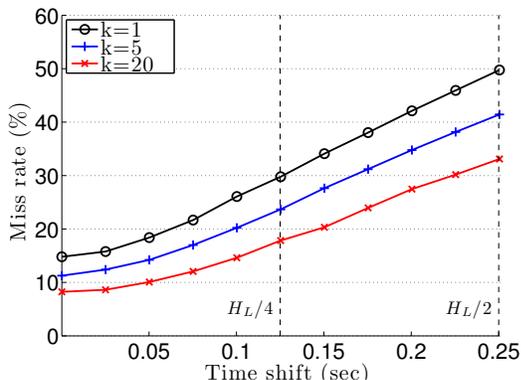
Fig. 4. Miss rate when searching a pitch shifted code among a database containing the expected plus 50000 random codes.

Expectedly, the miss rate dramatically increases when the pitch shift ratio decreases. However, consistently for any value of  $k$ , the cosine filter fingerprint is much less affected than the original fingerprint, based on rectangular filters. At

94% pitch shift, the miss rate decreases from 64.0% to 38.5%, for  $k = 20$ , representing about 40% relative gain.

### 4.1.3. Frame synchronization

The previous evaluation is also applied to the time shifting alteration, i.e. slightly shifting the frame position on the occurrences in order to study the effects on the code search. Figure 5 shows the miss rate for time shifts varying between 0 and  $H_L/2 = 0.5$  s, for  $k = 1, 5$  and 20 nearest neighbors search. Indeed, for  $k = 1$ , the miss rate raises from 14.8% with no time shift, to 30% with a time shift equal to the average distance between frames ( $H_L/4 = 125$  ms, as explained in section 3.2), and up to 50% with the maximum distance ( $H_L/2 = 0.25$  s). This shows that synchronizing frame positions is a major issue for frame-based audio fingerprint.

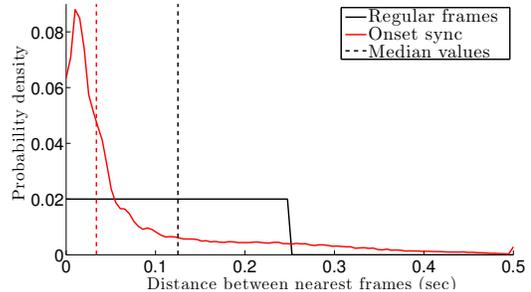


**Fig. 5.** Miss rate when searching a time shifted code among a database containing the expected plus 50000 random codes.

Figure 6 shows the distribution of the distance between the pairs of nearest item/occurrence frames on the whole SyncOccur corpus. The red line shows the empirical distribution observed with the proposed synchronization scheme, while the black line shows the expected distribution when using regular frames (uniform distribution bounded by  $H_L/2$ ). It clearly shows that the proposed algorithm concentrates most distances below 50 ms (note that any precision below the short windows hop size of  $H_S = 25$  ms can not be reached anyway). This is confirmed by the reduction of the median distance (dashed lines for both distributions): 125 ms for regular frames and only 33 ms with the synchronization. According to Figure 5, both median distances respectively correspond to a miss rate of 30% and 18%, for  $k = 1$ .

## 4.2. Real-world evaluation

We conclude this evaluation by a study on a real-world corpus, that was used in a previous publication [7]. It consists of a collection of 10 whole days (i.e. 240 hours) of broadcast radio stream encoded in WMA at a very low bitrate (about 10 kbps), and was provided by the media monitoring company



**Fig. 6.** Distribution of the distance between nearest frames from item and occurrence (100 bins, normalized to unit sum).

Yacast, as a partner of the Quaero project. The reference items are excerpts of 30 s of the same quality. The corpus contains around 2000 occurrences in the streams, searched among a database of 1000 training items.

Method	FR %	FA %	RES %
Shazam	15.4	0.1	84.6
Philips	10.5	0.0	89.5
Baseline	5.9	0.1	94.0
Cosine filters	3.8	0.0	96.2
Cosine + Onset sync	2.9	0.0	97.1
Cosine + Energy sync	2.0	0.0	<b>98.0</b>

**Table 2.** Compared results on a real-world corpus.

Table 2 shows the results of AudioPrint with the improvements presented in this paper. The compared results of our own implementation of the methods published by Philips [1] and Shazam [2], first show that AudioPrint performs better than classical state-of-the-art systems. We can then note that the cosine filters brings an important improvement (from 94.0% to 96.2%), that is completed with the use of the new energy synchronization scheme (98.0% score). Note that most of the improvement, when compared to the previous algorithm based on onset detection (97.1% with cosine filters), lies in the new anchoring process that ensures a global regularity of the frames. Finally this improvement comes with a drastic reduction of the computational cost (Table 1).

## 5. CONCLUSION

We have presented latest updates to the AudioPrint technology, that significantly improve its performance, with almost no extra computational load. In particular, the new frame synchronization scheme, based on the so-called "anchoring" process, provides a very efficient way to determine frame positions that are robust to audio alterations. We also provided a new corpus of synchronized audio sample pairs, that allows comprehensive studies on the fingerprint design. Future works will focus on improving robustness with respect to additional noise and time stretching.

## 6. REFERENCES

- [1] Jaap Haitsma and Ton Kalker, “A highly robust audio fingerprinting system,” in *Proc. ISMIR '02*, October 13–17 2002.
- [2] Avery Li-Chun Wang, “An industrial-strength audio search algorithm,” in *Proc. ISMIR '03*, 2003.
- [3] Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Bernhard Fröba, Throsten Kastner, and Markus Cremer, “Content-based identification of audio material using MPEG-7 low level description,” in *Proc. ISMIR '01*, 2001.
- [4] Sébastien Fenet, Gaël Richard, and Yves Grenier, “A scalable audio fingerprint method with robustness to pitch-shifting,” in *Proc. ISMIR '11*, Miami, Florida, USA, October 2011, pp. 121–126.
- [5] Bilei Zhu, Wei Li, Zhurong Wang, and Xiangyang Xue, “A novel audio fingerprinting method robust to time scale modification and pitch shifting,” in *Proceedings of the ACM International Conference on Multimedia*, Firenze, Italy, October 25–29 2010, pp. 987–990.
- [6] Kimberly Moravec and Ingemar J. Cox, “A comparison of extended fingerprint hashing and locality sensitive hashing for binary audio fingerprints,” in *Proc. ICMR '11*, April 17–20 2011.
- [7] Mathieu Ramona and Geoffroy Peeters, “Audio identification based on spectral modeling of bark-bands energy and synchronisation through onset detection,” in *Proc. ICASSP*, May 22–27 2011, pp. 477–480.
- [8] E. Zwicker and E. Terhardt, “Analytical expression for critical-band rate and critical bandwidth as a function of frequency,” *Journal of the Acoustical Society of America*, vol. 68, pp. 1523–1525, 1980.
- [9] Axel Röbel, “Onset detection in polyphonic signals by means of transient peak classification,” in *ISMIR/MIREX*, Vancouver, 2006.
- [10] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler, “A tutorial on onset detection in music signals,” *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, September 2005.
- [11] Mathieu Ramona and Geoffroy Peeters, “Automatic alignment of audio occurrences: application to the verification and synchronization of audio fingerprinting annotation,” in *Proc. DAFX '11*, September 2011, pp. 429–436.
- [12] Mathieu Ramona, Sébastien Fenet, Raphaël Blouet, Hervé Bredin, Thomas Fillon, and Geoffroy Peeters, “A public audio identification evaluation framework for broadcast monitoring,” *Applied Artificial Intelligence: Special Issue on Event Recognition*, pp. 119–136, February 2012.