HAL
open science

Audio Fingerprint Identification by Approximate String Matching

Jerome Lebossé, Luc Brun

To cite this version:

Jerome Lebossé, Luc Brun. Audio Fingerprint Identification by Approximate String Matching. Proceedings of ISMIR 2007, Sep 2007, Vienna, Austria. pp.143-144. hal-00824741

HAL Id: hal-00824741

https://hal.science/hal-00824741v1

Submitted on 22 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AUDIO FINGERPRINT IDENTIFICATION BY APPROXIMATE STRING MATCHING

Jerome Lebosse
France Telecom R&D
42 rue des coutures
14000 Caen, FRANCE
jerome.lebosse@orange-ftgroup.com

Luc Brun
GREYC UMR 6072
6 boulevard du Marchal Juin
14050 Caen, FRANCE
luc.brun@greyc.ensicaen.fr

ABSTRACT

An audio fingerprint is a small digest of an audio file which allows to identify it among a database of candidates. This paper first presents a fingerprint extraction algorithm. The identification task is performed by a new identification scheme which combines string matching algorithms and q-grams filtration.

1 ROBUST FINGERPRINT EXTRACTION METHOD

Given an input signal sample eventually corrupted, its fingerprint allows to quickly retrieve the original file among a database of known audio files when it exists. Since an input audio sample should be identified, audio fingerprints should be composed of elementary keys (called *sub-fingerprints*) computed continuously along the signal.

Fingerprints construction schemes are usually based on a sequence of overlapping windows on which sub-fingerprint values are computed [3]. This method called *enframing* insures a fixed detection rate of sub-fingerprints but is sensitive to cropping or shifting operations (Section 3). On the other hand, onset methods [1] are less sensitive to cuts alterations. However, these methods don't insure any minimal detection rate of frames per second which may allow to bound the time required for the identification of an audio file.

Our idea is to combine the respective advantages of enframing and onsets methods by selecting a small time interval (I_{emax}) with maximal energy within a larger one (I_o) (Figure 1). Then, this signal segmentation process allows to quickly synchronize two fingerprints despite eventual shift operations. We then define the sub-fingerprint values as the lengths (in ms) between two consecutive I_{emax}. More details about our fingerprint extraction algorithm may be found in [4].

2 FINGERPRINT RECOGNITION

Several recognition techniques [3] may be used to compare an input fingerprint to a database. Within our framework, string matching methods provide a natural framework to account for shifted or wrongly detected I_{emax} intervals between two fingerprints. Highly similar sequences between two strings are identified by computing an edit distance. Using a finite alphabet, the edit distance is often scored using a simple metric where a match is awarded a positive score while a mismatch is awarded a negative score. However within the fingerprint recognition framework, two co-derivative fingerprints should share long sequences of common symbols with few mismatches. We thus designed a scoring function which increases non linearly when a sequence of matches is encountered. This scoring function is defined as:

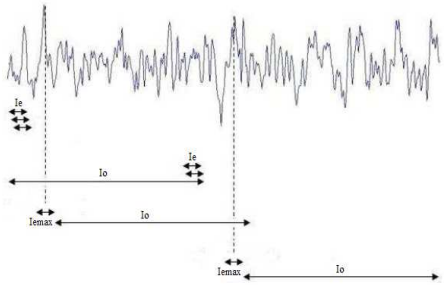

Figure 1. Audio Segmentation Method

$$S(i,j) = \begin{cases} \alpha S(i-1,j-1) + \beta & \text{If } s_i = s_j \\ \frac{1}{\gamma} max \begin{pmatrix} 0, \\ S(i,j-1), \\ S(i-1,j) \end{pmatrix} - \beta & \text{otherwise} \end{cases} \quad (1)$$

The constants α, β, γ are determined experimentally (Section 3) and satisfy $1 < \gamma < \alpha$ in order to decrease the score when mismatches are encountered at a lower step than it increases during a sequence of matching symbols.

A comparison of an input fingerprint with all the database fingerprints may be performed using string matching methods based on our scoring function (equation 1). This comparison is usually speed up using a q-grams filtering approach [2]. The basic idea of our filtering method consists to weight each common q-gram between two strings by a score defined by equation 1.

More formally, let us denote by $Q_{D,I}$ a common q-gram between the input fingerprint I and a database's fin-

© 2007 Austrian Computer Society (OCG).

gerprint D. Given a maximal string's length m the score of $Q_{D,I}$ is defined as:

$$score(Q_{D,I}) = \sum_{i=1}^{p} \sum_{j=1}^{q} S(I[I_i, I_i + m], D[D_j, D_j + m]) \quad (2)$$

where $S(I[I_i, I_i + m], D[D_j, D_j + m])$ denotes our scoring function computed on the two sub-strings of length m starting at indexes I_i and D_j. Given equation 2, let us denote by $Q_{D,I} \subset D$ the set of common q-grams between D and I. The score of a database's fingerprint is thus defined as the sum of scores of its common q-grams with the input:

$$score(I, D) = \sum_{Q_{D,I} \subset D} score(Q_{D,I}) \quad (3)$$

Experiments (Section 3) show that the co-derivative fingerprint, when it exists within the database, is always the one getting the higher score. We can thus retrieve a co-derivative fingerprint when it exists within the database. However, an identification method should also be able to decide if an input fingerprint has no co-derivative within the database.

Since our filtering process always ranks first the co-derivative fingerprint when it exists, our decision rule has only to confirm the existence within the database of the fingerprint with higest score. The fact that our filtering process does not provide a valid decision rule is mainly due to the small value of m considered for each common q-gram in equation 3. We thus designed a decision criterion based on the score between the input fingerprint and the database fingerprint ranked first by our filtering process process. Both fingerprints are compared on larger sub-strings of length $M > m$ and at the two positions (i_{max}, j_{max}) such that $I[i_{max}, i_{max}+q] = D[j_{max}, j_{max}+q])$ and $S(I[i_{max}, i_{max} + m] = D[j_{max}, j_{max} + m]))$ is maximal amount all the scores computed between I and D using equation 2. Our input fingerprint's score is thus defined as:

$$score(I) = score(I[i_{max}, i_{max} + M], D[j_{max}, j_{max} + M]) \quad (4)$$

3 RECOGNITION EXPERIMENTS

Our database contains the fingerprints of 350 songs of approximately 4 minutes each. In our experiments, a sample of 5 seconds of each database's audio file has been extracted randomly and compressed at 128kbps. The fingerprints of these samples have then been compared to the ones of the database.

The minimum size of the q-grams for our filtration procedure has been fixed to 5 in these experiments. The values of α,γ and β (equation 1) have been respectively set to: $\alpha = 1.5$; $\gamma = 1.1$ and $\beta = 20$.

Let us first consider the results obtained using equation 3. The first thee columns of Table 2 untitled `Filtering scores` shows min, max and mean scores according to equation 3 of the three database fingerprints with highest rank. In all our experiments, the database fingerprint corresponding to the input has always been ranked first. The score of the second fingerprint corresponds to the score that would be obtained if the fingerprint of the input audio file was removed from the database. As shown by the cells (Min,1st) and (Max, 2nd), one input fingerprint not present within the database may obtain a higher score than another input fingerprint whose co-derivative belong to the database. This negative property of equation 3 does not allow to decide if a fingerprint is present within the database.

	Filtering scores			Final Scores		
Scores	1st	2nd	3rd	1st	2nd	3rd
Min	**14878**	0	0	**100000**	0	0
Mean	11.10^5	240.72	143.74	3.10^{15}	16	3
Max	4.10^6	**20.**10^3	10.10^3	5.10^{17}	**2800**	590

Figure 2. Filtering and Final scores

The last three columns of Table 2 show the scores computed according to equation 4 of the three database's fingerprint selected in the previous experiment. As shown by the cells (Min,1st) and (Max, 2nd) of the last three columns, the highest score (according to equation 4) of a fingerprint not present within the database is much lower than the lowest score obtained by an input fingerprint whose co-derivative is present within the database. In these experiments, any threshold between these two values allows to decide if a fingerprint is present within the database.

4 CONCLUSIONS

We have presented in this paper an audio fingerprint identification method. A first filtering based on q grams and string matching algorithms allows to retrieve the closest database fingerprint from the input. A final score based on a new scoring function is attached to the fingerprint returned by the database filtering. This final score allows to check if an input fingerprint has a co-derivative within the database. In future work we plan to improve the indexation scheme of our database according to the use of q-grams performed within our filtering step.

5 REFERENCES

[1] S. Hainsworth and M. Macleod, "Onset detection in musical audio signals," in *Proc. of ICMC*, 2003.

[2] P. Jokinen and E. Ukkonen, "Two algorithms for approximate string matching in static texts," in *Proc. of MFCS*, pp. 240-248, 1991.

[3] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. of ISMIR*, pp. 144-148, 2002.

[4] J. Lebossé and L. Brun and J.C. Pailles, "A robust audio fingerprint extraction algorithm," in *Proc. of SPPRA*, Innsbruck(Austria), February 2007.