A ROBUST AUDIO FINGERPRINT'S BASED IDENTIFICATION METHOD

ABSTRACT

An audio fingerprint is a small digest of an audio file computed from its main perceptual properties. Like human fingerprints, audio fingerprints allow to identify an audio file among a set of candidates but does not allow to retrieve any other characteristics of the files. Applications of audio fingerprint include audio monitoring on broadcast chanels, filtering peer to peer networks, meta data restoration in large audio library and the protection of author's copyrights within a Digital Right Management(DRM) system. We propose in this paper a new fingerprint extraction algorithm based on a new audio segmentation method. A scoring function based on q-grams is used to determine if an input signal is a derivated version of a fingerprint stored in the database. A rule based on this scoring function allows to either recover the original input file or to decide that no fingerprint belonging to the database correspond to the signal. The proposed method is robust against compression and time shifting alterations of audio files.

KEY WORDS
audio fingerprint, segmentation, q-grams, scoring function, identification.

1 Introduction

Audio fingerprint [9, 11, 2] aims at defining a small signature (the fingerprint) from a content based on its perceptual properties. Audio fingerprints share several properties with their human counter parts. Firstly, an audio fingerprint allows to identify an audio file from a small amount of data. Secondly, as with human fingerprints no properties of an audio file may be readily derived from its fingerprint. Applications of fingerprints include audio monitoring on broadcast channels, filetering on peer to peer networks, meta data check or restoration on large audio library and digital rights managements.

In order to correctly identify an audio file from its perceptual properties, fingerprint methods have to be robust against alterations (compression, cuts,...). Two audio files, one defined from the other using such alterations, are called *co-derivatives* [1, 8]. By extension, we will say that the fingerprints asociated to these audio files are co-derivative. Moreover, a fingerprint system should be able to recover a file from a short sample in a short time interval. The computational cost of a fingerprint system should thus be low and the size of the fingerprint should be small in order to be intergrated into low cost terminalsxs. Moreover, in order to recover a file from any sample, the fingerprint should be composed of elementary keys (called sub-fingerprints) based on small parts of the signal. The sub-fingerprints are then computed either continuously along the signal or at a rate which provides a sufficient number of sub-fingerprints to characterize a file from a short sample.

A fingerprint system consists of two components: a method which extracts the fingerprint and an identification method which matches fingerprints. In this study, we propose a new approach for both of these two steps of an audio fingerprint based identification process. We first present in section 2 alternative approaches. Then, we describe our method in section 3. The robustness of the proposed fingerprint is evaluated in Section 4 through two experiments. Then, the efficiency of the scoring function used in the contents identification task is shown in Section 5.

2 State of the art

2.1 Fingerprint Computation

As mentioned in Section 1, a fingerprint is usually made of a sequence of consecutive keys in order to identify any part of the signal. The first step of a fingerprint algorithm consists thus to extract from the signal a sequence of small intervals and to associate to each interval a value (called the sub-fingerprint) which characterises it.

A usual method [11, 9] consists to decompose the signal into a sequence of overlapping intervals called frames, of a few milliseconds. Enframing the audio stream allows to consider each frame as a relatively stationary sound. The frames are then weighted by a Hamming window in order to minimize the discontinuities at the beginning and at the end of each frame. A frame overlap which depends on the frame's size, is used to reduce the shifting effects.

Several methods combine the enframing decomposition with a design of the sub-fingerprint based on the FFT of the signal. However, a cut of the signal or the concatenation of a silence at its beginning is roughly equivalent to a translation. While the energy of the whole signal is preserved by such a transformation, the computed energy on each interval may be drastically changed [14]. The use of overlapping windows only reduces the influence of such cuts (Section 5.4).

Another way to segment an audio waveform is to find particular positions in the audio signal, called onsets [10, 7]. Typical onset detection schemes decompose the signal into frames and associate some perceptual quantities to each frame. The onsets are then detected from the signal encoding the distance between the feature vectors of adjacent frames. The main drawback of onset methods within the audio identification framework is the fact that the number of onsets detected in a short time interval (say 1s) is unpredictable and is usually too low to provide an efficient characterization of the signal. Therefore a fingerprint algorithm based on an onset approach may delay the identification for an unpredictable period until it has collected a sufficient number of sub-fingerprint. This unpredictability is a major drawback for fingerprinting systems which should identify a file in a fixed and usually short time interval. Enframing is thus generally used to decompose the signal into small intervals in audio identification systems.

Once the signal is divided into intervals, discriminant features have to be associated to each interval. The sequence of features computed along the waveform defines the audio fingerprint. Kurth [11] proposes a sub-fingerprint design based on the global energy of each interval. However, this method does not capture enough information from the spectrum to provide a reliable fingerprint indexation scheme. Additional information about each interval may be obtained by considering the FFT of each interval and the energy of each of its frequency bin [5]. Different heuristics have been proposed based on this basic idea. For example, Johnson and Woodland associate to each frame the mel-frequency PLP cepstral coefficients of its spectrum together with the derivative of these coefficients. Burges et al. [2] use a particular decomposition of the spectrum called the Modulated Complex Lapped Transform (MCLT).

The method of Kalker and Haitsma [9] follows the above approach and uses a decomposition of the spectrum of each frame into bands using a logarithmic spacing. The authors decompose the signal into frames of 0.37s with an overlap factor of 31/32. The sub-fingerprint of each frame is then defined as a 32 digit number computed from the decomposition of the spectrum. The sequence of bits of each frame is defined from the sign of the energy differences computed both between two consecutive bands of a same frame and between two consecutive frames. More precisely, let us define $EB(n, m)$ as the energy of the m^{th} band within the n^{th} frame and $\Delta EB(n, m) = EB(n, m) - EB(n, m + 1)$ as the difference of the energy of two successive band within a same frame. The value of the m^{th} bit of the n^{th} frame ($F(n, m)$) is then defined as:

$$F(n,m) = \begin{cases} 1 & if \quad \Delta EB(n,m) - \Delta EB(n-1,m) \geq 0 \\ 0 & if \quad \Delta EB(n,m) - \Delta EB(n-1,m) \leq 0 \end{cases}$$

The sub fingerprint of each frame n, is defined as the concatenation of the 32 bits $F(n, m), m \in \{1, \ldots, 32\}$. The audio fingerprint is then defined as the concatenation of its sequence of sub-fingerprints.

2.2 Fingerprint Recognition

Given a fingerprint method, a database made of fingerprints of known audio file may be built. The fingerprint recognition task consists to compare an input fingerprint to all the fingerprints of the database and to decide if it can be considered as a co-derivative of a known fingerprint or if it does not belong to the database. Note that, this last point makes a strong difference between indexation and Fingerprint recognition methods. Indeed, an audio indexation method retrieves audio files similar to the input file while a Fingerprint recognition method should be restricted to the retreival of co derivative signals.

The Fingerprint Recognition method of Haitsma and Kalker [9] is based on the notion of fingerprint block which corresponds to a sequence of 256 sub-fingerprints. The distance between two fingerprint blocks is defined as the sum of Hamming distances between the 256 sub-fingerprints. The recognition of a fingerprint proceed as follows: A fingerprint block is first extracted from an input signal. This block is then compared to all sequences of 256 sub-fingerprints composing the database. If one of these distances is below a treshold the input signal is considered as a co-derivative. Otherwise the input fingerprint is considered as not belonging to the database. The brute force comparison of one fingerprint block with all the fingerprints of the database would induce cumbersome computations. In order to improve the efficiency of the algorithm Haitsma and Kalker restrict the search to the blocks of the database which share at least one sub-fingerprint value with the input block. These blocks may be reteived efficiently using a hash table which associates to each sub-fingerprint value a set of candidates fingerprint blocks. Note that the proposed distance between two fingerprint blocks supposes that no sub fingerprint may be added nor removed from any of the two sequences of sub fingerprints. Such an hypothesis is valid within the Haitsma and Kalker method which uses overlapping window (Section 2.1). However, this hypothesis would not hold using sub fingerprints based on a segmentation scheme such as onset detection methods (section 2.1) where additional or missing onsets may occur.

An alternative method within the fingerprint recognition framework consists to consider the set of all possible sub-fingerprint values as an alphabet. A fingerprint may then be interpreted as a string and the distance between two fingerprints is performed using approximate string matching techniques [4, 3]. The similarity between two strings is defined as the minimum number of insertion, deletion and substitution operations needed to transform one string into the other. Each operation may be weighted in order to encode the cost of each operation within a particular framework. If a small part of the input signal is used to compute the fingerprint the recognition task consists to find the lowest editing cost between the input fingerprint and a sub-sequence of one of the database's fingerprint. If the database is made of M fingerprints of size n and if an priori threshold k is set for the edit distance such a search work in $\mathcal{O}(Mkn)$ time [4]. Due to the expected large values of both n and M within the fingerprint recognition framework such a method is usually not acceptable. The problem should thus be decomposed into two phases: The first phase called filtration finds potential matchs which are later verified by the second phase. The filtration step may be based on the computation of the common q-grams (word of length q) between two strings [3]. If two fingerprints share a sufficient amount of common q grams a potential match is detected and verified during the second phase. Note that, some authors within the music retreival framework [6, 16] avoid the second phase and consider that two strings sharing a sufficient number of q-grams should be considered as "similar". Applied to the fingerprint recognition framework [1, 8] approximate string matching techniques provide two interesting properties: Firstly, unlike the Hamming distance, the weight of a substitution allows us to account for the fact that when the difference between two sub-fingerprints is below a given threshold these sub-fingerprints should be considered as equal. Secondly, the insertion and deletion operations provide a natural tool to account for additional and missing sub-fingerprint values between two co-derivative fingerprints.

3 A Robust Fingerprint Extraction Method

As mentioned in Section 1, enframing methods insure that a sufficient number of frames is selected within an input segment. However, the selection of a sequence of contiguous frames is sensitive to random cropping or shifting operations which may be performed on the signal (Sections 2 and 5). This drawback is attenuated but not completely overcomed by the use of overlapping frames. On the other hand, segmentation methods are less sensitive to cropping or shifting operations but do not insure that a sufficient number of intervals within the signal will be selected during the limited time allowed for fingerprint recognition.

The basic idea of our method is to combine the respective advantages of enframing and segmentation methods by selecting a small time intervals within a larger one. The small interval allows the detection of characteristic parts of the signal whereas the larger interval insures a minimum selection rate of intervals.This process could be decomposed into three steps (Figure 1):

- In the first step, an interval, called the Observation Interval (I_o) is set at the beginning of the waveform. The length of this intervall is typically equal to few hundredths of seconds.
- The waveform inside I_o is analysed in the second step. We divide in this step, the interval I_o into shorter overlapping sub-intervals of a few millisecond, called Energy Intervals (I_e). The energy of each I_e interval is computed by taking the mean of the samples amplitude within the interval.
- In the third step, the interval I_e with a maximal energy (I_{emax}) within I_o is selected.

Given an I_{emax} interval, the begining of the next I_o interval is set to the end of I_{emax} (Figure 1). The distance between the centers of two I_{emax} intervals lies thus

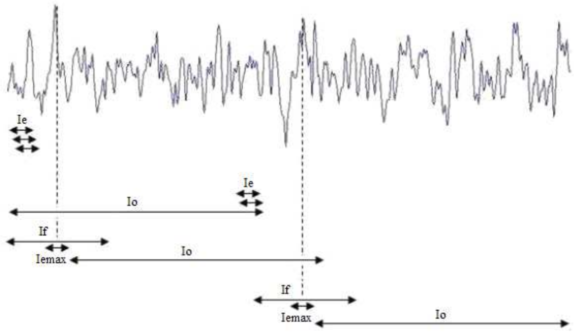

Figure 1. Audio Segmentation : The interval I_{emax} corresponds to the interval of greatest energy within I_o.

between I_{emax} and I_o-I_{emax}. This method provides a higher robustness against time shifting than the basic strategy which consists to select a sequence of consecutive I_o intervals. Indeed, if a strong peak is present within a signal this peak will be selected as I_{emax} both in the original and shifted signal. The next I_o interval will thus start at a same position both within the original signal and its shifted version. This strategy allows thus to synchronise the two fingerprints on significant peaks of the signal.

Moreover, using the basic strategy based on consecutive I_o intervals, a maximum I_e interval located at the transition between two I_o intervals would not be detected. Finally, our strategy allows to select several I_{emax} intervals located in a same I_o with near maximal energies. Using the basic strategy only one of them would be detected while the degradation of the signal may exchange the selection of two I_e interval whose energy is near the maximum. This strategy enforces thus the robustness of out method against compression.

Given a sequence of detected I_{emax} intervals a sub-fingerprint value should be associated to each interval in order to build the fingerprint. We first explored [12] a sub-fingerprint construction scheme which associates to each I_{emax} interval a sub fingerprint value using a method similar to the one of Haitsma and Kalker [9](Section 2). However, such a construction scheme of a fingerprint implies the sequential application of two processes: The segmentation step and the computation of a sub fingerpint value around each I_{emax} interval. The number of common sub-fingerprints between two co-derivative signals is thus equal to the number of sub-fingerprints which have been detected and computed without any error by *both* processes. Has shown by our experiments (Section 5) the additional step required by the sub-fingerprint computation decreases significantly the number of common sub-fingerprints between two co-derivatives signals. We thus use an approach similar to the one of Hoad [8] and define the sequence of sub-fingerprint values as the lengths (in ms) between the different detected I_{emax} intervals. The performance of our fingerprint extraction algorithm are thus the one of our segmentation step. Moreover, using the values of I_o

and I_e choosen in our experiments (Section 5) each sub-fingerprint may be stored on 14 bits instead of the 32 bits required by the method of Kalker and Haitsma.

4 Fingerprint Recognition

As mentioned in Section 2.2 approximate string matching appears to be a natural distance between fingerprints when sub-fingerprints may be added or removed between two co-derivative signals. However, the time required to compare an input signal with all the fingerprints of the database is unacceptable for common database sizes used within the fingerprint recognition framework. One common solution, used for example within the genomic framework [?] consists to filter the database using q-grams (Section 2.2). The computation of common q-grams between two strings is indeed, much more efficient then the string matching operation since it does not imply any editing operation. As schown below the probability that two strings share n_q q-grams decreases drastically for $n_q \geq 2$ when the size of the alphabet on which the fingerprints are built is large. Since the number of sub-fingerprints used within our method (i.e. the size of our alphabet) is equal to 2^{14}, we use this last property in order to avoid the computation of string edit distance. The basic idea of our Fingerprint recognition method consists to compute a score proportional to the amount of information bring by the existence of common q grams between two fingerprints.

More formally, given an alphabet of size s, let us consider random fingerprints generated by drawning independently the sub-fingerprints with an uniform distribution $\frac{1}{s}$. Given two fingerprints, we denote by X_q the random variable which corresponds to the number of common q grams between two fingerprints and n_q one realisation of X_q. Let us define $Q(n_{min}, \ldots, n_{max}) = p(X_{min} = n_{min} \& X_4 = n_4 \ldots \& X_{max} = n_{max})$ the probability that the event $X_{min} = n_{min} \& \ldots \& X_{max} = n_{max}$ occurs between the two random fingerprints. The symbol min denotes the minimal lenth of q grams considered by the score function while max is the longuest q grams common to the two fingerprints. Our aim consits to define a score function between two fingerprints which is in $\mathcal{O}(-\log(Q(n_{min}, \ldots, n_{max}))$.

Let us first consider the simpler case where only one length of q gram is considered. Nicodeme [15] has schown that the law of probability of X_q may be efficiently approximated by an urn and balls model where each of the s^q urn is associated to one q gram. Given two fingerprints of size n and m one thrown n white balls and m black balls and the number of common q-grams X_q between the two fingerprints may be efficiently modeled as the number of urns containing simultaneously white and black balls. In the same article, Nicodème shows, that the variation distance between the Poisson law of X_q and a gaussian law is always bounded. Moreover, the law of X_q converges to a gaussian when the parameters of the Poissons law modeling X_q tends to infinity. We can thus approximate the law of X_q using the Gaussian law:

$$p(X_q = n_q) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(n_q-\gamma)^2}{2\sigma^2}} \quad (1)$$

where γ and σ represent respectively the mean and standard deviation of X_q.

If a uniform probability is used to model the distribution of q grams, we deduce after some calculus from the results obtained by Nicodème [15](section 3.3) that :

$$\left\{ \begin{array}{lcl} \gamma & = & nmp + \mathcal{O}(p^2) \\ \sigma & = & nmp + \mathcal{O}(p^2) \end{array} \right.$$

where $p = \frac{1}{s^q}$. Since, in our application $s = 2^{14}$ and n_q is an integer the parameter γ may be neglected in equation 1. Note that as mentioned in the introduction of this section, the probability $p(X_q = n_q)$ decreases drastically when s becomes large.

Let us now consider q grams of length i and j with $i > j$ between two fingerprints. Each i-gram induces $i - j + 1$ j-grams between the two fingerprints. Therefore, the existence of n_i i-grams between the two fingerprints induces the existence of $n_i(i - j + 1)$ j grams between the same fingerprints. Given, the number n_j of j grams we consider that the event relative to the apparition of the $n_j - n_i(i - j + 1)$ remaining j gram is independent of the event having produced the n_i i-gram. We have thus:

$$\forall i > j \quad p(X_j = n_j | X_i = n_i) = p(X_j = n_j - n_i(i-j+1))$$

Note that this last relation is only an approximation of the real case where the different q grams are not independent. However, this approach is consistent with the urn and balls model used by Nicodème where the different q grams are considered as independent.

If we iterate the Bayes relationship $max - 2$ times on $Q(n_{min}, \ldots, n_{max})$ we obtain:

$$Q(n_{min}, \ldots, n_{max}) = \Pi_{i=min}^{max} p(X_i = u_i)$$

with $u_{max} = n_{max}, u_{max-1} = n_{max-1} - 2u_{max}$, $u_i = n_i - \sum_{j=i+1}^{max} \alpha_{i,j} u_i$ and $\alpha_{i,j} = i - j + 1$.

The expression $-\log_2(Q(n_{min}, \ldots, n_{max}))$ may thus be written as:

$$\begin{array}{ll} & -\sum_{i=3}^{max} log_2(p(X_i = u_i)) \\ = & log_2(e) \sum_{i=3}^{max} \frac{u_i^2}{\sigma_i^2} + \frac{1}{2} \sum_{i=3}^{max} \log_2(2\pi\sigma_i) \end{array}$$

where σ_i denotes the standard deviation of X_i. Note that the mean of X_i is neglected in the above equation.

Since the second term of the above equation is independent of $(n_i)_{i \in \{min, max\}}$ and that we are only searching for an asymptotic approximation of $-\log_2(Q(n_{min}, \ldots, n_{max}))$ we define our score $S(n_{min}, \ldots, n_{max})$ between two fingerprints as :

$$S(n_{min}, \ldots, n_{max}) = \sum_{i=3}^{max} \frac{u_i^2}{\sigma_i^2} = \frac{1}{(mn)^2} \sum_{i=3}^{max} u_i^2 s^{2i}$$

If we forget the factor $\frac{1}{mn}$, since $u_i << s$, our scoring function may be understood as the value of the number $(0, 0, 0, u_3, \ldots, u_{max})$ in base s^2. This last formula provides us a practical way to weight the information concerning the number of q grams of different length.

5 Experiments

Our database contains 357 songs of approximately 4 minutes each (around 5300 values per song). All songs were subjected to the MP3 encoding/decoding at various rates and shifted by adding silence of various length at the beginning of each song. The intervals I_o, I_f and I_e have been respectively set to 100ms, 80ms and 1ms for these experiments.

5.1 Size of the fingerprints

The size of the intervals I_e and I_o being respectively equal to 1 and 100 milliseconds, the minimum and maximum detection rates of I_f intervals during one second are respectively equals to 10 and 1000 (Section **??**). These lower and upper bounds of the detection rate represent extremal values which have not been reached within our test database. Indeed, the minimal and maximal detection rates measured on our database are respectively equal to 18 and 34. The mean detection rate on the whole database is equal to 21.9 with a standard deviation of 3.5.

Since each sub-fingerprint is stored on 14 bits, and that 21.9 sub-fingerprints are computed per seconds, the size required by our method to store one minute of signal is equal to $21.9 * (14/8) * 60 = 2.3$ Kilo bytes per minute. On the other hand, the method of Kalker and Haitsma, uses intervals of $370ms$ with an overlap of $31/32$. Each new frame, adds thus $11.56ms$ to the signal covered by the fingerprint. Since each sub-fingerprint is stored on 32 bits, the size required by a fingerprint corresponding to one minute of signal is equal to $(32/8) * 60/(11.56\,10^{-3}) = 20.5$ Kilo bytes. The size of the fingerprint used by our method is thus approximately 9 times lower than the one of Kalker and Haitsma. These results have been confirmed by measuring on our database the mean size required to store both types of fingerprints.

5.2 Measurement of the performances

Let us denote by T_i the set of intervals used to compute the sub-fingerprints of a signal s_i. Using the method of Kalker and Haitsma [9], this set is equal to the number of sliding windows used by this algorithm. Using our method this set is equal to the number of I_f intervals defined by the segmentation step.

Some additional quantities may be usefully defined to measure the performances of our method: Given an audio file s_i, let us denote by $SP_i \subset T_i$ the set of intervals located at a same position within s_i and a degraded version of s_i. We consider, that two intervals have a same position if the distance between the centers of the two intervals is less than $0.25ms$. When the degraded version of s_i is shifted, the position of the interval within the shifted version is translated by the shift before computing the distance. Moveover, let us additionally consider the set $SV_i \subset SP_i$ of intervals having a same location and a same sub-fingerprint value within s_i and its degraded version.

Given a specific degradation, several quantities may be defined in order to measure the performances of our algorithm:

Segmentation rate: This quantity represents the mean value of I_f intervals located at a same position within the original signal and its degraded version. This quantity is thus a measure of the performances of our segmentation algorithm. It is formally defined by :

$$SR = \frac{1}{N} \sum_{i=1}^{N} \frac{|SP_i|}{|T_i|} \qquad (2)$$

where $|.|$ denote the cardinal of the set and N the number of audio files of the database.

Recognition rate: The robustness of our method used to compute sub-fingerprints is measured for each s_i by the ratio between the cardinals of SV_i and SP_i. We measure thus the ratio of intervals whose sub-fingerprint value remains unchanged by a degradation. The mean value of this ratio over the whole database defines the recognition rate and is formally defined by:

$$RR = \frac{1}{N} \sum_{i=1}^{N} \frac{|SV_i|}{|SP_i|} \qquad (3)$$

Total recognition rate : The recognition rate defined above, measures the robustness of our sub-fingerprints independently of the segmentation step. A global measure of both steps may be achieved by computing for each signal s_i the ratio between SV_i and T_i. This measure may be understood, as the product, for each s_i, of $\frac{|SP_i|}{|T_i|}$ and $\frac{|SV_i|}{|SP_i|}$ respectively used to define the segmentation and the recognition rates. The mean value of this ratio over the database defines the total recognition rate formally defined as :

$$TRR = \frac{1}{N} \sum_{i=1}^{N} \frac{|SV_i|}{|T_i|} \qquad (4)$$

We also measured the performance of Kalker and Haitsma [9] algorithm. The segmentation rate and the recognition rate are meaningless for this method since it does not perform a segmentation step. We thus only use the total recognition rate to measure the performances of this algorithm. Further experiments based on the bit error rate may be found in [13].

Note that the quantities defined in this section measure the robustness of our fingerprints against degradation. These quantities don't readily allow to measure the efficiency of an indexing scheme which does not constitute the core of this paper (section 6). Indeed, the sets SP_i and SV_i are usually not known when a request on the fingerprint database is performed using the fingerprint of an unknown signal.

5.3 Influence of compression

All the audio files of our database are endoded using 705 $Kbps$. The compression rate may thus be measured using either the bit rates of the compressed files or the ratio between $705Kbps$ and these bit rates. The bit rates 48, 64, 96, 128, 192 and 256 $Kbps$ used in our experiments are thus respectively equivalent to the ratios 14.7, 11.02, 7.35, 5.5, 3.67 and 2.75.

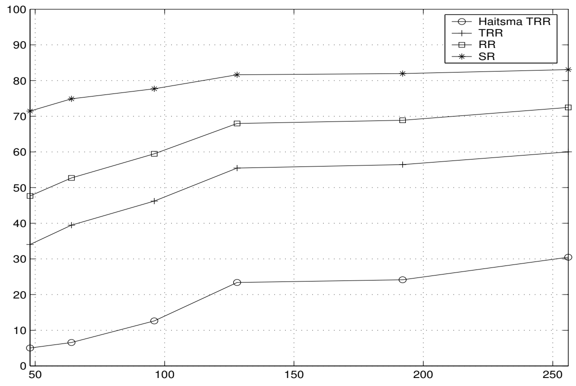

Figure 2. Common values and segmentation rates when comparing an audio file with its compressed version (at 48, 64, 96, 128, 192 and 256 Kbps

The top level curve of Figure 2 (-*-) represents the segmentation rates (equation 2) obtained by our method against various compression rates. As schown by Figure 2 the segmentation rate is equal to 93% for an encoding of the signal at $256Kbps$. This ratio drecrease at 87% for an encoding at $128Kbps$ and up to 80% for a signal encoded at $48Kbps$. High values of the segmentation rate are thus obtained for all the usual compression rates used within this experiment.

The second (-+-) curve of Figure 2 represent recognition rate (equation 4) obtained by our first idea of fingerprint construction scheme combined with the commom framing method (Section 2). The curve decrease slightly from an encoding at $256Kbps$ until an encoding at $128Kbps$ and then more abruptly for an encoding at $48Kbps$. The values of the recognition rate at the key bit rates $256Kbps$, $128Kbps$ and 48 $Kbps$ are respectively equal to 73%, 69% and 48%. Therefore a large part of the fingerprint (at least 48%) is computed without any error for all the compression rates tested in this experiments. Considering the total recognition rate, at least 34% of all the intervals are correctly detected and associated to the same sub-fingerprint value for all compression rates.

The last curve of Figure 2 (-o-) shows the total recognition rate obtained by the method of Kalker and Haitsma (equation 4). As shown by Figure 2, the best ratio obtained by the method of Kalker and Haitsma is equal to 30% for an encoding at $256Kbps$. This ratio decreases to 5% for an encoding at $48Kbps$. The best performance obtained by the method of Kalker and Haitsma is thus lower than the worse one obtained by our method(-+-).

The greater robustness of our method against compression compared to the one of Kalker and Haitsma may be explained by the two following reasons: Firstly, the sub-fingerprints are computed on important peaks of the signal which are less sensible to the degradation induced by compression algorithm than other parts of the signal with few information. Secondly, the proposed construction scheme of our sub-fingerprint (Section **??**) is more resistant to small differences of the signal within each interval.

5.4 Influence of time shifting

Figures 3 represents the influence of time shifting on our fingerprint algorithm. The shift operations have been performed by adding a silence of 1, 2, 3, 5 and 6.25 ms at the begining of the signal.

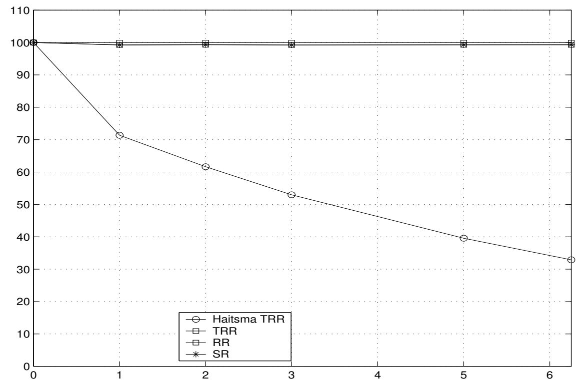

Figure 3. Common values and segmentation rates when comparing an audio file with its time shifted (1, 2, 3, 5 and 6.25 ms) version

The top level curve (-*-) of Figure 3 represents the segmentation rate (equation 2) obtained by our method against the sequence of time shifting operations. This curve represents thus as in Figure 2 the performance of our segmentation algorithm. One can observe in Figure 3 a slight decrease from 100% to 99% of the segmentation rate when a shift is introduced. This decrease may be interpreted as the time required by our method to be synchronized on a significant peak of the signal (Section **??**). Note that after, this slight decrease the number of detected I_f intervals remains contant for any shift value and even for higher time-shift (10, 25 and 50 ms). This last result may be interpreted as the fact that the position on the peack on which the two fingerprints become synchronised is weakly influenced by the shift operation.

Given a signal s_i and an interval I_f of SP_i within s_i, the content of the signal within I_f will be the same within s_i and its shifted version up to the slight shift allowed between two intervals considered as equal. The shift operation alters thus only the segmentation step and we have in this case $SP_i = SV_i$. The recognition rate should thus be equal to 100% while the total recognition rate should be equal to the segmentation rate. This result is confirmed by the second curve of Figure 3(⊟) which represents the total recognition rates obtained by our method (equation 4). This curves presents as the one representing the segmentation rate a slight decreases from 100% to 99% when a shift is introduced. The total recognition rate remains then constant for all shifts further introduced.

The last curve of Figure 3 (⊖) represents the total recognition rates of the method of Kalker and Haitsma. This curve present a strong gap when the first shift is introduced (from 100% to 71%) and then decreases approximately linearly until a rate of 33% for a shift of $6.5ms$.

6 Toward Database search

If the fingerprint algorithm is based a decomposition of the signal into overlapping intervals, the distance between two fingerprints may be defined as a sum of distances between two sequences of sub-fingerprints. However, using a segmentation step, the alteration of the signal may remove or add sub-fingerprints in any of the two sequences of sub-fingerprints whose distance should be computed. The sum of the distance of the consecutive sub-fingerprints become then meaningless since the signal is not compared at the same locations in both fingerprints. As schown in Section 5, a large number of sub-fingerprints are preservered by our method using either compression or shift alterations. This result suggests that a distance between fingerprints based on approximate string matching [8] should provide low distances for fingerprints associated to a same audio file. Indeed, the approximate string matching distance may be roughly understood as the number of insertion, deletion and substitutions of sub-fingerprints that should be performed in order to transform one fingerprint into another. As shown in Section 5 the low number of non corresponding sub-fingerprints obtained by our method insures that a low distance should be obtained between fingerprints associated to a same signal.

7 Conclusions

We have presented in this paper a new audio fingerprint extraction method based on an audio segmentation algorithm and a new construction scheme of the sub-fingerprints. The segmentation algorithm determines important peaks within the signal while insuring a relatively constant detection rate. The sub-fingerprints are generated by looking at the differences along the frequency between the energy of one bin and the mean energy of the previous bins.

The robustness of our method against compression comes from the segmentation step which compute sub-fingerprints on important peaks of the signal and from our new construction scheme of sub-fingerprints. The segmentation algorithm additionally provides an important robustness against time shifting due to the synchronisation of the fingerprints on important peaks of the signal. Finally, the selection rate of our segmentation algorithm provides a fingerprint database whose size is approximately 4 time smaller than the one defined by Kalker and Haitsma.

In future work, we plan to further investigate the structure of our fingerprint's database and to design on it an appropriate indexation scheme based for example on the string edit distance already investigated by several authors.

References

[1]

[2] J. P. C. Burges and S. Jana. Distorsion discriminant analysis for audio fingerprnting. *IEEE Transactions on Speech and Audio Processing*, 11(3):165–174, 2003.

[3] W. Chang and T. Marr. Approximate string matching and local similarity. In M. Crochemore and D. Gusfield, editors, *Combinatorial Pattern Matching, Proceedings of the 5th Annual Symposium*, volume 807 of *Lecture Notes in Computer Science*, pages 259–273. Springer Verlag, 1994.

[4] M. Crochemore and W. Rytter. *Text algorithms*. Oxford University Press, 1995.

[5] P. Doets, M. Gisbert, and R. Lagendijk. On the comparison of audio fingerprints for extracting quality parameters of compressed audio. In *In Proceedings of SPIE*, volume 6072, February 2006.

[6] S. Downie. *Evaluating a Simple Approach to Music Information Retreival: Conceiving Melodic n-grams as Text*. PhD thesis, University of Western Ontario, 1999.

[7] S. Hainsworth and M. Macleod. Onset detection in musical audio signals. In *Proceedings of the International Computer Music Conference*, Singapore, September 2003.

[8] T. C. Hoad. *Video Representations for Effective Rettrieval From Large Collections*. PhD thesis, RMIT University, Melbourne, Australia, 2004.

[9] T. Kalker and J. Haitsma. A highly robust audio fingerprinting system. In *Proceedings of ISMIR'2002*, pages 144–148, 2002.

[10] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999.

[11] F. Kurth. A ranking technique for fast audio identification. In *Proceedings of the International Workshop on Multimedia Signal Processing*, 2002.

[12] J. Lebossé, L. Brun, and J. C. Pailles. Fingerprint audio robuste pour la gestion de droits. In *Actes de CORESA 2006*, Caen, November 2006.

[13] J. Lebossé, L. Brun, and J. C. Pailles. A robust audio fingerprint for digital rights management. Technical report, Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen (UMR 6072), http://www.greyc.ensicaen.fr/∼luc/ARTICLES/tr-lebosse2006.ps, September 2006.

[14] S. Mallat. *A Wavelet tour of signal processing*. Academic Press, 1999. chapter VIII p. 363.

[15] P. nicodème. Q-gram analysis and urn models. In *Proc. of Discrete Random Walks DRW2003*, pages 243–258, 2003.

[16] D. Shyamala and S. Rüger. Robust polyphonic music retreival with n-grams. *Journal of Intelligent Information systems*, 21(1):53–70, 2003.