A ROBUST AUDIO FINGERPRINT EXTRACTION ALGORITHM

Jérôme Lebossé
France Télécom R&D
32 rue des coutures
14000 Caen,France
jerome.lebosse@orange-ft.com

Luc Brun
GREYC UMR 6072
ENSICAEN,6 Boulevard du Maréchal Juin
14050 Caen, France
luc.brun@greyc.ensicaen.fr

Jean Claude Pailles
France Télécom R&D,
32 rue des coutures
14000 Caen,France
jeanclaude.pailles@orange-ft.com

ABSTRACT

An Audio fingerprint is a small digest of an audio file computed from its main perceptual properties. Like human fingerprints, Audio fingerprints allows to identify an audio file among a set of candidates but does not allow to retreive 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 which combines a segmentation method with a new fingerprint construction scheme. The proposed method is robust against compression and time shifting alterations of the audio files.

KEY WORDS
audio fingerprint, segmentation, indexation.

1 Introduction

Audio fingerprint [5, 7, 1] 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, one 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,…). 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. Moreover, the fingerprint should be composed of elementary keys (called subfingerprints) based on small parts of the signal. The subfingerprints are then computed either continuously along the signal or at a sufficient rate in order to be able to characterize a file from a short sample.

A fingerprint system consists of two components: a method which extract the fingerprint and a search method which match fingerprints. In this study, we focus our attention on the first part of the system : the fingerprint extraction. 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.

2 State of the art

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 subfingerprint) which characterises it.

A usual method [7, 5] consists to decompose the signal into a sequence of overlapping intervals called frames, of a few milliseconds. Enframing the audio stream allows to treat each frame as a relatively stationary sound. The frames are then weighted by a Hamming window to minimize the discontinuities at the beginning and at the end of each frame. An overlap factor between frames, 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 subfingerprint 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 [8]. Moreover, the use of overlapping windows only reduces the influence of such cuts (Section 4.4).

Another way to segment an audio waveform is to find particular positions in the audio signal, called onsets [6, 3]. 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 subfingerprint. This unpredictability is a major drawback for fingerprinting systems which should identify a file in a 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 [7] proposes a subfingerprint 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 [2]. 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. [1] use a particular decomposition of the spectrum called the Modulated Complex Lapped Transform (MCLT).

The method of Kalker and Haitsma [5] 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 subfingerprint 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:

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

3 Robust Identification

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 4). 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 sufficient time intervals will be selected in a given time inerval.

3.1 Audio segmentation

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. The interval I_e with a maximal energy (I_{emax}) within I_o is then selected.

- In the third step, a last interval, called the Features Interval (I_f) is defined centered on I_{emax}.

Finally, a features extraction algorithm is applied on I_f to compute a sub-fingerprint (Section 3.2).

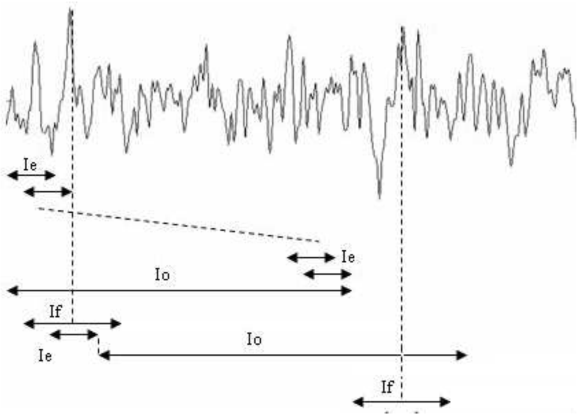

Figure 1. Audio Segmentation : The interval I_{emax} corresponds to the interval of greatest energy within I_o. The interval I_f is centered around I_{emax}.

Given a selected I_f interval, the begining of the next I_o interval is set to the end of I_e (Fig. 1). The distance between the centers of two I_f intervals lies thus between I_e and I_o-I_e. 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. Since I_f is centered around I_{emax} the next I_o interval will 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 detect, and select as I_f several I_e 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.

3.2 SubFingerprint design

Our method to compute a sub fingerprint on each I_f interval is based on the same scheme than the one of Haistma and Kalker [5](Section 2). We thus use as these authors a decomposition of the spectrum of I_f into a sequence of bands with a logarithmic spacing. However, as shown by our experiments (Section 4), strong compression rates may alter significantly the robustess of the sub fingerprint extraction algorithm. Indeed, the corruptions of the signal by noise, compression, or cutting operations reduces drastically the number of I_f intervals with a same sub fingerprint. Haistma and Kalker attenuate this problem by using Hamming distance between the sequences of sub fingerprint of two audio files [5]. However, the corruption of subfingerprints by noise and alterations corrupts the Haming distances and reduces the amount of information that an indexation algorithm may deduce from such distances. The robustness of the sub fingerprint extraction algorithm may be improved using the two following remarks:

1. The uses of two successive frames to design a sub fingerprint involves the corruption of two subfingerprints if an error occurs in the measure of their common frame.

2. The comparison of the energies of two successive bands of a spectrum within a same frame is sensitive to the errors which may corrupt a single band.

We solve the first source of errors by using only one frame for each subfingerprint computation. The second source of errors is connected to the corruption of one band of the I_f spectrum. Using the same notations than in Section 2, the alteration of the measure of the energy of a single band ($EB(n,m)$) alters the values of $\Delta EB(n,m-1)$ and $\Delta EB(n,m)$. This alteration of the bands energies may be considered as the presence of a random noise on the signal ($EB(n,m)$) $m \in \{1,..,M\}$ where M represents the index of the last energy band. If we suppose that the noise is non correlated between the different samples of the signal ($EB(n,m)$) $m \in 1,..,M$, the influence of noise may be reduced by using a function of m defined as a sum of some $EB(n,m)$. We thus define the mean energy $S(n,m)$ of a band m, within a frame n as the mean of all the band's energies from 0 to m:

$$S(n,m) = \frac{1}{m}\sum_{j=0}^{m} EB(n,j)$$

We then replace $EB(n,m)$ by $S(n,m)$ in the computation of the differences of energies band and define the m^{th} bit of the sub fingerprint associated to the frame n ($F(n,m)$) as follows:

$$F(n,m) = \begin{cases} 1 & if\ S(n,m) - S(n,m-1) \geq 0 \\ 0 & otherwise \end{cases}$$

One can easily show that $S(n,m) - S(n,m-1) = \frac{1}{m}(EB(n,m) - S(n,m-1))$. The above formula may thus be simplified as follows:

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

The sub fingerprint of each frame n, is defined by the concatenation of the M bits $F(n,m)$ with $m \in \{1,...M\}$. The parameter M is fixed to 32 in our experiments (Section 4). The audio fingerprint is then defined as the concatenation of its sequence of sub-fingerprints .

3.3 Resistance to attacks

If our method is used within the DRM framework, one possible attack could consist to consider an audio file whose fingerprint belongs to the database and to modify it outside the I_f intervals used to compute the fingerprint. The DRM application would thus be unable to distiguish this new signal from the older one. Such an attack is possible only if the I_f intervals constitute a small parts of the whole signal. However, in the experiments presented below (Section 4) the size of the I_f intervals has been fixed to $80ms$ and the mean number of I_f interval per second computed on our database is equal to $21,9$. The time interval used to compute subfingerprints within 1 second is thus equal to $80ms * 21,9 = 1,76s$. The I_f intervals are thus defined on a large part of the signal with many overlaps. A modification of the signal, outside the I_f intervals would thus perform very few modifications and lead to an altered version close to the original.

One alternative attack consists to insert random peaks within the signal in order to destroy its fingerprint. However, using fingerprint methods within a DRM framework, an audio file not identified within the database has no associated rights (including the right of reading). Such a manipulation would thus be meaningless.

4 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.

4.1 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 3.1). 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 subfingerprint is stored on 4 bytes, and that 21.9 subfingerprints are computed per seconds, the size required by our method to store one minute of signal is equal to $21.9 * 4 * 60 = 5.2$ 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 subfingerprint is stored on 4 bytes, the size required by a fingerprint corresponding to one minute of signal is equal to $4 * 60/(11.56\,10^{-3}) = 20.5$ Kilo bytes. The size of the fingerprint used by our method is thus approximately 4 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.

4.2 Measurement of the performances

Let us denote by T_i the set of intervals used to compute the subfingerprints of a signal s_i. Using the method of Kalker and Haitsma [5], 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 center 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 subfingerprint 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 :

$$SR = \frac{1}{N}\sum_{i=1}^{N}\frac{|SP_i|}{|T_i|} \tag{1}$$

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 subfingerprints 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 subfingerprint 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:

$$RR = \frac{1}{N}\sum_{i=1}^{N}\frac{|SV_i|}{|SP_i|} \tag{2}$$

Total recognition rate : The recognition rate defined above, measures the robustness of our subfingerprints 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 :

$$TRR = \frac{1}{N}\sum_{i=1}^{N}\frac{|SV_i|}{|T_i|} \tag{3}$$

We also measured the performance of Kalker and Haitsma [5] 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 using the bit rate error between files may be found in [**?**]

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 5). 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.

4.3 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.

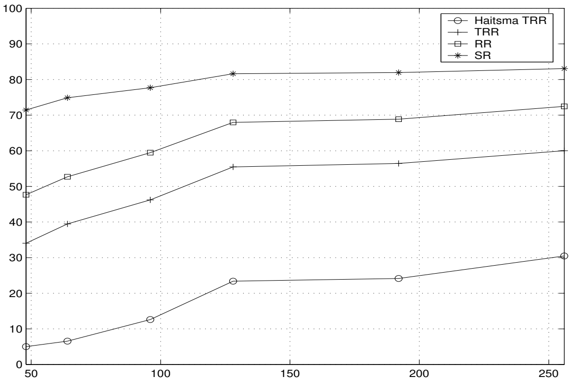

Figure 2. Common values and segmentation rates when comparing an audio file with its compressed version (at 48, 64, 96, 128, 192 and 256 Kbps

The top level curve of Figure 2 (-*-) represents the segmentation rates (equation 1) obtained by our method against various compression rates. As schown by Figure 2 the segmentation rate is equal to 83% for an encoding of the signal at $256Kbps$. This ratio remains approximately stable until an encoding at $128Kbps$ and decreases up to 71% 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 (-□-) and third (-+-) curves of Figure 2 represent respectively the recognition rates (equation 2) and the total recognition rates (equation 3) obtained by our method. Both curves decrease slightly from and encoding at $256Kbps$ until an encoding at $128Kbps$ and then more abruptly for an encoding at $48Kbps$. The values of the recognition rate (resp. total recognition rate) at the key bit rates $256Kbps$, $128Kbps$ and 48 $Kbps$ are respectively equal to 73% (resp. 60%), 69% (resp. 56%) and 48% (resp. 34%). 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 subfingerprint 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 3). 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 subfingerprints 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 subfingerprint (Section 3.2) is more resistant to small differences of the signal within each interval.

4.4 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.

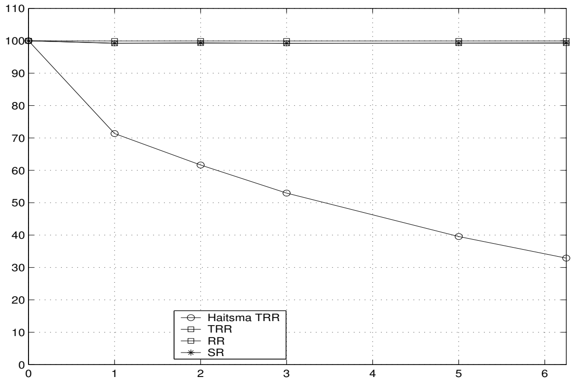

Figure 3. Common values and segmentation rates when comparing an audio file with its time shifted (1, 2, 3, 5 and 6.25 ms) version

The top level curve (-*-) of Figure 3 represents the segmentation rate (equation 1) 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 3.1). 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 3). 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 $6.5ms$.

5 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 subfingerprints. However, using a segmentation step, the alteration of the signal may remove or add subfingerprints in any of the two sequences of subfingerprints whose distance should be computed. The sum of the distance of the consecutive subfingerprints become then meaningless since the signal is not compared at the same locations in both fingerprints. As schown in Section 4, a large number of subfingerprints are preservered by our method using either compression or shift alterations. This result suggests that a distance between fingerprints based on approximate string matching [4] 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 subfingerprints that should be performed in order to transform one fingerprint into another. Has shown in Section 4 the low number of non corresponding subfingerprints obtained by our method insure that a low distance should be obtained between fingerprints associated to a same signal.

6 Conclusions

We have presented in this paper a new audio fingerprint method based on an audio segmentation algorithm and a new construction scheme of the subfingerprints. The segmentation algorithm determines important peaks within the signal while insuring a relatively constant detection rate. The subfingerprints 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 subfingerprints on important peaks of the signal and from our new construction scheme of subfingerprints. 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] 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.

[2] 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.

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

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

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

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

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

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