A Highly Robust Audio Fingerprinting System

Jaap Haitsma
Philips Research
Prof. Holstlaan 4, 5616 BA,
Eindhoven, The Netherlands
++31-40-2742648
Jaap.Haitsma@philips.com

Ton Kalker
Philips Research
Prof. Holstlaan 4, 5616 BA,
Eindhoven, The Netherlands
++31-40-2743839
Ton.Kalker@ieee.org

ABSTRACT

Imagine the following situation. You're in your car, listening to the radio and suddenly you hear a song that catches your attention. It's the best new song you have heard for a long time, but you missed the announcement and don't recognize the artist. Still, you would like to know more about this music. What should you do? You could call the radio station, but that's too cumbersome. Wouldn't it be nice if you could push a few buttons on your mobile phone and a few seconds later the phone would respond with the name of the artist and the title of the music you're listening to? Perhaps even sending an email to your default email address with some supplemental information. In this paper we present an audio fingerprinting system, which makes the above scenario possible. By using the fingerprint of an unknown audio clip as a query on a fingerprint database, which contains the fingerprints of a large library of songs, the audio clip can be identified. At the core of the presented system are a highly robust fingerprint extraction method and a very efficient fingerprint search strategy, which enables searching a large fingerprint database with only limited computing resources.

1. INTRODUCTION

Fingerprint systems are over one hundred years old. In 1893 Sir Francis Galton was the first to "prove" that no two fingerprints of human beings were alike. Approximately 10 years later Scotland Yard accepted a system designed by Sir Edward Henry for identifying fingerprints of people. This system relies on the pattern of dermal ridges on the fingertips and still forms the basis of all "human" fingerprinting techniques of today. This type of forensic "human" fingerprinting system has however existed for longer than a century, as 2000 years ago Chinese emperors were already using thumbprints to sign important documents. The implication is that already those emperors (or at least their administrative servants) realized that every fingerprint was unique. Conceptually a fingerprint can be seen as a "human" summary or signature that is unique for every human being. It is important to note that a human fingerprint differs from a textual summary in that it does not allow the reconstruction of other aspects of the original. For example, a human fingerprint does not convey any information about the color of the person's hair or eyes.

Recent years have seen a growing scientific and industrial interest in computing fingerprints of multimedia objects [1][2][3][4] [5][6]. The growing industrial interest is shown among others by a large number of (startup) companies [7][8][9][10][11][12][13] and the recent request for information on audio fingerprinting technologies by the International Federation of the Phonographic Industry (IFPI) and the Recording Industry Association of America (RIAA) [14].

The prime objective of multimedia fingerprinting is an efficient mechanism to establish the perceptual equality of two multimedia objects: not by comparing the (typically large) objects themselves, but by comparing the associated fingerprints (small by design). In most systems using fingerprinting technology, the fingerprints of a large number of multimedia objects, along with their associated meta-data (e.g. name of artist, title and album) are stored in a database. The fingerprints serve as an index to the meta-data. The meta-data of unidentified multimedia content are then retrieved by computing a fingerprint and using this as a query in the fingerprint/meta-data database. The advantage of using fingerprints instead of the multimedia content itself is three-fold:

1. Reduced memory/storage requirements as fingerprints are relatively small;
2. Efficient comparison as perceptual irrelevancies have already been removed from fingerprints;
3. Efficient searching as the dataset to be searched is smaller.

As can be concluded from above, a fingerprint system generally consists of two components: a method to extract fingerprints and a method to efficiently search for matching fingerprints in a fingerprint database.

This paper describes an audio fingerprinting system that is suitable for a large number of applications. After defining the concept of an audio fingerprint in Section 2 and elaborating on possible applications in Section 3, we focus on the technical aspects of the proposed audio fingerprinting system. Fingerprint extraction is described in Section 4 and fingerprint searching in Section 5.

2. AUDIO FINGERPRINTING CONCEPTS

2.1 Audio Fingerprint Definition

Recall that an audio fingerprint can be seen as a short summary of an audio object. Therefore a fingerprint function F should map an audio object X, consisting of a large number of bits, to a fingerprint of only a limited number of bits.

Here we can draw an analogy with so-called hash functions[1], which are well known in cryptography. A cryptographic hash function H maps an (usually large) object X to a (usually small) hash value (a.k.a. message digest). A cryptographic hash function allows comparison of two large objects X and Y, by just comparing their respective hash values $H(X)$ and $H(Y)$. Strict mathematical equality of the latter pair implies equality of the former, with only a very low probability of error. For a properly designed cryptographic hash function this probability is 2^{-n}, where n equals the number of bits of the hash value. Using cryptographic hash functions, an efficient method exists to check whether or not a particular data item X is contained in a given and large data set $Y=\{Y_i\}$. Instead of storing and comparing with all of the data in Y,

[1] In the literature fingerprinting is sometimes also referred to as robust or perceptual hashing[5].

it is sufficient to store the set of hash values $\{h_i = H(Y_i)\}$, and to compare $H(X)$ with this set of hash values.

At first one might think that cryptographic hash functions are a good candidate for fingerprint functions. However recall from the introduction that, instead of strict mathematical equality, we are interested in perceptual similarity. For example, an original CD quality version of 'Rolling Stones – Angie' and an MP3 version at 128Kb/s sound the same to the human auditory system, but their waveforms can be quite different. Although the two versions are perceptually similar they are mathematically quite different. Therefore cryptographic hash functions cannot decide upon perceptual equality of these two versions. Even worse, cryptographic hash functions are typically bit-sensitive: a single bit of difference in the original object results in a completely different hash value.

Another valid question the reader might ask is: "Is it not possible to design a fingerprint function that produces mathematically equal fingerprints for perceptually similar objects?" The question is valid, but the answer is that such a modeling of perceptual similarity is fundamentally not possible. To be more precise: it is a known fact that perceptual similarity is not transitive. Perceptual similarity of a pair of objects X and Y and of another pair of objects Y and Z does not necessarily imply the perceptual similarity of objects X and Z. However modeling perceptual similarity by mathematical equality of fingerprints would lead to such a relationship.

Given the above arguments, we propose to construct a fingerprint function in such a way that perceptual similar audio objects result in similar fingerprints. Furthermore, in order to be able discriminate between different audio objects, there must be a very high probability that dissimilar audio objects result in dissimilar fingerprints. More mathematically, for a properly designed fingerprint function F, there should be a threshold T such that with very high probability $\|F(X)-F(Y)\|\leq T$ if objects X and Y are similar and $\|F(X)-F(Y)\|>T$ when they are dissimilar.

2.2 Audio Fingerprint System Parameters

Having a proper definition of an audio fingerprint we now focus on the different parameters of an audio fingerprint system. The main parameters are:

- **Robustness**: can an audio clip still be identified after severe signal degradation? In order to achieve high robustness the fingerprint should be based on perceptual features that are invariant (at least to a certain degree) with respect to signal degradations. Preferably, severely degraded audio still leads to very similar fingerprints. The false negative rate is generally used to express the robustness. A false negative occurs when the fingerprints of perceptually similar audio clips are too different to lead to a positive match.
- **Reliability**: how often is a song incorrectly identified? E.g. "Rolling Stones – Angie" being identified as "Beatles – Yesterday". The rate at which this occurs is usually referred to as the false positive rate.
- **Fingerprint size**: how much storage is needed for a fingerprint? To enable fast searching, fingerprints are usually stored in RAM memory. Therefore the fingerprint size, usually expressed in bits per second or bits per song, determines to a large degree the memory resources that are needed for a fingerprint database server.
- **Granularity**: how many seconds of audio is needed to identify an audio clip? Granularity is a parameter that can depend on the application. In some applications the whole song can be used for identification, in others one prefers to identify a song with only a short excerpt of audio.
- **Search speed and scalability:** how long does it take to find a fingerprint in a fingerprint database? What if the database contains thousands and thousands of songs? For the commercial deployment of audio fingerprint systems, search speed and scalability are a key parameter. Search speed should be in the order of milliseconds for a database containing over 100,000 songs using only limited computing resources (e.g. a few high-end PC's).

These five basic parameters have a large impact on each other. For instance, if one wants a lower granularity, one needs to extract a larger fingerprint to obtain the same reliability. This is due to the fact that the false positive rate is inversely related to the fingerprint size. Another example: search speed generally increases when one designs a more robust fingerprint. This is due to the fact that a fingerprint search is a proximity search. I.e. a similar (or the most similar) fingerprint has to be found. If the features are more robust the proximity is smaller. Therefore the search speed can increase.

3. APPLICATIONS

In this section we elaborate on a number of applications for audio fingerprinting.

3.1 Broadcast Monitoring

Broadcast monitoring is probably the most well known application for audio fingerprinting[2][3][4][5][12][13]. It refers to the automatic playlist generation of radio, television or web broadcasts for, among others, purposes of royalty collection, program verification, advertisement verification and people metering. Currently broadcast monitoring is still a manual process: i.e. organizations interested in playlists, such as performance rights organizations, currently have "real" people listening to broadcasts and filling out scorecards.

A large-scale broadcast monitoring system based on fingerprinting consists of several monitoring sites and a central site where the fingerprint server is located. At the monitoring sites fingerprints are extracted from all the (local) broadcast channels. The central site collects the fingerprints from the monitoring sites. Subsequently, the fingerprint server, containing a huge fingerprint database, produces the playlists of all the broadcast channels.

3.2 Connected Audio

Connected audio is a general term for consumer applications where music is somehow connected to additional and supporting information. The example given in the abstract, using a mobile phone to identify a song is one of these examples. This business is actually pursued by a number of companies [10][13]. The audio signal in this application is severely degraded due to processing applied by radio stations, FM/AM transmission, the acoustical path between the loudspeaker and the microphone of the mobile phone, speech coding and finally the transmission over the mobile network. Therefore, from a technical point of view, this is a very challenging application.

Other examples of connected audio are (car) radios with an identification button or fingerprint applications "listening" to the audio streams leaving or entering a soundcard on a PC. By pushing an "info" button in the fingerprint application, the user could be directed to a page on the Internet containing information about the artist. Or by pushing a "buy" button the user would be able to buy the album on the Internet. In other words, audio fingerprinting can provide a universal linking system for audio content.

3.3 Filtering Technology for File Sharing

Filtering refers to active intervention in content distribution. The prime example for filtering technology for file sharing was Napster [15]. Starting in June 1999, users who downloaded the Napster client could share and download a large collection of music for free. Later, due to a court case by the music industry, Napster users were forbidden to download copyrighted songs. Therefore in March 2001 Napster installed an audio filter based on file names, to block downloads of copyrighted songs. The filter was not very effective, because users started to intentionally misspell filenames. In May 2001 Napster introduced an audio fingerprinting system by Relatable [8], which aimed at filtering out copyrighted material even if it was misspelled. Owing to Napster's closure only two months later, the effectiveness of that specific fingerprint system is, to the best of the author's knowledge, not publicly known.

In a legal file sharing service one could apply a more refined scheme than just filtering out copyrighted material. One could think of a scheme with free music, different kinds of premium music (accessible to those with a proper subscription) and forbidden music.

Although from a consumer standpoint, audio filtering could be viewed as a negative technology, there are also a number of potential benefits to the consumer. Firstly it can organize music song titles in search results in a consistent way by using the reliable meta-data of the fingerprint database. Secondly, fingerprinting can guarantee that what is downloaded is actually what it says it is.

3.4 Automatic Music Library Organization

Nowadays many PC users have a music library containing several hundred, sometimes even thousands, of songs. The music is generally stored in compressed format (usually MP3) on their hard-drives. When these songs are obtained from different sources, such as ripping from a CD or downloading from file sharing networks, these libraries are often not well organized. Meta-data is often inconsistent, incomplete and sometimes even incorrect. Assuming that the fingerprint database contains correct meta-data, audio fingerprinting can make the meta-data of the songs in the library consistent, allowing easy organization based on, for example, album or artist. For example, ID3Man [16], a tool powered by Auditude [7] fingerprinting technology is already available for tagging unlabeled or mislabeled MP3 files. A similar tool from Moodlogic [11] is available as a Winamp plug-in [17].

4. AUDIO FINGERPRINT EXTRACTION

4.1 Guiding Principles

Audio fingerprints intend to capture the relevant perceptual features of audio. At the same time extracting and searching fingerprints should be fast and easy, preferably with a small granularity to allow usage in highly demanding applications (e.g. mobile phone recognition). A few fundamental questions have to be addressed before starting the design and implementation of such an audio fingerprinting scheme. The most prominent question to be addressed is: what kind of features are the most suitable. A scan of the existing literature shows that the set of relevant features can be broadly divided into two classes: the class of semantic features and the class of non-semantic features. Typical elements in the former class are *genre, beats-per-minute,* and *mood.* These types of features usually have a direct interpretation, and are actually used to classify music, generate play-lists and more. The latter class consists of features that have a more mathematical nature and are difficult for humans to 'read' directly from music. A typical element in this class is *AudioFlatness* that is proposed in MPEG-7 as an audio descriptor tool [2]. For the work described in this paper we have for a number of reasons explicitly chosen to work with non-semantic features:

1. Semantic features don't always have a clear and unambiguous meaning. I.e. personal opinions differ over such classifications. Moreover, semantics may actually change over time. For example, music that was classified as *hard rock* 25 years ago may be viewed as *soft listening* today. This makes mathematical analysis difficult.
2. Semantic features are in general more difficult to compute than non-semantic features.
3. Semantic features are not universally applicable. For example, *beats-per-minute* does not typically apply to classical music.

A second question to be addressed is the representation of fingerprints. One obvious candidate is the representation as a vector of real numbers, where each component expresses the weight of a certain basic perceptual feature. A second option is to stay closer in spirit to cryptographic hash functions and represent digital fingerprints as bit-strings. For reasons of reduced search complexity we have decided in this work for the latter option. The first option would imply a similarity measure involving real additions/subtractions and depending on the similarity measure maybe even real multiplications. Fingerprints that are based on bit representations can be compared by simply counting bits. Given the expected application scenarios, we do not expect a high robustness for each and every bit in such a binary fingerprint. Therefore, in contrast to cryptographic hashes that typically have a few hundred bits at the most, we will allow fingerprints that have a few thousand bits. Fingerprints containing a large number bits allow reliable identification even if the percentage of non-matching bits is relatively high.

A final question involves the granularity of fingerprints. In the applications that we envisage there is no guarantee that the audio files that need to be identified are complete. For example, in broadcast monitoring, *any* interval of 5 seconds is a unit of music that has commercial value, and therefore may need to be identified and recognized. Also, in security applications such as file filtering on a peer-to-peer network, one would not wish that deletion of the first few seconds of an audio file would prevent identification. In this work we therefore adopt the policy of *fingerprints streams* by assigning *sub-fingerprints* to sufficiently small atomic intervals (referred to as *frames*). These sub-fingerprints might not be large enough to identify the frames themselves, but a longer interval, containing sufficiently many frames, will allow robust and reliable identification.

4.2 Extraction Algorithm

Most fingerprint extraction algorithms are based on the following approach. First the audio signal is segmented into frames. For every frame a set of features is computed. Preferably the features are chosen such that they are invariant (at least to a certain degree) to signal degradations. Features that have been proposed are well known audio features such as Fourier coefficients [4], Mel Frequency Cepstral Coefficients (MFFC) [18], spectral flatness [2], sharpness [2], Linear Predictive Coding (LPC) coefficients [2] and others. Also derived quantities such as derivatives, means and variances of audio features are used. Generally the extracted

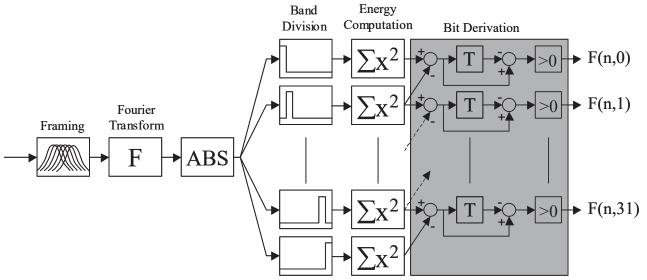

Figure 1. Overview fingerprint extraction scheme.

features are mapped into a more compact representation by using classification algorithms, such as Hidden Markov Models [3], or quantization [5]. The compact representation of a single frame will be referred to as a *sub-fingerprint*. The global fingerprint procedure converts a stream of audio into a stream of sub-fingerprints. One sub-fingerprint usually does not contain sufficient data to identify an audio clip. The basic unit that contains sufficient data to identify an audio clip (and therefore determining the granularity) will be referred to as a *fingerprint-block.*

The proposed fingerprint extraction scheme is based on this general streaming approach. It extracts 32-bit sub-fingerprints for every interval of 11.6 milliseconds. A fingerprint block consists of 256 subsequent sub-fingerprints, corresponding to a granularity of only 3 seconds. An overview of the scheme is shown in Figure 1. The audio signal is first segmented into *overlapping* frames. The overlapping frames have a length of 0.37 seconds and are weighted by a Hanning window with an overlap factor of 31/32. This strategy results in the extraction of one sub-fingerprint for every 11.6 milliseconds. In the worst-case scenario the frame boundaries used during identification are 5.8 milliseconds off with respect to the boundaries used in the database of pre-computed fingerprints. The large overlap assures that even in this worst-case scenario the sub-fingerprints of the audio clip to be identified are still very similar to the sub-fingerprints of the same clip in the database. Due to the large overlap subsequent sub-fingerprints have a large similarity and are slowly varying in time. Figure 2a shows an example of an extracted fingerprint block and the slowly varying character along the time axis.

The most important perceptual audio features live in the frequency domain. Therefore a spectral representation is computed by performing a Fourier transform on every frame. Due to the sensitivity of the phase of the Fourier transform to different frame boundaries and the fact that the Human Auditory System (HAS) is relatively insensitive to phase, only the absolute value of the spectrum, i.e. the power spectral density, is retained.

In order to extract a 32-bit sub-fingerprint value for every frame, 33 non-overlapping frequency bands are selected. These bands lie in the range from 300Hz to 2000Hz (the most relevant spectral range for the HAS) and have a logarithmic spacing. The logarithmic spacing is chosen, because it is known that the HAS operates on approximately logarithmic bands (the so-called *Bark* scale). Experimentally it was verified that the sign of energy differences (simultaneously along the time and frequency axes) is a property that is very robust to many kinds of processing. If we denote the energy of band *m* of frame *n* by *E(n,m)* and the *m*-th bit of the sub-fingerprint of frame *n* by *F(n,m)*, the bits of the sub-fingerprint are formally defined as (see also the gray block in Figure 1, where T is a delay element):

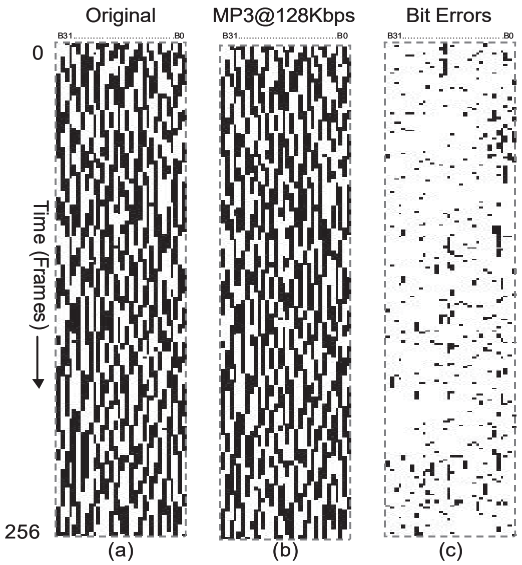

Figure 2. (a) Fingerprint block of original music clip, (b) fingerprint block of a compressed version, (c) the difference between a and b showing the bit errors in black (BER=0.078).

$$F(n,m)=\begin{cases}1 & \text{if } E(n,m)-E(n,m+1)-\big(E(n-1,m)-E(n-1,m+1)\big)>0\\ 0 & \text{if } E(n,m)-E(n,m+1)-\big(E(n-1,m)-E(n-1,m+1)\big)\le 0\end{cases} \quad (1)$$

Figure 2 shows an example of 256 subsequent 32-bit sub-fingerprints (i.e. a fingerprint block), extracted with the above scheme from a short excerpt of *"O Fortuna"* by Carl Orff. A '1' bit corresponds to a white pixel and a '0' bit to a black pixel. Figure 2a and Figure 2b show a fingerprint block from an original CD and the MP3 compressed (32Kbps) version of the same excerpt, respectively. Ideally these two figures should be identical, but due to the compression some of the bits are retrieved incorrectly. These bit errors, which are used as *the similarity measure* for our fingerprint scheme, are shown in black in Figure 2c.

The computing resources needed for the proposed algorithm are limited. Since the algorithm only takes into account frequencies below 2kHz the received audio is first down sampled to a mono audio stream with a sampling rate of 5kHz. The sub-fingerprints are designed such that they are robust against signal degradations. Therefore very simple down sample filters can be used without introducing any performance degradation. Currently 16 tap FIR filters are used. The most computationally demanding operation is the Fourier transform of every audio frame. In the down sampled audio signal a frame has a length of 2048 samples. If the Fourier transform is implemented as a fixed point real-valued FFT the fingerprinting algorithm has been shown to run efficiently on portable devices such as a PDA or a mobile phone.

4.3 False Positive Analysis

Two 3-second audio signals are declared similar if the Hamming distance (i.e. the number of bit errors) between the two derived fingerprint blocks is below a certain threshold *T*. This threshold value *T* directly determines the false positive rate P_f, i.e. the rate at which audio signals are incorrectly declared equal: the smaller *T*, the smaller the probability P_f will be. On the other hand, a small value of *T* will negatively effect the false negative probability P_n,

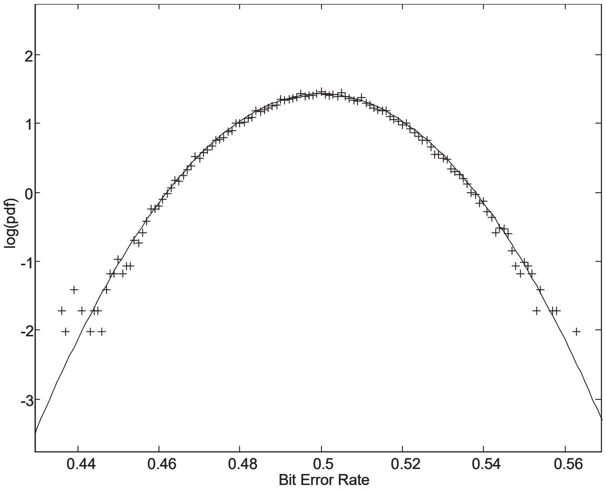

Figure 3. Comparison of the probability density function of the BER plotted as '+' and the normal distribution.

i.e. the probability that two signals are 'equal', but not identified as such.

In order to analyze the choice of this threshold T, we assume that the fingerprint extraction process yields random i.i.d. (independent and identically distributed) bits. The number of bit errors will then have a binomial distribution *(n,p)*, where n equals the number of bits extracted and p (= 0.5) is the probability that a '0' or '1' bit is extracted. Since n (= 8192 = 32 × 256) is large in our application, the binomial distribution can be approximated by a normal distribution with a mean $\mu = np$ and standard deviation $\sigma = \sqrt{np(1-p)}$. Given a fingerprint block F_1, the probability that a randomly selected fingerprint block F_2 has less than $T = \alpha n$ errors with respect to F_1 is given by:

$$P_f(\alpha)=\frac{1}{\sqrt{2\pi}}\int_{(1-2\alpha)\sqrt{n}}^{\infty} e^{-x^2/2}dx=\frac{1}{2}\operatorname{erfc}\left(\frac{(1-2\alpha)}{\sqrt{2}}\sqrt{n}\right) \tag{2}$$

where α denotes the Bit Error Rate (BER).

However, in practice the sub-fingerprints have high correlation along the time axis. This correlation is due not only to the inherent time correlation in audio, but also by the large overlap of the frames used in fingerprint extraction. Higher correlation implies a larger standard deviation, as shown by the following argument.

Assume a symmetric binary source with alphabet {-1,1} such that the probability that symbol x_i and symbol x_{i+1} are the same is equals to q. Then one may easily show that

$$\mathrm{E}[x_i x_{i+k}] = a^{|k|}, \tag{3}$$

where $a = 2 \cdot q - 1$. If the source Z is the exclusive-or of two such sequences X and Y, then Z is symmetric and

$$\mathrm{E}[z_i z_{i+k}] = a^{2|k|}. \tag{4}$$

For N large, the standard deviation of the average $\overline{Z}_N$ over N consecutive samples of Z can be approximately described by a normal distribution with mean 0 and standard deviation equal to

$$\sqrt{\frac{1+a^2}{N(1-a^2)}}. \tag{5}$$

Translating the above back to the case of fingerprints bits, a correlation factor a between subsequent fingerprint bits implies an increase in standard deviation for the BER by a factor

$$\sqrt{\frac{1+a^2}{1-a^2}}. \tag{6}$$

To determine the distribution of the BER with real fingerprint blocks a fingerprint database of 10,000 songs was generated. Thereafter the BER of 100,000 randomly selected pairs of fingerprint blocks were determined. The standard deviation of the resulting BER distribution was measured to be 0.0148, approximately 3 times higher than the 0.0055 one would expect from random i.i.d. bits.

Figure 3 shows the log Probability Density Function (PDF) of the measured BER distribution and a normal distribution with mean of 0.5 and a standard deviation of 0.0148. The PDF of the BER is a close approximation to the normal distribution. For BERs below 0.45 we observe some outliers, due to insufficient statistics. To incorporate the larger standard deviation of the BER distribution Formula (2) is modified by inclusion of a factor 3.

$$P_f(\alpha)=\frac{1}{2}\operatorname{erfc}\left(\frac{(1-2\alpha)}{3\sqrt{2}}\sqrt{n}\right) \tag{7}$$

The threshold for the BER used during experiments was $\alpha = 0.35$. This means that out of 8192 bits there must be less than 2867 bits in error in order to decide that the fingerprint blocks originate from the same song. Using formula (7) we arrive at a very low false positive rate of **erfc**(6.4)/2= $3.6 \cdot 10^{-20}$.

4.4 Experimental Robustness Results

In this subsection we show the experimental robustness of the proposed audio fingerprinting scheme. That is, we try to answer the question of whether or not the BER between the fingerprint block of an original and a degraded version of an audio clip remains under the threshold α (=0.35).

We selected four short audio excerpts (Stereo, 44.1kHz, 16bps) from songs that belong to different musical genres: *"O Fortuna"* by Carl Orff, *"Success has made a failure of our home"* by Sinead o'Connor, *"Say what you want"* by Texas and *"A whole lot of Rosie"* by AC/DC. All of the excerpts were subjected to the following signal degradations:

- **MP3 Encoding/Decoding** at 128 Kbps and 32 Kbps.
- **Real Media Encoding/Decoding** at 20 Kbps.
- **GSM Encoding at Full Rate** with an error-free channel and a channel with a carrier to interference (C/I) ratio of 4dB (comparable to GSM reception in a tunnel).
- **All-pass Filtering** using the system function: $H(z)=(0.81z^2-1.64z+1)/(z^2-1.64z+0.81)$.
- **Amplitude Compression** with the following compression ratios: 8.94:1 for $|A| \ge -28.6$ dB; 1.73:1 for -46.4 dB $< |A| < -28.6$ dB; 1:1.61 for $|A| \le -46.4$ dB.
- **Equalization** A typical10-band equalizer with the following settings:

Freq.(Hz)	31	62	125	250	500	1k	2k	4k	8k	16k
Gain(dB)	-3	+3	-3	+3	-3	+3	-3	+3	-3	+3

- **Band-pass Filtering** using a second order Butterworth filter with cut-off frequencies of 100Hz and 6000Hz.
- **Time Scale Modification** of +4% and -4% . Only the tempo changes, the pitch remains unaffected.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.