Open access • Journal Article • DOI:10.1007/S11265-005-4151-3

A Review of Audio Fingerprinting — Source link

Pedro Cano, Eloi Batlle, Ton Kalker, Jaap A. Haitsma

Institutions: Pompeu Fabra University, Philips

Published on: 01 Nov 2005 - Signal Processing Systems

Topics: Watermark, Fingerprint (computing), Speech coding and Digital watermarking

Journal of VLSI Signal Processing 41, 271–284, 2005

DOI: 10.1007/s11265-005-4151-3

A Review of Audio Fingerprinting

PEDRO CANO AND ELOI BATLLE
Music Technology Group, IUA Universitat Pompeu Fabra, Ocata, 8 08003 Barcelona, Spain

TON KALKER AND JAAP HAITSMA
Philips Research Laboratories Eindhoven, Prof. Holslaan 4, 5656 AA Eindhoven, The Netherlands

Abstract. An audio fingerprint is a compact content-based signature that summarizes an audio recording. Audio Fingerprinting technologies have attracted attention since they allow the identification of audio independently of its format and without the need of meta-data or watermark embedding. Other uses of fingerprinting include: integrity verification, watermark support and content-based audio retrieval. The different approaches to fingerprinting have been described with different rationales and terminology: Pattern matching, Multimedia (Music) Information Retrieval or Cryptography (Robust Hashing). In this paper, we review different techniques describing its functional blocks as parts of a common, unified framework.

Keywords: audio fingerprinting, content-based audio identification, watermarking, integrity verification, audio information retrieval, robust hashing

1. Introduction

Audio fingerprinting is best known for its ability to link unlabeled audio to corresponding meta-data (e.g. artist and song name), regardless of the audio format. Audio fingerprinting or content-based audio identification (CBID) systems extract a perceptual digest of a piece of audio content, i.e. a fingerprint and store it in a database. When presented with unlabeled audio, its fingerprint is calculated and matched against those stored in the database. Using fingerprints and matching algorithms, distorted versions of a recording can be identified as the same audio content.

A source of difficulty when automatically identifying audio content derives from its high dimensionality, the significant variance of the audio data for perceptually similar content and the necessity to efficiently compare the fingerprint with a huge collection of registered fingerprints. The simplest approach that one may think of—the direct comparison of the digitalized waveform—is neither efficient nor effective. A more efficient implementation of this approach could use a hash method, such as MD5 (Message Digest 5) or CRC (Cyclic Redundancy Checking), to obtain a compact representation of the binary file. In this setup, one compares the hash values instead of the whole files. However, hash values are fragile, a single bit flip is sufficient for the hash to completely change. Of course this setup is not robust to compression or minimal distortions of any kind and, in fact, it cannot be considered as content-based identification since it does not consider the content, understood as information, just the bits.

An ideal fingerprinting system should fulfill several requirements. It should be able to accurately identify an item, regardless of the level of compression and distortion or interference in the transmission channel. Depending on the application, it should be able

to identify the titles from excerpts of only a few seconds. The fingerprinting system should also be computationally efficient. Efficiency is critical in a real application both in the calculation of the fingerprint of the unknown audio and, even more so, in the search for a best match in huge repository of fingerprints. This computational cost is related to the size of the fingerprints, the complexity of the search algorithm and the complexity of the fingerprint extraction.

The design principles and needs behind audio fingerprinting are recurrent in several research areas. Compact signatures that represent complex multimedia objects are employed in Information Retrieval for fast indexing and retrieval. In order to index complex multimedia objects it is necessary to reduce their dimensionality (to avoid the "curse of dimensionality") and perform the indexing and searching in the reduced space [1–3]. In analogy to the cryptographic hash value, content-based digital signatures can be seen as evolved versions of hash values that are robust to content-preserving transformations [4, 5]. Also from a pattern matching point of view, the idea of extracting the essence of a class of objects retaining its main characteristics is at the heart of any classification system [6–10].

This paper aims to give a vision on it Audio Fingerprinting. The rationale along with the differences with respect to watermarking are presented in 2. The main requirements of fingerprinting systems are described in 3. The basic modes of employing audio fingerprints, namely identification, authentication, content-based secret key generation for watermarking and content-based audio retrieval and processing are commented in Section 4. We then present in Section 5 some concrete scenarios and business models where the technology is used. In the lasts sections of the article (from Section 6 to 10), we introduce the main contribution of the article: a general framework of audio fingerprinting systems. Although the framework focuses on identification, some of its functional blocks are common to content-based audio retrieval or integrity verification.

2. Definition of Audio Fingerprinting

An audio fingerprint is a compact content-based signature that summarizes an audio recording. Audio fingerprinting has attracted a lot of attention for its audio identification capabilities. Audio fingerprinting or content-based identification (CBID) technologies extract acoustic relevant characteristics of a piece of audio content and store them in a database. When presented with an unidentified piece of audio content, characteristics of that piece are calculated and matched against those stored in the database. Using fingerprints and matching algorithms, distorted versions of a single recording can be identified as the same music title [11].

The approach differs from an alternative existing solution to identify audio content: *Audio Watermarking*. In audio watermarking [12], research on psychoacoustics is conducted so that an arbitrary message, the watermark, can be embedded in a recording without altering the perception of the sound. The identification of a song title is possible by extracting the message embedded in the audio. In audio fingerprinting, the message is automatically derived from the perceptually most relevant components of sound. Compared to watermarking, it is ideally less vulnerable to attacks and distortions since trying to modify this message, the fingerprint, means alteration of the quality of the sound. It is also suitable to deal with legacy content, that is, with audio material released without watermark. In addition, it requires no modification of the audio content. As a drawback, the computational complexity of fingerprinting is generally higher than watermarking and there is the need of a connection to a fingerprint repository. In addition, contrary to watermarking, the message is not independent from the content. It is therefore for example not possible to distinguish between perceptually identical copies of a recording. Just like with watermarking technology, there are more uses to fingerprinting than identification. Specifically, it can also be used for verification of content-integrity; similarly to fragile watermarks.

At this point, we should clarify that the term "fingerprinting" has been employed for many years as a special case of watermarking devised to keep track of an audio clip's usage history. Watermark fingerprinting consists in uniquely watermarking each legal copy of a recording. This allows to trace back to the individual who acquired it [13]. However, the same term has been used to name techniques that associate an audio signal to a much shorter numeric sequence (the "fingerprint") and use this sequence to e.g. identify the audio signal. The latter is the meaning of the term "fingerprinting" in this article. Other terms for audio fingerprinting are robust matching, robust or perceptual hashing, passive watermarking, automatic music recognition, content-based digital signatures and content-based audio identification. The areas

relevant to audio fingerprinting include information retrieval, pattern matching, signal processing, databases, cryptography and music cognition to name a few.

3. Properties of Audio Fingerprinting

The requirements depend heavily on the application but are useful in order to evaluate and compare different audio fingerprinting technologies. In their *Request for Information on Audio Fingerprinting Technologies* [11], the IFPI (International Federation of the Phonographic Industry) and the RIAA (Recording Industry Association of America) tried to evaluate several identification systems. Such systems have to be computationally efficient and robust. A more detailed enumeration of requirements can help to distinguish among the different approaches [14, 15]:

Accuracy: The number of correct identifications, missed identifications, and wrong identifications (false positives).

Reliability: Methods for assessing that a query is present or not in the repository of items to identify is of major importance in play list generation for copyright enforcement organizations. In such cases, if a song has not been broadcast, it should not be identified as a match, even at the cost of missing actual matches. In other applications, like automatic labeling of MP3 files (see Section 6), avoiding false positives is not such a mandatory requirement.

Robustness: Ability to accurately identify an item, regardless of the level of compression and distortion or interference in the transmission channel. Other sources of degradation are pitching, equalization, background noise, D/A-A/D conversion, audio coders (such as GSM and MP3), etc.

Granularity: Ability to identify whole titles from excerpts a few seconds long. It requires to deal with shifting, that is lack of synchronization between the extracted fingerprint and those stored in the database and it adds complexity to the search (it needs to compare audio in all possible alignments).

Security: Vulnerability of the solution to cracking or tampering. In contrast with the robustness requirement, the manipulations to deal with are designed to fool the fingerprint identification algorithm.

Versatility: Ability to identify audio regardless of the audio format. Ability to use the same database for different applications.

Scalability: Performance with very large databases of titles or a large number of concurrent identifications. This affects the accuracy and the complexity of the system.

Complexity: It refers to the computational costs of the fingerprint extraction, the size of the fingerprint, the complexity of the search, the complexity of the fingerprint comparison, the cost of adding new items to the database, etc.

Fragility: Some applications, such as content-integrity verification systems, may require the detection of changes in the content. This is contrary to the robustness requirement, as the fingerprint should be robust to content-preserving transformations but not to other distortions (see Section 4.2).

Improving a certain requirement often implies losing performance in some other. Generally, the fingerprint should be:

- A perceptual digest of the recording. The fingerprint must retain the maximum of acoustically relevant information. This digest should allow the discrimination over a large number of fingerprints. This may be conflicting with other requirements, such as complexity and robustness.
- Invariant to distortions. This derives from the robustness requirement. Content-integrity applications, however, relax this constraint for content-preserving distortions in order to detect deliberate manipulations.
- Compact. A small-sized representation is interesting for complexity, since a large number (maybe millions) of fingerprints need to be stored and compared. An excessively short representation, however, might not be sufficient to discriminate among recordings, affecting thus accuracy, reliability and robustness.
- Easily computable. For complexity reasons, the extraction of the fingerprint should not be excessively time-consuming.

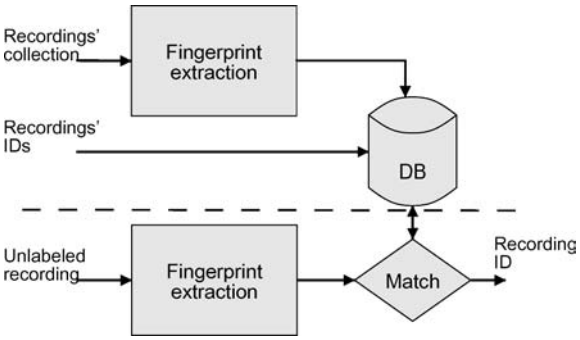

Figure 1. Content-based audio identification framework.

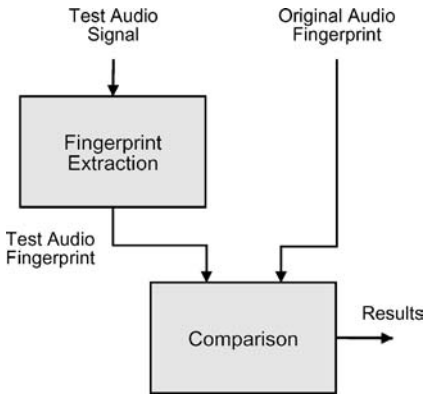

Figure 2. Integrity verification framework.

4. Usage Modes

4.1. Identification

Independently of the specific approach to extract the content-based compact signature, a common architecture can be devised to describe the functionality of fingerprinting when used for identification [11].

The overall functionality mimics the way humans perform the task. As seen in Fig. 1, a memory of the recordings to be recognized is created off-line (top); in the identification mode (bottom), unlabeled audio is presented to the system to look for a match.

Database creation: The collection of recordings to be recognized is presented to the system for the extraction of their fingerprint. The fingerprints are stored in a database and can be linked to a tag or other meta-data relevant to each recording.

Identification: The unlabeled recording is processed in order to extract a fingerprint. The fingerprint is subsequently compared with the fingerprints in the database. If a match is found, the tag associated with the recording is obtained from the database. Optionally, a reliability measure of the match can be provided.

4.2. Integrity Verification

Integrity verification aims at detecting the alteration of data. The overall functionality (see Fig. 2) is similar to identification. First, a fingerprint is extracted from the original audio. In the verification phase, the fingerprint extracted from the test signal is compared with the fingerprint of the original. As a result, a report indicating whether the signal has been manipulated is output. Optionally, the system can indicate the type of manipulation and where in the audio it occurred. The verification data, which should be significantly smaller than the audio data, can be sent along with the original audio data (e.g. as a header) or stored in a database. A technique known as *self-embedding* avoids the need of a database or a special dedicated header, by embedding the content-based signature into the audio data using watermarking (see Fig. 3). An example of such a system is described in [16].

4.3. Watermarking Support

Audio fingerprinting can assist watermarking. Audio fingerprints can be used to derive secret keys from the actual content. As described by Mihçak et al. [5], using the same secret key for a number of different audio items may compromise security, since each item may leak partial information about the key. Audio fingerprinting/perceptual hashing can help generate input-dependent keys for each piece of audio. Haitsma et al. [4] suggest audio fingerprinting to enhance the security of watermarks in the context of copy attacks. Copy attacks estimate a watermark from watermarked content and transplant it to unmarked content. Binding the watermark to the content can help to defeat this type of attacks. In addition, fingerprinting can be useful against insertion/deletion attacks that cause desynchronization of the watermark detection: by using the fingerprint, the detector is able to find anchor points in the audio stream and thus to resynchronize at these locations [5].

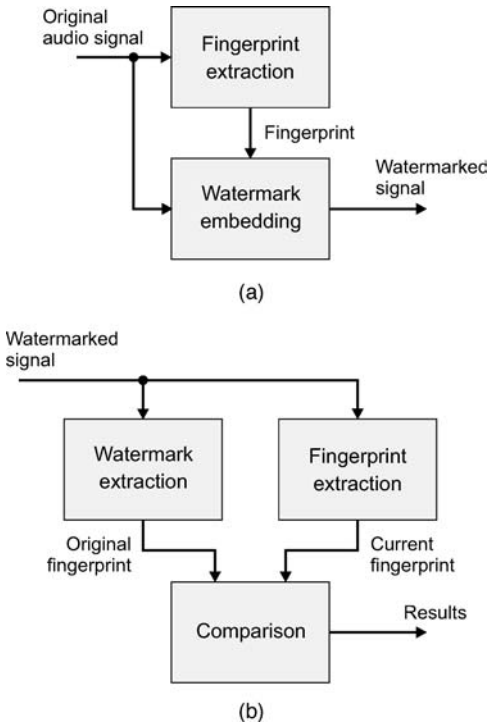

Figure 3. Self-embedding integrity verification framework: (a) fingerprint embedding and (b) fingerprint comparison.

4.4. Content-Based Audio Retrieval and Processing

Deriving compact signatures from complex multimedia objects is an essential step in Multimedia Information Retrieval. Fingerprinting can extract information from the audio signal at different abstraction levels, from low level descriptors to higher level descriptors. Especially, higher level abstractions for modeling audio hold the possibility to extend the fingerprinting usage modes to content-based navigation, search by similarity, content-based processing and other applications of Music Information Retrieval. In a query-by-example scheme, the fingerprint of a song can be used to retrieve not only the original version but also "similar" ones [17].

5. Application Scenarios

Most of the applications presented in this section are particular cases of the identification usage mode described above. They are therefore based on the ability of audio fingerprinting to link unlabeled audio to corresponding meta-data, regardless of audio format.

5.1. Audio Content Monitoring and Tracking

5.1.1. Monitoring at the Distributor End. Content distributors may need to know whether they have the rights to broadcast certain content to consumers. Fingerprinting helps identify unlabeled audio in TV and Radio channels repositories. It can also identify unidentified audio content recovered from CD plants and distributors in anti-piracy investigations (e.g. screening of master recordings at CD manufacturing plants) [11].

5.1.2. Monitoring at the Transmission Channel. In many countries, radio stations must pay royalties for the music they air. Rights holders are eager to monitor radio transmissions in order to verify whether royalties are being properly paid. Even in countries where radio stations can freely air music, rights holders are interested in monitoring radio transmissions for statistical purposes. Advertisers are also willing to monitor radio and TV transmissions to verify whether commercials are being broadcast as agreed. The same is true for web broadcasts. Other uses include chart compilations for statistical analysis of program material or enforcement of "cultural laws" (e.g. in France a certain percentage of the aired recordings needs to be in French). Fingerprinting-based monitoring systems can be and are actually being used for this purpose. The system "listens" to the radio and continuously updates a play list of songs or commercials broadcast by each station. Of course, a database containing fingerprints of all songs and commercials to be identified must be available to the system, and this database must be updated as new songs come out. Examples of commercial providers of such services are: Broadcast Data System (www.bdsonline.com), Music Reporter (www.musicreporter.net), Audible Magic (www.audiblemagic.com), Yacast (www.yacast.fr).

Napster and Web-based communities alike, where users share music files, have proved to be excellent channels for music piracy. After a court battle with the recording industry, Napster was enjoined from facilitating the transfer of copyrighted music. The first measure taken to conform with the judicial ruling was the introduction of a filtering system based on filename analysis, according to lists of copyrighted music recordings supplied by the recording companies. This simple system did not solve the problem, as users proved to be extremely creative in choosing file names that deceived the filtering system while still allowing

other users to easily recognize specific recordings. The large number of songs with identical titles was an additional factor in reducing the efficiency of such filters. Fingerprinting-based monitoring systems constitute a well-suited solution to this problem. Napster actually adopted a fingerprinting technology (see www.relatable.com) and a new file-filtering system relying on it. Additionally, audio content can be found in ordinary web pages. Audio fingerprinting combined with a web crawler can identify this content and report it to the corresponding right owners (e.g. www.baytsp.com).

5.1.3. Monitoring at the Consumer End. In usage-policy monitoring applications, the goal is to avoid misuse of audio signals by the consumer. We can conceive a system where a piece of music is identified by means of a fingerprint and a database is contacted to retrieve information about the rights. This information dictates the behavior of compliant devices (e.g. CD and DVD players and recorders, MP3 players or even computers) in accordance with the usage policy. Compliant devices are required to be connected to a network in order to access the database.

5.2. Added-Value Services

Content information is defined as information about an audio excerpt that is relevant to the user or necessary for the intended application. Depending on the application and the user profile, several levels of content information can be defined. Here are some of the situations we can imagine:

- Content information describing an audio excerpt, such as rhythmic, timbrical, melodic or harmonic descriptions.
- Meta-data describing a musical work, how it was composed and how it was recorded. For example: composer, year of composition, performer, date of performance, studio recording/live performance.
- Other information concerning a musical work, such as album cover image, album price, artist biography, information on the next concerts, etc.

Some systems store content information in a database that is accessible through the Internet. Fingerprinting can then be used to identify a recording and retrieve the corresponding content information, regardless of support type, file format or any other particularity of the audio data. For example, MusicBrainz, Id3man or Moodlogic (www.musicbrainz.org, www.id3man.com, www.moodlogic.com) automatically label collections of audio files; the user can download a compatible player that extracts fingerprints and submits them to a central server from which meta data associated to the recordings is downloaded. Gracenote (www.gracenote.com), who has been providing linking to music meta-data based on the TOC (Table of Contents) of a CD, recently offered audio fingerprinting technology to extend the linking from CD's table of contents to the song level. Their audio identification method is used in combination with text-based classifiers to enhance the accuracy.

Another example is the identification of an audio excerpt by mobile devices, e.g. a cell phone; this is one of the most demanding situations in terms of robustness, as the audio signal goes through radio distortion, D/A-A/D conversion, background noise and GSM coding, and only a few seconds of audio are available (e.g. www.shazam.com).

5.3. Integrity Verification Systems

In some applications, the integrity of audio recordings must be established before the signal can actually be used, i.e. one must assure that the recording has not been modified or that it is not too distorted. If the signal undergoes lossy compression, D/A-A/D conversion or other content-preserving transformations in the transmission channel, integrity cannot be checked by means of standard hash functions, since a single bit flip is sufficient for the output of the hash function to change. Methods based on fragile watermarking can also provide false alarms in such a context. Systems based on audio fingerprinting, sometimes combined with watermarking, are being researched to tackle this issue. Among some possible applications [16], we can name: Check that commercials are broadcast with the required length and quality, verify that a suspected infringing recording is in fact the same as the recording whose ownership is known, etc.

6. General Framework

In spite of the different rationales behind the identification task, methods share certain aspects. As depicted in Fig. 6, there are two fundamental processes: the finger-

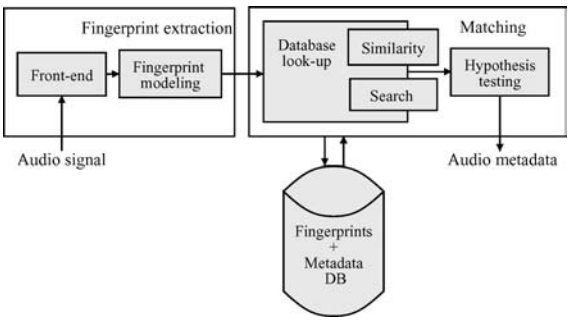

Figure 4. Content-based audio identification framework.

print extraction and the matching algorithm. The fingerprint extraction derives a set of relevant perceptual characteristics of a recording in a concise and robust form. The fingerprint requirements include:

- Discrimination power over huge numbers of other fingerprints,
- Invariance to distortions,
- Compactness,
- Computational simplicity.

The solutions proposed to fulfill the above requirements imply a trade-off between dimensionality reduction and information loss. The fingerprint extraction consists of a front-end and a fingerprint modeling block (see Fig. 5). The front-end computes a set of measurements from the signal (see Section 7). The fingerprint model block defines the final fingerprint representation, e.g: a vector, a trace of vectors, a codebook, a sequence of indexes to HMM sound classes, a sequence of error correcting words or musically meaningful high-level attributes (see Section 8).

Given a fingerprint derived from a recording, the matching algorithm searches a database of fingerprints to find the best match. A way of comparing fingerprints, that is a similarity measure, is therefore needed (see Section 9.1). Since the number of fingerprint comparisons is high in a large database and the similarity can be expensive to compute, we require methods that speed up the search. Some fingerprinting systems use a simpler similarity measure to quickly discard candidates and the more precise but expensive similarity measure for the reduced set of candidates. There are also methods that pre-compute some distances off-line and build a data structure that allows reducing the number of computations to do on-line (see Section 9.2). According to [1], good searching methods should be:

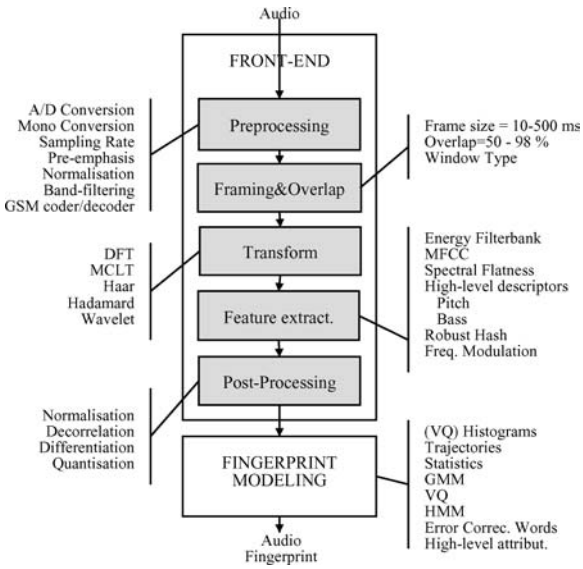

Figure 5. Fingerprint extraction framework: Front-end (*top*) and fingerprint modeling (*bottom*).

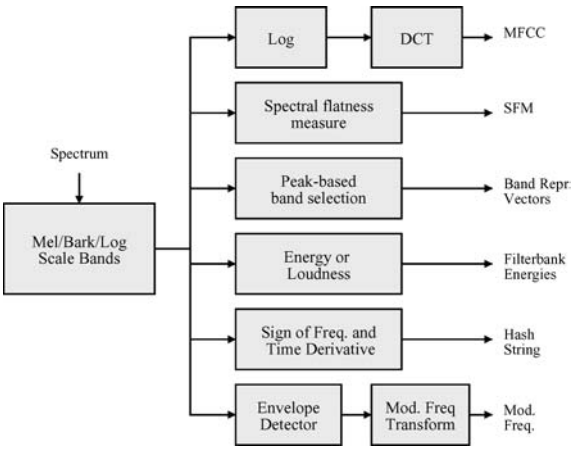

Figure 6. Feature extraction examples.

- Fast: Sequential scanning and similarity calculation can be too slow for huge databases.

- Correct: Should return the qualifying objects, without missing any—i.e. low False Rejection Rate (FRR).
- Memory efficient: The memory overhead of the search method should be relatively small.
- Easily updatable: Insertion, deletion and updating of objects should be easy.

The last block of the system—the hypothesis testing (see Fig. 6)—computes a reliability measure indicating how confident the system is about an identification (see Section 10).

7. Front-End

The front-end converts an audio signal into a sequence of relevant features to feed the fingerprint model block (see Fig. 6). Several driving forces co-exist in the design of the front-end:

- Dimensionality reduction
- Perceptually meaningful parameters (similar to those used by the human auditory system)
- Invariance/robustness (to channel distortions, background noise, etc.)
- Temporal correlation (systems that capture spectral dynamics).

In some applications, where the audio to identify is coded, for instance in mp3, it is possible to by-pass some blocks and extract the features from the audio coded representation.

7.1. Preprocessing

In a first step, the audio is digitalized (if necessary) and converted to a general format, e.g: mono PCM (16 bits) with a fixed sampling rate (ranging from 5 to 44.1 KHz). Sometimes the audio is preprocessed to simulate the channel, e.g: band-pass filtered in a telephone identification task. Other types of processing are a GSM coder/decoder in a mobile phone identification system, pre-emphasis, amplitude normalization (bounding the dynamic range to $(-1,1)$).

7.2. Framing and Overlap

A key assumption in the measurement of characteristics is that the signal can be regarded as stationary over an interval of a few milliseconds. Therefore, the signal is divided into frames of a size comparable to the variation velocity of the underlying acoustic events. The number of frames computed per second is called frame rate. A tapered window function is applied to each block to minimize the discontinuities at the beginning and end. Overlap must be applied to assure robustness to shifting (i.e. when the input data is not perfectly aligned to the recording that was used for generating the fingerprint). There is a trade-off between the robustness to shifting and the computational complexity of the system: the higher the frame rate, the more robust to shifting the system is but at a cost of a higher computational load.

7.3. Linear Transforms: Spectral Estimates

The idea behind linear transforms is the projection of the set of measurements to a new set of features. If the transform is suitably chosen, the redundancy is significantly reduced. There are optimal transforms in the sense of information packing and decorrelation properties, like Karhunen-Loève (KL) or Singular Value Decomposition (SVD) [9]. These transforms, however, are problem dependent and computationally complex. For that reason, lower complexity transforms using fixed basis vectors are common. Most CBID methods therefore use standard transforms from time to frequency domain to facilitate efficient compression, noise removal and subsequent processing. Lourens [18], (for computational simplicity), and Kurth et al. [19], (to model highly distorted sequences, where the time-frequency analysis exhibits distortions), use power measures. The power can still be seen as a simplified time-frequency distribution, with only one frequency bin.

The most common transformation is the Discrete Fourier Transform (DFT). Some other transforms have been proposed: the Discrete Cosine Transform (DCT), the Haar Transform or the Walsh-Hadamard Transform [2]. Richly et al. did a comparison of the DFT and the Walsh-Hadamard Transform that revealed that the DFT is generally less sensitive to shifting [20]. The Modulated Complex Transform (MCLT) used by Mihçak et al. [5] and also by Burges et al. [21] exhibits approximate shift invariance properties [5].

7.4. Feature Extraction

Once on a time-frequency representation, additional transformations are applied in order to generate the final acoustic vectors. In this step, we find a great diversity of algorithms. The objective is again to reduce the dimensionality and, at the same time, to increase the invariance to distortions. It is very common to include knowledge of the transduction stages of the human auditory system to extract

more perceptually meaningful parameters. Therefore, many systems extract several features performing a critical-band analysis of the spectrum (see Fig. 3). In [6, 22], Mel-Frequency Cepstrum Coefficients (MFCC) are used. In [7], the choice is the Spectral Flatness Measure (SFM), which is an estimation of the tone-like or noise-like quality for a band in the spectrum. Papaodysseus et al. [23] presented the "band representative vectors", which are an ordered list of indexes of bands with prominent tones (i.e. with peaks with significant amplitude). Energy of each band is used by Kimura et al. [3]. Haitsma et al. use the energies of 33 bark-scaled bands to obtain their "hash string," which is the sign of the energy band differences (both in the time and the frequency axis) [4].

Sukittanon and Atlas claim that spectral estimates and related features only are inadequate when audio channel distortion occurs [8]. They propose modulation frequency analysis to characterize the time-varying behavior of audio signals. In this case, features correspond to the geometric mean of the modulation frequency estimation of the energy of 19 bark-spaced band-filters.

Approaches from music information retrieval include features that have proved valid for comparing sounds: harmonicity, bandwidth, loudness [22].

Burges et al. point out that the features commonly used are heuristic, and as such, may not be optimal [21]. For that reason, they use a modified Karhunen-Loéve transform, the Oriented Principal Component Analysis (OPCA), to find the optimal features in an "unsupervised" way. If PCA (KL) finds a set of orthogonal directions which maximize the signal variance, OPCA obtains a set of possible non-orthogonal directions which take some predefined distortions into account.

7.5. Post-Processing

Most of the features described so far are absolute measurements. In order to better characterize temporal variations in the signal, higher order time derivatives are added to the signal model. In [6] and [24], the feature vector is the concatenation of MFCCs, their derivative (delta) and the acceleration (delta-delta), as well as the delta and delta-delta of the energy. Some systems only use the derivative of the features, not the absolute features [7, 19]. Using the derivative of the signal measurements tends to amplify noise [10] but, at the same time, filters the distortions produced in linear time invariant, or slowly varying channels (like an equalization). Cepstrum Mean Normalization (CMN) is used to reduce linear slowly varying channel distortions in [24]. If Euclidean distance is used (see Section 9.1), mean subtraction and component wise variance normalization are advisable. Some systems compact the feature vector representation using transforms (e.g. PCA [6, 24]).

It is quite common to apply a very low resolution quantization to the features: ternary [20] or binary [4, 19]. The purpose of quantization is to gain robustness against distortions [4, 19], normalize [20], ease hardware implementations, reduce the memory requirements and for convenience in subsequent parts of the system. Binary sequences are required to extract error correcting words utilized in [5, 19]. In [5], the discretization is designed to increase randomness in order to minimize fingerprint collision probability.

8. Fingerprint Models

The fingerprint modeling block usually receives a sequence of feature vectors calculated on a frame by frame basis. Exploiting redundancies in the frame time vicinity, inside a recording and across the whole database, is useful to further reduce the fingerprint size. The type of model chosen conditions the similarity measure and also the design of indexing algorithms for fast retrieval (see Section 9).

A very concise form of fingerprint is achieved by summarizing the multidimensional vector sequences of a whole song (or a fragment of it) in a single vector. Etantrum [25] calculates the vector out of the means and variances of the 16 bank-filtered energies corresponding to 30 s of audio ending up with a signature of 512 bits. The signature along with information on the original audio format is sent to a server for identification. Musicbrainz' TRM signature [26] includes in a vector: the average zero crossing rate, the estimated beats per minute (BPM), an average spectrum and some more features to represent a piece of audio (corresponding to 26 s). The two examples above are computationally efficient and produce a very compact fingerprint. They have been designed for applications like linking mp3 files to meta-data (title, artist, etc.) and are more tuned for low complexity (both on the client and the server side) than for robustness (cropping or broadcast streaming audio).

Fingerprints can also be sequences (traces, trajectories) of features. This fingerprint representation is found in [22], and also in [4] as binary vector se-

quences. The fingerprint in [23], which consists on a sequence of "band representative vectors," is binary encoded for memory efficiency.

Some systems, include high-level musically meaningful attributes, like rhythm ([28]) or prominent pitch (see [22, 26]).

Following the reasoning on the possible suboptimality of heuristic features, Burges et al. [21] employ several layers of OPCA to decrease the local statistical redundancy of feature vectors with respect to time. Besides reducing dimensionality, extra robustness requisites to shifting and pitching are accounted in the transformation.

"Global redundancies" within a song are exploited in [7]. If we assume that the features of a given audio item are similar among them (e.g: a chorus that repeats in a song probably hold similar features), a compact representation can be generated by clustering the feature vectors. The sequence of vectors is thus approximated by a much lower number of representative code vectors, a codebook. The temporal evolution of audio is lost with this approximation. Also in [7], short-time statistics are collected over regions of time. This results in both higher recognition, since some temporal dependencies are taken into account, and a faster matching, since the length of each sequence is also reduced.

Cano [6] and [24] use a fingerprint model that further exploits global redundancy. The rationale is very much inspired on speech research. In speech, an alphabet of sound classes, i.e. phonemes can be used to segment a collection of raw speech data into text achieving a great redundancy reduction without "much" information loss. Similarly, we can view a corpus of music, as sentences constructed concatenating sound classes of a finite alphabet. "Perceptually equivalent" drum sounds, say for instance a hi-hat, occurs in a great number of pop songs. This approximation yields a fingerprint which consists in sequences of indexes to a set of sound classes representative of a collection of recordings. The sound classes are estimated via unsupervised clustering and modeled with Hidden Markov Models (HMMs). Statistical modeling of the signal's time course allows local redundancy reduction. The fingerprint representation as sequences of indexes to the sound classes retains the information on the evolution of audio through time.

In [5], discrete sequences are mapped to a dictionary of error correcting words. In [19], the error correcting codes are at the basis of their indexing method.

9. Similarity Measures and Searching Methods

9.1. Similarity Measures

Similarity measures are very much related to the type of model chosen. When comparing vector sequences, a correlation metric is common. The Euclidean distance, or slightly modified versions that deal with sequences of different lengths, are used for instance in [22]. In [8], the classification is Nearest Neighbor using a cross entropy estimation. In the systems where the vector feature sequences are quantized, a Manhattan distance (or Hamming when the quantization is binary) is common [4, 20]. Mihçak et al. [5] suggest that another error metric, which they call "Exponential Pseudo Norm" (EPN), could be more appropriate to better distinguish between close and distant values with an emphasis stronger than linear.

So far we have presented an identification framework that follows a template matching paradigm [9]: both the reference patterns—the fingerprints stored in the database—and the test pattern—the fingerprint extracted from the unknown audio—are in the same format and are compared according to some similarity measure, e.g: hamming distance, a correlation and so on. In some systems, only the reference items are actually "fingerprints"—compactly modeled as a codebook or a sequence of indexes to HMMs [7, 24]. In these cases, the similarities are computed directly between the feature sequence extracted from the unknown audio and the reference audio fingerprints stored in the repository. In [7], the feature vector sequence is matched to the different codebooks using a distance metric. For each codebook, the errors are accumulated. The unknown item is assigned to the class which yields the lowest accumulated error. In [24], the feature sequence is run against the fingerprints (a concatenation of indexes pointing at HMM sound classes) using the Viterbi algorithm. The most likely passage in the database is selected.

9.2. Searching Methods

A fundamental issue for the usability of a fingerprinting system is how to efficiently do the comparison of the unknown audio against the possibly millions of fingerprints. A brute-force approach that computes the similarities between the unknown recording's fingerprint and those stored in the database can be prohibitory. The

time for finding a best match in this linear or sequential approach is proportional to $Nc(d()) + E$, where N is the number of fingerprints in the repository and $c(d())$ the time needed for a single similarity calculation and E accounts for some extra CPU time.

9.2.1. Pre-Computing Distances Off-Line. One cannot pre-calculate off-line similarities with query fingerprint because the fingerprint has not been previously presented to the system. How ever one can precompute distances among the fingerprints registered in the repository and build a data structure to reduce the number of similarity evaluations once the query is presented. It is possible to build sets of equivalence classes off-line, calculate some similarities on-line to discard some classes and search exhaustively the rest (see for example [3]). If the similarity measure is a metric, i.e. the similarity measure is a function that satisfies the following properties: positiveness, symmetry, reflexivity and the triangular inequality, there are methods that reduce the number of similarity evaluations and guarantee no false dismissals (see [29]). Vector spaces allow the use of efficient existing spatial access methods [30].

9.2.2. Filtering Unlikely Candidates with a Cheap Similarity Measure. Another possibility is to use a simpler similarity measure to quickly eliminate many candidates and the more precise but complex on the rest, e.g. in [31, 32]. As demonstrated in [30], in order to guarantee no false dismissals, the simple (coarse) similarity used for discarding unpromising hypothesis must lower bound the more expensive (fine) similarity.

9.2.3. Inverted File Indexing. A very efficient searching method is the use of inverted files indexing. Haitsma et al. proposed an index of possible pieces of a fingerprint that points to the positions in the songs. Provided that a piece of a query's fingerprint is free of errors (exact match), a list of candidate songs and positions can be efficiently retrieved to exhaustively search through [4]. In [6], indexing and heuristics similar to those used in computational biology for the comparison of DNA are used to speed up a search in a system where the fingerprints are sequences of symbols. Kurth et al. [19] present an index that use code words extracted from binary sequences representing the audio. Sometimes this approaches, although very fast, make assumptions on the errors permitted in the words used to build the index which could result in false dismissals.

9.2.4. Candidate Pruning. A simple optimization to speed up the search is to keep the best score encountered thus far. We can abandon a similarity measure calculation if at one point we know we are not going to improve the best-so-far score (see for instance [3]). Some similarity measures can profit from structures like suffix trees to avoid duplicate calculations [1]. Miller et al. [27] propose a tree to avoid redundancies in the calculation of the best-match in a framework built on the fingerprint representation of [4]. Combining the tree structure with a "best-so-far" heuristic avoids not only current fingerprint similarity computation but also all the fingerprints that have a common starting.

9.2.5. Other Approaches. In one of the setups of [33], the repository of fingerprints is split into two databases. The first and smaller repository holds fingerprints with higher probability of appearance, e.g. the most popular songs of the moment, and the other repository with the rest. The queries are confronted first with the small and more likely repository and only when no match is found does the system examine the second database. Production systems actually use several of the above depicted speed-up methods. Wang and Smith [33] for instance, besides searching first in the most popular songs repository, uses an inverted file indexing for fast accessing the fingerprints along with a heuristic to filter out unpromising candidates before it exhaustively searches with the more precise similarity measure.

10. Hypothesis Testing

This last step aims to answer whether the query is present or not in the repository of items to identify. During the comparison of the extracted fingerprint to the database of fingerprints, scores (resulting from similarity measures) are obtained. In order to decide that there is a correct identification, the score needs to be beyond a certain threshold. It is not easy to choose a threshold since it depends on: the used fingerprint model, the discriminative information of the query, the similarity of the fingerprints in the database, and the database size. The larger the database, the higher the probability of wrongly indicating a match by chance, that is a false positive. The false positive rate is also named false acceptance rate (FAR) or false alarm rate. The false negative rate appears also under the name of false rejected rate (FRR). The nomenclature is related to the Information Retrieval performance evaluation measures: Precision and Recall [1]. Approaches to

deal with false positives have been explicitly treated for instance in [4, 18, 34].

11. Summary

We have presented a review of the research carried out in the area of audio fingerprinting. Furthermore a number of applications which can benefit from audio fingerprinting technology were discussed. An audio fingerprinting system generally consists of two components: an algorithm to generate fingerprints from recordings and algorithm to search for a matching fingerprint in a fingerprint database. We have shown that although different researchers have taken different approaches, the proposals more or less fit in a general framework. In this framework, the fingerprint extraction includes a front-end where the audio is divided into frames and a number of discriminative and robust features is extracted from each frame. Subsequently these features are transformed to a fingerprint by a fingerprint modeling unit which further compacts the fingerprint representation. The searching algorithm finds the best matching fingerprint in a large repository according to some similarity measure. In order to speed up the search process and avoid a sequential scanning of the database, strategies are used to quickly eliminate non-matching fingerprints. A number of the discussed audio fingerprinting algorithms are currently commercially deployed, which shows the significant progress that has been made in this research area. There is, of course, room for improvement in the quest for more compact, robust and discriminative fingerprints and efficient searching algorithms. It also needs to be seen how the identification framework can be extended to browsing and similarity retrieval of audio collections.

References

1. R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, 1999.
2. S. Subramanya, R. Simha, B. Narahari, and A. Youssef, "Transform-Based Indexing of Audio Data for Multimedia Databases," in *Proc. of Int. Conf. on Computational Intelligence and Multimedia Applications*, New Delhi, India, Sept. 1999.
3. A. Kimura, K. Kashino, T. Kurozumi, and H. Murase, "Very Quick Audio Searching: Introducing Global Pruning to the Time-Series Active Search," in *Proc. of Int. Conf. on Computational Intelligence and Multimedia Applications*, Salt Lake City, Utah, May 2001.
4. J. Haitsma and T. Kalker, "A Highly Robust Audio Fingerprinting System," in *Proceedings of the International Symposium on Music Information Retrieval*, Paris, France, 2002.
5. M. Mihçak and R. Venkatesan, "A Perceptual Audio Hashing Algorithm: A Tool for Robust Audio Identification and Information Hiding," in *4th Int. Information Hiding Workshop*, Pittsburg, PA, April 2001.
6. P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust Sound Modeling for Song Detection in Broadcast Audio," in *Proc. AES 112th Int. Conv.*, Munich, Germany, May 2002.
7. E. Allamanche, J. Herre, O. Helmuth, B. Fröba, T. Kasten, and M. Cremer, "Content-Based Identification of Audio Material Using Mpeg-7 Low Level Description," in *Proc. of the Int. Symp. of Music Information Retrieval*, Indiana, USA, Oct. 2001.
8. S. Sukittanon and L. Atlas, "Modulation Frequency Features for Audio Fingerprinting," in *Proc. of the ICASSP*, May 2002.
9. S. Theodoris and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.
10. J. Picone, "Signal Modeling Techniques in Speech Recognition," *Proc. of the ICASSP*, vol. 81, no. 9, 1993, pp. 1215–1247.
11. Request for information on audio fingerprinting technologies (2001) [Online]. Available: [http://www.riaa.org/pdf/RIAA_IFPI_Fingerprinting_RFI.pdf]
12. L. Boney, A. Tewfik, and K. Hamdy, "Digital Watermarks for Audio Signals," in *IEEE Proceedings Multimedia*, 1996, pp. 473–480.
13. S. Craver, W.M., and B. Liu, "What Can We Reasonably Expect from Watermarks?" in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, Oct. 2001.
14. Audio identification technology overview. (2002) [Online]. Available: [http://www.audiblemagic.com/about]
15. T. Kalker, "Applications and Challenges for Audio Fingerprinting," in *Presentation at the 111th AES Convention*, New York, 2001.
16. E. Gómez, P. Cano, L. de C.T. Gomes, E. Batlle, and M. Bonnet, "Mixed Watermarking-Fingerprinting Approach for Integrity Verification of Audio Recordings," in *Proceedings of the International Telecommunications Symposium*, Natal, Brazil, Sept. 2002.
17. P. Cano, M. Kaltenbrunner, F. Gouyon, and E. Batlle, "On the Use of Fastmap for Audio Information Retrieval," in *Proceedings of the International Symposium on Music Information Retrieval*, Paris, France, 2002.
18. J. Lourens, "Detection and Logging Advertisements Using its Sound," in *Proc. of the COMSIG*, Johannesburg, 1990.
19. F. Kurth, A. Ribbrock, and M. Clausen, "Identification of Highly Distorted Audio Material for Querying Large Scale Databases," in *Proc. AES 112th Int. Conv.*, Munich, Germany, May 2002.
20. G. Richly, L. Varga, F. Kovàs, and G. Hosszú, "Short-Term Sound Stream Characterisation for Reliable, Real-Time Occurrence Monitoring of Given Sound-Prints," in *Proc. 10th Mediterranean Electrotechnical Conference, MEleCon*, 2000.
21. C. Burges, J. Platt, and S. Jana, "Extracting Noise-Robust Features from Audio Data," in *Proc. of the ICASSP*, Florida, USA, May 2002.
22. T. Blum, D. Keislar, J. Wheaton, and E. Wold, "Method and Article of Manufacture for Content-Based Analysis, Storage, Retrieval and Segmentation of Audio Information," U.S. Patent 5,918,223, June 1999.

23. C. Papaodysseus, G. Roussopoulos, D. Fragoulis, T. Panagopoulos, and C. Alexiou, "A New Approach to the Automatic Recognition of Musical Recordings," *J. Audio Eng. Soc.*, vol. 49, no. 1/2, 2001, pp. 23–35.
24. E. Batlle, J. Masip, and E. Guaus, "Automatic Song Identification in Noisy Broadcast Audio," in *Proc. of the SIP*, Aug. 2002.
25. Etantrum (2002) [Online]. Available: [http://www.freshmeat.net/projects/songprint].
26. Musicbrainz trm.(2002) musicbrainz-1.1.0.tar.gz. [Online]. Available: [http://ftp.musicbrainz.org/pub/musicbrainz].
27. M. Miller, M. Rodriguez, and I. Cox, "Audio Fingerprinting: Nearest Neighbor Search in High Dimensional Binary Spaces," in *5th IEEE Int. Workshop on Multimedia Signal Processing: Special session on Media Recognition*, US Virgin Islands, USA, Dec. 2002.
28. D. Kirovski and H. Attias, "Beat-id: Identifying Music via Beat Analysis," in *5th IEEE Int. Workshop on Multimedia Signal Processing: Special session on Media Recognition*, US Virgin Islands, USA, Dec. 2002.
29. E. Chávez, G. Navarro, R.A. Baeza-Yates, and J.L. Marroquin, "Searching in Metric Spaces," *ACM Computing Surveys*, vol. 33, no. 3, 2001, pp. 273–321.
30. C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," in *Proc. of the ACM SIGMOD*, Mineapolis, MN, 1994, pp. 419–429.
31. S. Kenyon, "Signal Recognition System and Method," U.S. Patent 5,210,820, 1993.
32. T. Kastner, E. Allamanche, J. Herre, O. Hellmuth, M. Cremer, and H. Grossmann, "MPEG-7 Scalable Robust Audio Fingerprinting," in *Proc. AES 112th Int. Conv.*, Munich, Germany, May 2002.
33. A.L.-C. Wang and J. Smith II, "System and Methods for Recognizing Sound and Music Signals in High Noise and Distortion," U.S. Patent Application Publication US 2002/0083060 A1, 2002.
34. P. Cano, M. Kaltenbrunner, O. Mayor, and E. Batlle, "Statistical Significance in Song-Spotting in Audio," in *Proceedings of the International Symposium on Music Information Retrieval*, Oct. 2001.

Pedro Cano received a B.Sc and M. Sc. Degree in Electrical Engineering from the Universitat Politica de Catalunya in 1999. In 1997, he joined the Music Technology Group of the Universitat Pompeu Fabra where he is currently pursuing his Ph.D. on Content-based Audio Identification. He has been assistant professor in the Department of Technologies of the Universitat Pompeu Fabra since 1999. His research interests and recent work include: signal processing for music applications, within a real-time voice morphing system for karaoke applications, pattern matching and information retrieval, specifically content-based audio identification.
pedro.cano@iua.upf.es

Eloi Batlle received his M.S. degree in electronic engineering in 1995 from the Politechnical University of Catalunya in Barcelona, Spain. He then joined the Signal Processing Group at the same university where he was working on robust speech recognition. He received a PhD on this subject in 1999. While he was a PhD student he also worked as a researcher at the Telecom Italia Lab during 1997. In 2000 he joined the Audiovisual Institute (a part of the Pompeu Fabra University). Currently he is a member of the Music Technology Group of the same Institute where he leads several reseach projects on music identification and similarity. In 2000 he also joined the Department of Technologies of the Pompeu Fabra University and he teaches several subjects to undergraduate and graduate students. From 2001 he is the Deputy Director of this Department. His research interests include information theory, music similary, statistical signal processing and pattern recognition.
eloi@iua.upf.es

Ton Kalker was born in The Netherlands in 1956. He received his M.S. degree in mathematics in 1979 from the University of Leiden, The Netherlands. From 1979 until 1983, while he was a Ph.D. candidate, he worked as a Research Assistant at the University of Leiden. From 1983 until December 1985 he worked as a lecturer at the Computer Science Department of the Technical University of Delft. In January 1986 he received his Ph.D. degree in Mathematics. In December 1985 he joined the Philips Research Laboratories Eindhoven. Until January 1990 he worked in the field of Computer Aided Design. He specialized in (semi) automatic tools for system verification. Currently he is a member of the Processing and Architectures for Content MANagement group (PACMAN) of Philips Research, where he is working on security of multimedia content, with an emphasis on watermarking and fingerprinting for video and audio. In November 1999 he became a part-time professor in the Signal Processing Systems group of Jan Bergmans in the area of 'signal processing methods for data protection'. He is a Fellow of the IEEE for his contributions to practical applications of watermarking, in particular watermarking for DVD-Video copy protection. His

other research interests include wavelets, multirate signal processing, motion estimation, psycho physics, digital video compression and medical image processing.
ton.kalker@ieee.org

Jaap Haitsma was born in 1974 in Easterein, the Netherlands. He received his B.Sc. in Electronic Engineering from the Noordelijke Hogeschool Leeuwarden in 1997. He did his thesis in 1997 at the Philips Research Laboratories in Redhill, England, on the topic of: "Colour Management for Liquid Crystal Displays". Currently he is with the Philips Research Laboratories, Eindhoven, the Netherlands, where he has been doing research into digital watermarking and fingerprinting of audio and video since late 1997. From 1999 to 2002 he was also a part-time student at the Technical University of Eindhoven, where he obtained his M.Sc. in Electronic Engineering. His areas of interest include digital signal processing, database search algorithms and software engineering.
jaap.haitsma@philips.com