EURASIP Journal on Applied Signal Processing 2005:12, 1780–1793

Perceptual Audio Hashing Functions

Hamza Özer
Department of Electrical and Electronics Engineering, Boğaziçi University, 34342 Bebek, Istanbul, Turkey
National Research Institute of Electronics and Cryptology, Tubitak, 41470 Gebze, Kocaeli, Turkey
Email: hozer@uekae.tubitak.gov.tr

Bülent Sankur
Department of Electrical and Electronics Engineering, Boğaziçi University, 34342 Bebek, Istanbul, Turkey
Email: sankur@boun.edu.tr

Nasir Memon
Department of Computer and Information Science, Polytechnic University, Brooklyn, NY 11201, USA
Email: memon@poly.edu

Emin Anarım
Department of Electrical and Electronics Engineering, Boğaziçi University, 34342 Bebek, Istanbul, Turkey
Email: anarim@boun.edu.tr

Received 13 September 2004; Revised 16 February 2005; Recommended for Publication by Mark Kahrs

Perceptual hash functions provide a tool for fast and reliable identification of content. We present new audio hash functions based on summarization of the time-frequency spectral characteristics of an audio document. The proposed hash functions are based on the periodicity series of the fundamental frequency and on singular-value description of the cepstral frequencies. They are found, on one hand, to perform very satisfactorily in identification and verification tests, and on the other hand, to be very resilient to a large variety of attacks. Moreover, we address the issue of security of hashes and propose a keying technique, and thereby a key-dependent hash function.

Keywords and phrases: perceptual audio hashing, content identification, singular value decomposition, least-square periodicity estimation.

1. INTRODUCTION

In this study, we develop algorithms for summarizing a long audio signal into a concise signature sequence, which can then be used to identify the original record. We call this signature the perceptual hash function, because it is purported to reflect the perceptible component of the content. In other words, we aim to obtain audio hash functions that are insensitive to "reasonable" signal processing and editing operations, such as filtering, compression, sampling rate conversion and so forth, but that are otherwise sensitive to the change in content. Such perceptual hash functions can be used as a tool to search for a specific record in a database, to verify the content authenticity of the record, to monitor broadcasts, to automatically index multimedia libraries, to detect content tampering attacks, and so forth [1]. For example, in database searching and broadcast monitoring, instead of comparing the whole sample set, the hash sequence would suffice to identify the content in a rapid manner. In tamper proofing and data content authentication applications, the hash values of the applicant object are compared with hash values of the stored ones.

In the watermarking context, it is desirable to embed in a document a content-dependent signature, coupled with ownership or authorship label. Such content-dependent watermarks [2] are instrumental against copy attacks, where the attacker may attempt to fool the system by copying the embedded watermark from one document and transport it into another document. The hash values can also be used for the purpose of synchronization in watermarking [3], where multiple embedding is often used as a solution against desynchronization attacks. However, one may not want to embed the watermark into several parts of the stream. Instead, perceptual hash values can be used to select frames

pseudorandomly with a secret key, where the watermark will be embedded, and locate them later after modifications and attacks.

The two desiderata of the perceptual hash function are robustness and uniqueness. The uniqueness qualification implies that the hash sequence should reflect the content of the audio document in a unique way. Uniqueness is sometimes called randomness, which implies that any two distinct audio documents yield different and apparently random hash values. Consequently, the collision probability, the probability that two perceptually dissimilar inputs yield the same hash value, is minimized. The robustness qualification entails that the audio input can be subjected to certain nonmalicious manipulations, such as analog-to-digital (A/D) conversion, compression, sample jitter, moderate clipping and so forth, and yet it should remain, in principle, the same in face of these modifications. The line of demarcation between what constitutes a nonmalicious signal processing operation and what constitutes a change in content depends on the application.

There exists a number of perceptual audio hashing algorithms in the literature. Haitsma et al. proposed an audio hashing algorithm [4], where the hash extraction scheme is based on thresholding of the energy differences between frequency bands. They split the incoming audio into overlapping frames and, for each of the 33 logarithmically spaced frequency bands, they compute the energy. A 32-bit hash sequence is obtained for each time frame by comparing adjacent band energies. In another algorithm, Mihçak and Venkatesan [5] extract statistical parameters from randomly selected regions of the time-frequency representation of the signal. These parameters are discretized to form the hash values via an adaptive quantization scheme. The hash sequence is further rendered robust with an error correction decoder. The robustness of the algorithms against signal processing distortions and their employment for database searching are detailed in [4, 5]. In another vein, hash functions are used for database search purposes in [6, 7, 8, 9]. Burges et al. propose a distortion discriminant analysis technique to summarize the input audio signal [6]. They first compute the log spectrum by MCLT (modulated complex lapped transform) and summarize the spectral coefficient by PCA (principal component analysis) in a hierarchical manner. Kurth and Scherzer propose a database search technique by summarizing the audio signal through an up-down quantization and block coding method [7]. Sukittanon and Atlas use modulation frequency features as a summarization of audio signals and use them for database searching [8]. They characterize the time-varying behavior of the audio signal through modulation frequency analysis. After acoustic frequency detection by Fourier analysis, a wavelet transform is proposed for modulation frequency decomposition. Gruhne extracts a set of psychoacoustic features, such as the partial loudness in different frequency bands, spectral flatness measure, and spectral crest factor, from the spectrum of the audio signal and uses them as features in database searching [9]. Other studies are focused on audio signal for classification purposes, such as music, speech, silence and noise only frames [10, 11, 12]. Lu et al. use zero-crossing rate, short-time energy ratio, spectrum flux, LSP (line spectral pair) distance measure, band periodicity, and noise frame ration as features of the audio. Foote and Logan use mel-frequency cepstral coefficients as a feature set. In another study [13] Zhang and Kuo use energy, zero-crossing rates, harmonicity, and short-time spectra to determine that the incoming segment is speech, music, noise, applause, rain, cry, thunder, and so forth.

In this work, we investigate three perceptual audio hashing algorithms. Two of them operate in the time domain, and use the inherent periodicity of audio signals. In these schemes, the time profile of the dominant frequency of the audio track constitutes the discriminating information. The third one uses the time-frequency landscape, as given by the frame-by-frame MFCCs (mel-frequency cepstral coefficients), which is further summarized via singular value decomposition. The two periodicity-based schemes are original propositions and the third MFCC-based one is an improvement on the work of [12]. We demonstrate the merit of these hash functions in terms of correct identification probability and in terms of verification performance in a database search with corrupted documents.

The rest of the paper is organized as follows. In Section 2, periodicity-based hash techniques and methods for estimation of periodicity are presented. The audio hash method based on the singular value decomposition is given in Section 3. Experimental results are discussed in Section 4. Finally in Section 5, conclusions are drawn and feature studies are discussed.

2. PERIODICITY-BASED HASH FUNCTIONS

We conjecture that the periodicity profile of an audio frame can be used as a signature for identification and tamper control. The periodicity property of the audio signals has been used in such applications as voice activity detection [14], silence detection, and speech compression. We have considered two different periodicity-estimation methods, one based on a parametric estimation, while the other method is correlation based.

The block diagram of a generic periodicity-based hash extraction is depicted in Figure 1. The incoming audio object is processed frame by frame, and a single periodicity value is extracted for each frame. The audio signal is preprocessed in order to bring forward periodic behavior of the signal. Ideally, the goal of smoothing in the preprocessing stage [15] is spectral enhancement, that is, to remove spectral characteristics due to the audio content, while leaving spectral fine structure (fundamental frequency) intact. Inverse linear prediction (LP) filtering is a common way of performing this task. First a lowpass filter is applied followed by a 4-tap linear prediction inverse filter. This signal is denoted as $s_0(i)$, $i = 1, \ldots, N$, and used in Sections 2.1, 2.2, and Section 3. The audio signal is then segmented into overlapping frames, and each frame is windowed by a hamming window in order to reduce edge effects. The framing rate is 25 milliseconds and the overlap percentage is 50%

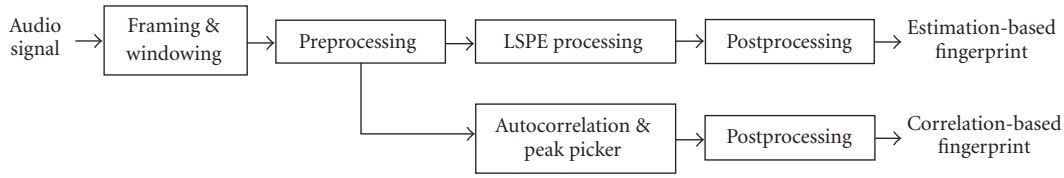


Figure 1: Block diagram of the hash extraction based on the two periodicity-estimation methods.

(i.e., the overlap length is 12.5 milliseconds), which are adequate to extract quasistationary segments from the audio signal. The period estimator operates on each such processed audio frame. Finally, the estimated time-series of frame-by-frame periods is postprocessed by a seven-tap finite impulse filter in order to mitigate the effects of distortions that could lead to a desynchronization effect. The term desynchronization refers to the fact that, as one searches for an audio document given a short clip (say, 5 seconds), the starting and terminating points of its hash will appear as randomly located on the whole hash sequence of the document. The smoothing mitigates this abrupt starting and stopping of the hash portion of the clip.

A few words are in order for the selected range of audio frequencies. It is known that the typical dominant frequency for human voice is between 50–400 Hz, whereas it can be much wider for music signals. However, even though the frequencies present in music can span a much wider range (about 50–4000 Hz), the range of 50–400 Hz still encompasses most of the musical sounds. For instance, Fitch and Shabana determined that the pitch period for guitar, saxophone, tanpura (an Indian instrument), and a male singing voice are 147 Hz, 154.7 Hz, 157.5 Hz, and 110.8 Hz, respectively [16]. Though, depending of the required accuracy and complexity constraints, some wider pitch range can always be accommodated, we will employ the 50–400 Hz range in our hash study. It is known that the audio signals have also nonperiodic intervals. Thus whenever a pitch algorithm returns a low-pitch confidence value, we will treat the frame as aperiodic and assign a score of zero for its periodicity.

2.1. Periodicity measure by least-squares estimation

Irwin investigated an optimum method for measuring the periodicity of audio signals by applying a least-squares periodicity-estimation (LSPE) technique [17]. In this scheme, the signal is conceived to be composed of a periodic and a nonperiodic component. The LSPE solves for the period P_0 that would maximize the energy of a periodic component with a given N-sample input signal $s_0(i)$, $i = 1,\ldots,N$. The details of the computation for each frame are in [18]. Let

$$s(i) = s_0(i) + n(i), \quad \text{for } i = 1,2,\ldots,N, \tag{1}$$

where $s_0(i)$ is a periodic component of input signal and $n(i)$ is the nonperiodic component. The periodic component possesses the property $s_0(i) = s_0(i + kP_0)$ for integer k and where P_0 is the period of $s_0(i)$. We now let $\hat{P}_0$ be our estimate and $\hat{s}_0(i;\hat{P}_0)$ the corresponding estimate of the periodic component. Omitting for simplicity the $\hat{P}_0$ dependence, the estimate $\hat{s}_0(i)$ is obtained from the input signal:

$$\hat{s}_0(i) = \sum_{h=0}^{K_0} \frac{s(i+h\hat{P}_0)}{K_0}, \quad 1 \le i \le \hat{P}_0,\ P_{\min} \le \hat{P}_0 \le P_{\max}, \tag{2}$$

where $P_{\min}$ and $P_{\max}$ are the lower and upper bounds of the period, and $K_0 = [(N-i)/(P_0)] + 1$ is the number of periods of $\hat{s}_0(i)$ fitting in the analysis frame. In (2), the variable i enumerates the folded signal samples within the range of hypothesized period $\hat{P}_0$.

The objective of the least-squares method is to find the period $\hat{P}_0$ that minimizes the mean square error $\sum_{i=1}^{N}[s(i) - \hat{s}_0(i)]^2$ over each analysis frame, which is shown to be equivalent to maximizing the $\sum_{i=1}^{N}\hat{s}_0^2(i)$ [18] component within the observed signal. Friedman [18] suggests that the estimated weighted energy of $\hat{s}_0(i)$ with normalization with respect to signal energy can be a periodicity measure as follows:

$$R_1(\hat{P}_0) = \frac{I_0(\hat{P}_0) - I_1(\hat{P}_0)}{\sum_{i=1}^{N} s^2(i) - I_1(\hat{P}_0)}, \tag{3}$$

which, when maximized, yields an unbiased estimate of the periodicity. In this expression the functional $I_0(\hat{P}_0)$ represents the estimated weighted energy of $\hat{s}_0(i)$, and $I_1(\hat{P}_0)$ is the energy contribution of the diagonal terms of the weighted energy sums. These functionals are defined as follows:

$$\begin{aligned} I_0(\hat{P}_0) &= \sum_{i=1}^{\hat{P}_0} \frac{\left[\sum_{h=0}^{K_0} s(i+h\hat{P}_0)\right]^2}{K_0}, \\ I_1(\hat{P}_0) &= \sum_{i=1}^{\hat{P}_0}\sum_{h=0}^{K_0} \frac{s^2(i+h\hat{P}_0)}{K_0}. \end{aligned} \tag{4}$$

Notice that the energy contribution of the diagonal terms is subtracted from the total signal energy before normalization. For each frame, $R_1(\hat{P}_0)$ as in (3) is computed for values of $\hat{P}_0$ between $P_{\min}$ and $P_{\max}$, and $\hat{P}_0$ that maximizes the value of $R_1(\hat{P}_0)$ is determined as the estimated period of the processed frame. The $R_1(\hat{P}_0)$ takes values in the interval [0,1] and acts as a confidence score for a frame to be periodic or not.

We thresholded this confidence score at the value of 0.5, such that any frame that reports a value of $R_1(\hat{P}_0)$ less than

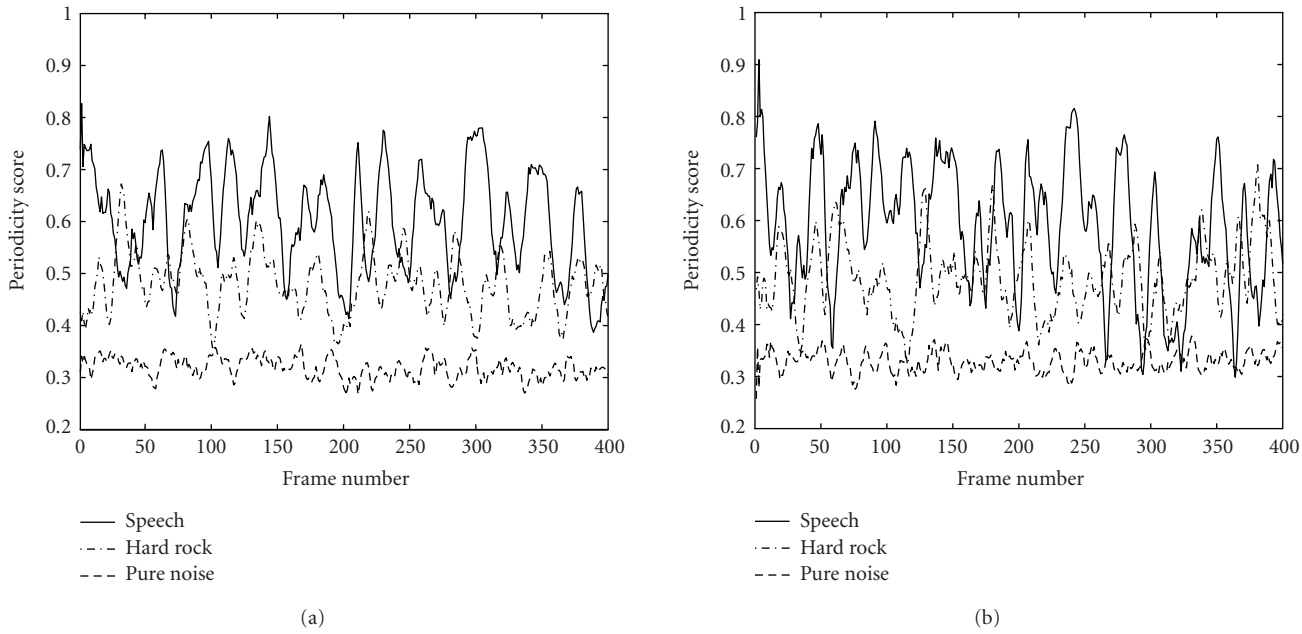


FIGURE 2: (a) CPE results: periodicity profiles of hard rock music, speech, and pure noise. (b) LSPE results: periodicity profiles of hard rock music, speech, and pure noise.

0.5 is labeled as aperiodic. The experiments that we have conducted show that the periodicity score, $R_1(\widehat{P}_0)$, obtained from white noise is generally below the threshold 0.5, as in Figure 2b. Values of the threshold above and below the threshold of 0.5 did not improve the performance. Note that Tucker [14] also found the same empirical result.

2.2. Periodicity measure by a correlation-based analysis

The first peak of the autocorrelation of the linear prediction residue indicates the pitch period and is commonly used as a pitch estimator. This correlation-based periodicity estimate, called CPE, has the following expression:

$$\widehat{P}_0 = \begin{cases} \arg\max R(k), & \text{for } k \neq 0 \quad \text{if } R(\widehat{P}_0) \geq 0.5, \\ 0 & \text{if } R(\widehat{P}_0) < 0.5 \end{cases} \tag{5}$$

$$R(k) = \frac{(1/(N-k)) \sum_{i=0}^{N-k} s(i)s(i+k)}{(1/N) \sum_{i=0}^{N} s^2(i)}.$$

The efficacy of the CPE method is enhanced by a four-tap prediction and decimation process. The advantage of the correlation-based method is that it requires about three times less computation as compared to the parametric estimation method in Section 2.1. We decided that the audio frame is pitchless, without an explicit periodicity, as in the case of unvoiced speech or silence, whenever the first correlation peak in $R(k)$ of (5) falls below 0.5.

One can question whether the periodicity profile is available from any audio track, for example, a hard rock track. Figure 2 illustrates the periodicity time sequence of a popular hard rock song ("Be quick or be dead" from Iron Maiden). It can be observed that, albeit lower as compared to speech, even hard rock music results in some estimated periodicity profile, which is definitely much higher than the noise case. Other hard rock songs gave similar results.

3. A TRANSFORM-DOMAIN-BASED HASH FUNCTION

In this section, we focus on transform-domain hash functions in contrast to the previous section, where we essentially worked on the time domain. More specifically, the audio signal is divided into possibly overlapping frames and each frame is represented by its mel-frequency cepstral coefficients (MFCCs), which are short-term spectral-based features [15]. A singular value decomposition (SVD) further summarizes these features. Note that in the SVD-based method we use the original signal, and not its lowpass filtered version, as in the periodicity-based schemes.

The block diagram of the computational procedure for MFCC features is given in Figure 3. One computes the discrete Fourier transform (DFT) of each windowed frame, and the log magnitudes of these coefficients are retained. This spectrum is then partitioned into mel-spaced frequency bins in accordance with the human auditory system's nonlinear perception, which is linear below 1 kHz and logarithmic above [15]. The mel-spectral components are averaged to obtain a smooth spectrum through mel-filtering. Mel-filters have nonlinear and overlapped mel barks [15]. Finally, MFCC features are obtained by applying a discrete cosine transform (DCT) on the mel-spectral vectors. More specifically, one starts by computing the N points (DFT) of the

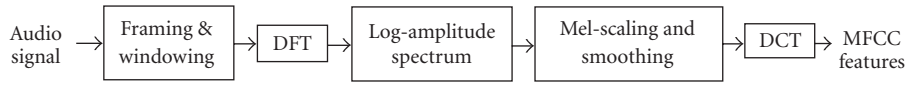


Figure 3: Block diagram of the hash extraction based on the MFCC method.

input signal

$$S(k) = \sum_{i=0}^{N-1} s(i)e^{-j2\pi ik/N}, \quad 0 \le k \le N-1. \tag{6}$$

One also defines a filterbank of M filters, where the triangular filters, $H_m(k)$; $m = 1, 2, \ldots, M$; $k = 0, 1, \ldots, N-1$, have increasing bandwidth according to the mel-scale. These filters are used to compute the average spectrum around each center frequency in (6). The log-energy spectrum at the output of each filter is computed as

$$\Psi(k) = \ln\left[\sum_{k=0}^{N-1} |S(k)|^2 H_m(k)\right], \quad 1 \le m \le M. \tag{7}$$

Finally the mel-frequency cepstrum is computed as the DCT of the M filter outputs of (7):

$$c(n) = \sum_{m=0}^{M-1} \Psi(m) \cos\left(\frac{\pi n(m-0.5)}{M}\right), \quad 1 \le n \le M. \tag{8}$$

The number of MFCC terms in (8) is typically between 24 and 40, although for speech the first 13 cepstrum coefficients are often used.

This results in an $F \times M$ matrix, where each row consists of the M MFCC values for a frame, and there are F rows, the number of frames into which the whole audio signal has been segmented. This matrix expresses the evolution of the signal in the time-frequency landscape. A concise summary of this landscape is computed by the SVD of the calculated MFCC matrix. The singular value decomposition effectively reduces the $F \times M$-dimensional MFCC-feature matrix into a much smaller invertible square matrix. Thus, the given $F \times M$ matrix is decomposed as $A = UDV^T$, where A is the $F \times M$ matrix that we want to summarize, D is an $F \times M$ matrix with only $\min(F, M)$ diagonal elements, U is an $F \times F$ orthogonal matrix, and V is an $M \times M$ orthogonal matrix. In general, a few singular values (first few components of the diagonal matrix D) give a good summarization of the matrix A [19]. In our study we employed the first one to three singular values only.

4. EXPERIMENTAL RESULTS

We have performed simulation experiments in order to test (i) the robustness of the perceptual hash for identification, where the critical behavior is the statistical spread of the hash function when an audio document is subjected to various signal processing attacks; (ii) the uniqueness of the perceptual hash, where the important behavior is the fact that the hashes differ significantly between two different contents. In other words, in the first case, we want to identify a document (the genuine version) and its variants under signal processing attacks. In the second case, we want to classify documents with different contents, so that if we want to verify a document, the others in the database appear as "impostors." In a decision-theoretic sense, the uniqueness property is related to the probability of false alarm or false alarm rate (FAR), while the robustness property is linked to the probability of misses or false rejection rate (FRR).

In our database we used 900 3–4 second-long utterances, which were distinct sentences in Turkish and recorded from the same speaker. For uniqueness tests, recordings from the same speaker represent the worst case, since there are only differences in content, but no interspeaker differences. We know at least that the pitch levels from the same speaker will be closer than the pitch levels from different speakers. The utterances were recorded in an acoustically shielded room and digitized at a 16 kHz sampling rate. In addition we conducted some experiments with music data, that is, 650 music pieces overall, where the fragments had durations of 6 seconds. These fragments were extracted from songs of popular artists, such as Celine Dion, Luis Miguel, Mariah Carey, Rolling Stones, and U2. Each fragment was treated as a separate object to be recognized.

4.1. Parameters used in the experiments

The settings of the feature parameters were as follows. For the LSPE periodicity estimator, $P_{\min}$ and $P_{\max}$ were set, respectively, to 40 and 320 samples, which means that the admissible periods were between 50 Hz to 400 Hz for a 16 Hz sampled signal. The frames, taken to be 25-millisecond long, were overlapped by 50 percent. Frames were preprocessed by first lowpass filtering them with a cutoff frequency of 900 Hz and then through a 4-tap linear prediction filter [15]. For the correlation-based periodicity method, the signal was decimated by a factor of four before the correlation analysis was performed. The resulting hash consisted of a sequence of 79 samples/s, which represents a compression of the signal by a factor of approximately 200.

For the SVD-based method, we considered 13 features, so that the MFCC data formed an $F \times 13$ feature matrix. We experimented with up to three singular values, and it was observed that even a single singular value was often adequate. This is again the basic tradeoff between uniqueness, which improves by including more singular values, and robustness, which conversely improves with a smaller number of eigenvalues. The hash size depends upon the number of frames and the number of singular values chosen, which, for the choice of 1 to 3 singular values, becomes 26, 52, and 78 samples per second, respectively. In our study we employed three

singular values in order to make the hash size (which is 78 samples per second in that case) compatible with the other two methods.

4.2. The simulated attacks

We programmed eleven types of attacks (some attacks also applied to different degrees) to evaluate the performance of the proposed hash functions. The hash sequence of the original record ($X(f)$, $f = 1, 2, \ldots, N$) is compared with the hash value of the attacked version ($Y(f)$, $f = 1, 2, \ldots, N$). We used normalized correlation coefficient as the similarity measure between the hash sequence of the original sound file and that of the test file, that is, the modified file. This similarity measure is defined as

$$r = \left| \frac{N \sum_f X(f)Y(f) - \sum_f X(f) \sum_f Y(f)}{\sqrt{\left[N \sum_f X^2(f) - \left(\sum_f X(f)\right)^2\right]\left[N \sum_f Y^2(f) - \left(\sum_f Y(f)\right)^2\right]}} \right| \tag{9}$$

and takes values in the range (0,1), since the terms of the hash sequence are always positive. We have also attempted to use L_2 distance as a similarity measure and compared the results with correlation measures. The L_2 distance that we have used is given by

$$d = \frac{1}{N} \sqrt{\sum_f^N (X(f) - Y(f))^2}. \tag{10}$$

The attacks consist of upsampling by a factor 44.1/16 (final rate 44.1 kHz), downsampling by a factor two (final rate 8 kHz), adding white Gaussian noise resulting in 20, 25, 30, 35 dB signal-to-noise ratios, denoising operations with and without noise addition, pitch downconversion and upconversion by 1% and 2%, time compression by 2%, 4%, and 6%, random cropping by 8% and 10% of total length, telephone filtering, and finally 3:1 amplitude compression below 10 dB and above 20 dB. Some of these attacks were slightly audible, as in the cases of 20 and 25 dB additive noise, 2% pitch conversions, 6% time compression, and 10% random cropping. We have forced the attacks beyond their perception thresholds in order to gauge them, that is, to scale the attacks up to their ultimate acceptable level to simulate the worst cases in database search. By using several runs of the attacks, the receiver operating curves (ROC) are calculated, where the probability of correctly identifying an audio record is plotted against the probability of falsely accepting another audio track as the genuine version. The list of all attacks is shown in Table 1.

The effects of the sample attacks are presented in Figure 4, where we show the original audio clip and the attacked versions of the clip (see Figures 4b and 4c) that have inaudible or slightly audible modifications.

Table 1: The attacks and levels used in the experiments.

Type of attack	Attack level
Subsampling	16 kHz to 8 kHz
Upsampling	16 kHz to 44.1 kHz
Noise addition (20, 25, 30, 35 dB SNR)	Additive white Gaussian noise
Denoise filtering after noise addition	Wavelet-based denoising
Denoise filtering of clear signal	Wavelet-based denoising
Raise pitch	1% and 2%
Lower pitch	1% and 2%
3:1 amplitude compression below 10 dB	With CoolEdit prog.
3:1 amplitude compression above 20 dB	With CoolEdit prog.
Time compression	2%, 4%, and 6%
Random cropping	Total amount of 8% and 10%
Telephone filtering	135–3700 Hz
MP3 compression	32 Kbps

4.3. Robustness and uniqueness performance

We calculate the interrecord distances and the intrarecord distances. The interrecord distances are the (dis-)similarity scores between altered (attacked) versions of a record and altered versions of all other records. To this effect, for each of the L records in the database we calculate the (dis-)similarity to the remaining $L - 1$. Since there are 900 speech and 600 music records, we calculate a total amount of $L(L-1)/2$ or 619, 970 distance values. The intrarecord distances are the (dis-)similarity scores between the attacked versions of the same audio segment. For this purpose we have randomly selected 200 music records and 200 speech records and applied upon them twenty varieties of attacks, some with more than one parameter setting as in Table 1. Thus we collected $20 \times 400 = 8000$ intradistance figures.

Robustness characteristics

Robustness of a perceptual hash scheme implies that the hash function is not affected by signal manipulations and editing operations, which do not change the perceived content.

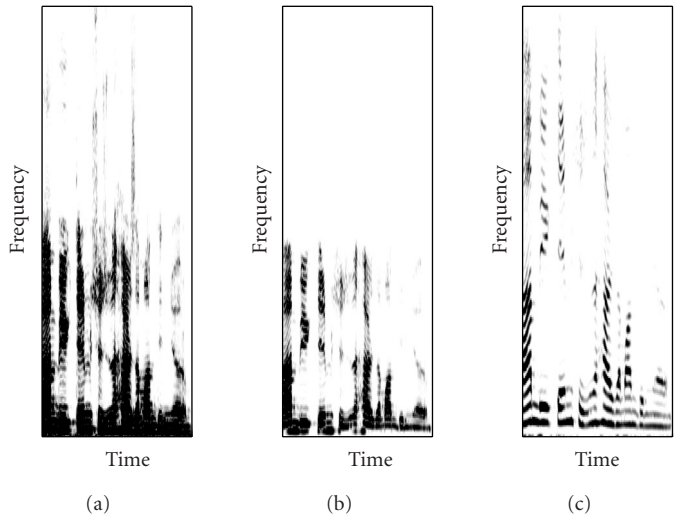


Figure 4: (a) Original spectrogram of the record. (b) Spectrogram after telephone filtering attack. (c) Spectrogram after attack with factor-two downsampling.

The hash lengths are 79, 79, and 78 samples/s, respectively, for LSPE, CPE, and SVD-MFCC techniques. Notice that we could have made the SVD-MFCC rate smaller, that is, 26, without compromising any of its robustness performance. However, experiments have shown that uniqueness suffers if we consider less than three eigenmodes.

In Figures 5 and 6, we present the histograms of the similarity (correlation coefficient) scores for speech and music records. The dispersion of the histograms on the right are indicative of the degree to which the hash value is affected by the signal processing attacks, hence its robustness. Histograms on the left indicate the randomness of the hash, hence uniqueness, as explained at the end of this section. In Figure 7, the results with L_2 distance as the similarity measure are also presented. For the L_2 distance, the spread of the left histograms shows the degree to which the hash value is affected by the signal processing attacks since ideally their L_2 distance should be zero. Comparison of the distance histograms and similarity of performance scores has indicated to us that the specific distance metric used does not have much effect.

In addition, we tested the 32 kbps MP3 compression attack using the commercial CoolEdit compressor program. The experiments were carried out with 200 speech and 200 music excerpts where we compared hash values of the original unmodified audio files with those of MP3 compressed-decompressed files. The minimum and average similarity measures (normalized correlation scores) for CPE, LSPE, and SVD-MFCC methods are shown in Table 2. Thus even at such low compression rates of MP3, the proposed hashing scheme is adequately robust.

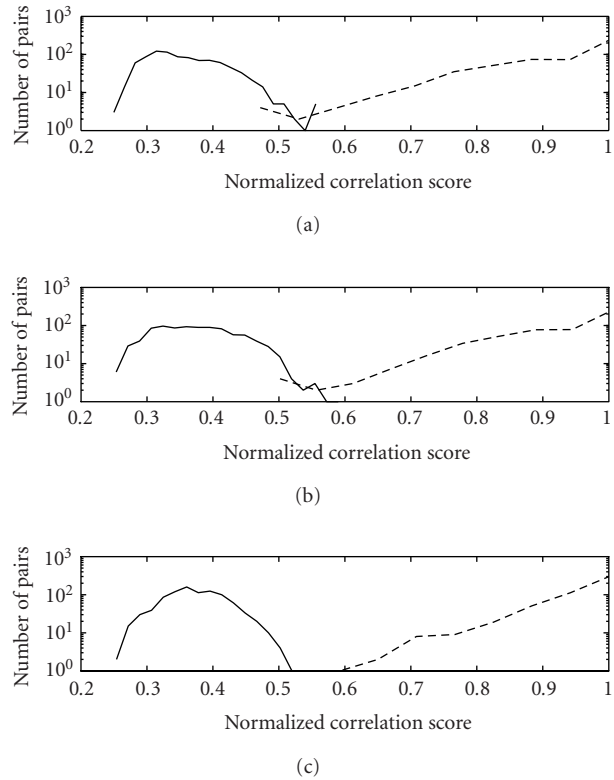


Figure 5: Histograms of the difference of the hash functions extracted from speech data and using the correlation measure: different objects (solid lines) and distorted versions of the same object (dashed lines). (a) LSPE, (b) CPE, and (c) SVD-MFCC.

Uniqueness characteristics

We tested whether hash sequences could be confounded in a large repertoire of audio files. Thus, for each of the 900 utterances and 650 music records, the hash value was computed and compared with all the other ones. The utterances were 3–4 second-long distinct sentences, uttered by the

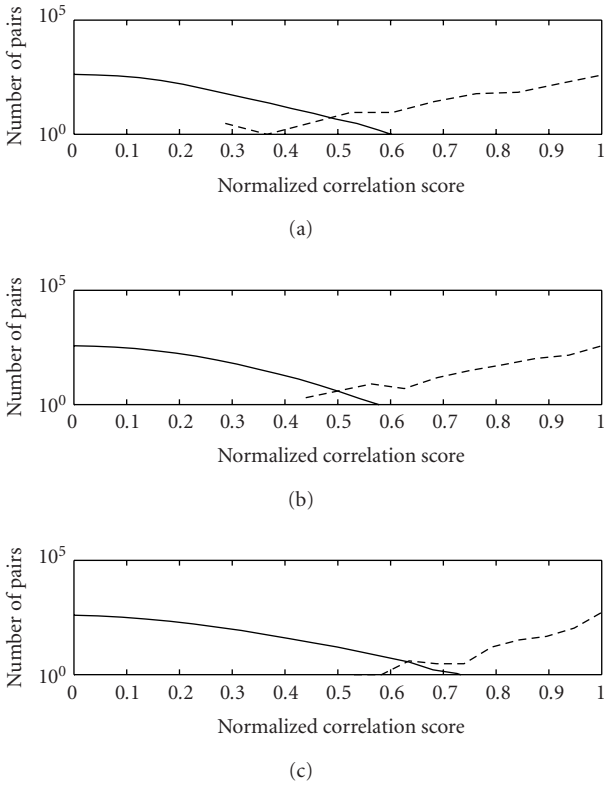


Figure 6: Histograms of the difference of the hash functions extracted from music data and using the correlation measure: different objects (solid lines) and distorted versions of the same object (dashed lines). (a) LSPE, (b) CPE, and (c) SVD-MFCC.

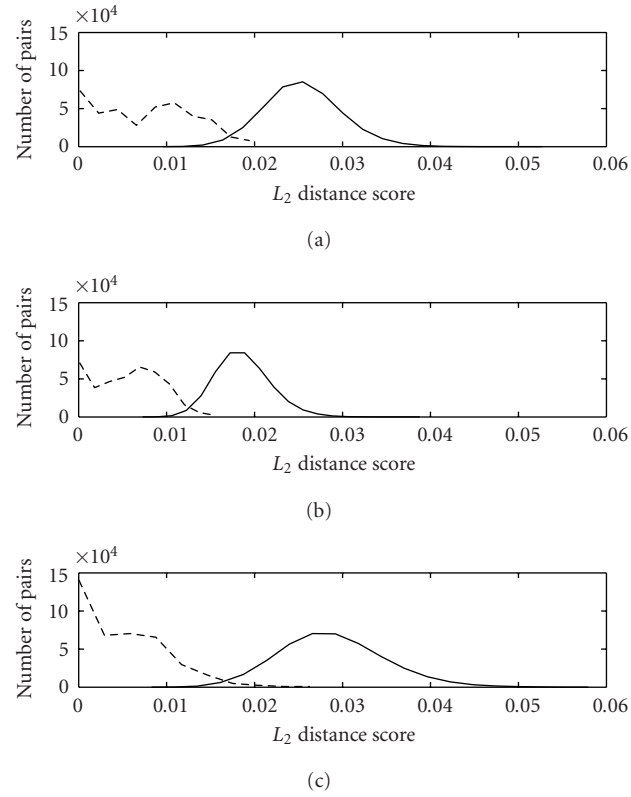


Figure 7: Histograms of the difference of the hash functions extracted from speech data and using L_2 distance measure: different objects (solid lines) and distorted versions of the same object (dashed lines). (a) LSPE, (b) CPE, and (c) SVD-MFCC.

same speaker. Notice that the use of only one speaker represents the worst case for confounding, as we forego inter-speaker variability. The music records are chosen from different types of music as explained above. Ideally, the similarity score between hashes should be zero for the correlation measure and as large as possible for the L_2 distance. The results are presented in Figures 5 and 6, for speech and music with correlation measure, and in Figure 7 for the L_2 distance.

It can be observed from Figures 5, 6, and 7 that the LSPE and CPE have very similar score distributions, with LSPE slightly more compact under attacks. SVD-MFCC seems to hold faster under attacks, as its robustness performance is better than the others. SVD-MFCC is similarly somewhat superior to the periodicity-based hash methods in that the impostor distribution overlaps less with the genuine distribution. Furthermore, there was not a significant difference between speech and music documents or a major difference between normalized correlation and L_1 (not plotted) or L_2 distances.

4.4. Identification and verification tests

The ultimate proof of the robustness and uniqueness properties of the proposed hash functions will show in their identification and verification performances. The identification problem is to recognize an audio record in a database of other audio records. For example, a short record from within a song can be given, and the algorithm has to identify the song within a large database of songs through this partial evidence. The identification or detection performance can thus be measured in terms of the percentage of correct recalls from a database. The verification problem, on the other hand, occurs when we want to prove and disprove that an audio record is indeed what it is claimed to be. In a verification experiment, one must test both the "genuine record" as well as all the other "impostor records" in their various altered versions, transfigured by the attacks described above. The verification performance is best given by the receiver operating characteristic (ROC) curves. In ROC we plot correct detection (or alternately, the probability of FRR) versus FAR. We have a false alarm situation when an impostor record

Table 2: Minimum and average correlation scores with the three hashing methods.

Hashing method	Minimum score	Average score
CPE	0.848	0.920
LSPE	0.828	0.911
SVD-MFCC	0.982	0.993

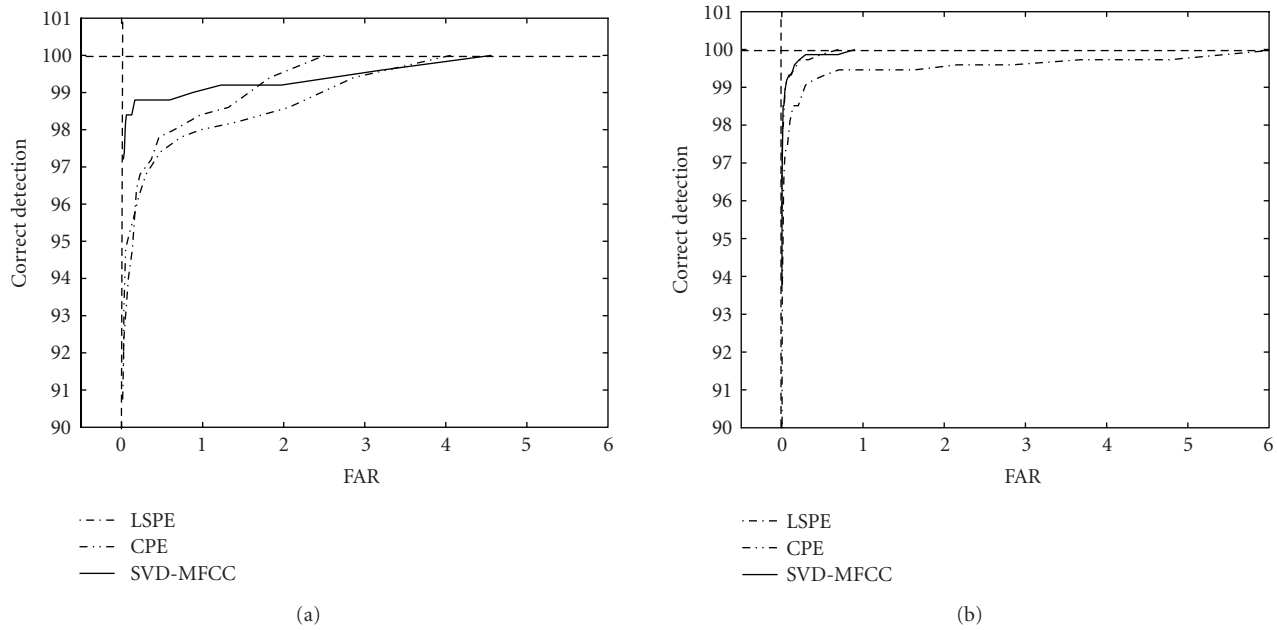


Figure 8: ROC plots of the three methods, where FARs are given in percentages, and where hash function similarity is measured with correlation coefficient: (a) speech data set and (b) music data set.

(that is, any other content) is identified in lieu of the genuine record; in contrast, we have a correct detection whenever the claimed identity of the genuine record is detected correctly, that is, we hit the correct content. Finally, we have a false rejection, whenever the claimed identity of the genuine record is rejected by the test.

The correlation-based FAR and correct detection performance for both speech and music are given in Figure 8, while Figure 9 shows ROC curves based on the L_2 distance. These experiments reveal that, in general, the hash function derived from SVD-MFCC has better performance, especially in the low range of FARs. On the other hand, LSPE has slightly better performance than either CPE or SVD-MFCC but only at higher FAR scores.

For identification purposes, we choose random parts of the records to be identified as test data (a token), and search for the object in the database where the most similar hash occurs. For speech, the tokens are chosen as 1.5-second clips within the records of 3–5 seconds, and for music, the token is chosen as a 3-second clip within records of 6 seconds. We position the test segments randomly within the original records in order to simulate misalignments. The correct detection rates are summarized in Tables 3a and 3b, respectively, for the original objects (unattacked) and for their attacked versions. The performance with attacked records is the average of the scores of all the attacks described in Section 4.2. These results indicate that all three perceptual hashing techniques perform on a par, with SVD-MFCC marginally superior. Generally LSPE performs slightly better than CPE except when applied to a database consisting only of music. SVD-MFCC performs better than the other two methods, though for music only, CPE and SVD-MFCC are alike.

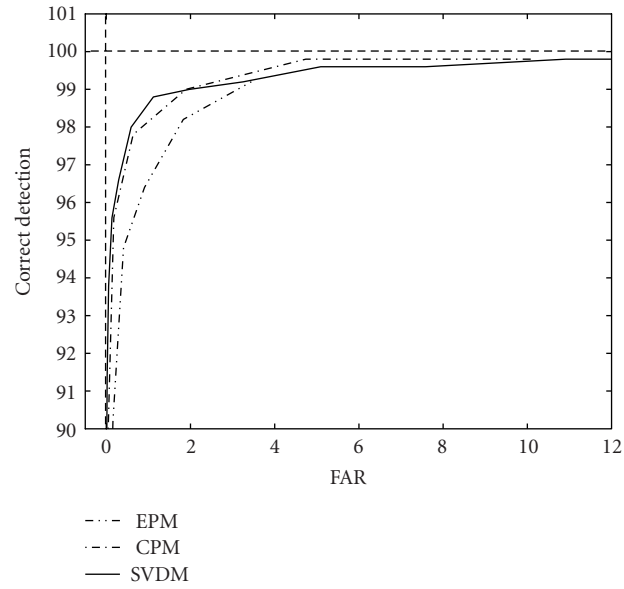


Figure 9: ROC plots of the three methods, where FARs are given in percentages, and where hash function dissimilarity is measured with L_2 metric for the speech data set.

In a separate experiment, we tested the effect of the token length on identification. For relatively small databases, as the token length increases the probability of correct detection saturates quickly toward a well-defined asymptote. Hence we increased the database size to a more challenging figure of 2302 6-second excerpts from popular music, and varied the token size between one and five seconds in steps of one second. The results, as tabulated in Table 3c, show that token

Table 3: (a) Identification performance of the original speech and music documents for different hash functions. (b) Identification performance of the attacked speech and music documents for different hash functions. (c) Identification performance of the 2302 music documents with different search sample sizes.

(a)

Database size (original documents)	LSPE performance	CPE performance	SVD-MFCC performance
200 (mixed)	100%	99.5%	100%
650 (music)	100%	99.84%	99.84%
900 (speech)	98.15%	98%	100%
1550 (mixed)	96%	95.6%	96.7%

(b)

Database size (attacked documents)	LSPE performance	CPE performance	SVD-MFCC performance
200 (mixed)	99%	98.9%	99.2%
650 (music)	99.4%	99.8%	99.78%
900 (speech)	96.1%	94.5%	98.1%
1550 (mixed)	89.1%	88.3%	90.2%

(c)

Search sample size	LSPE performance	CPE performance	SVD-MFCC performance
1 s	66.5%	75.1%	76%
2 s	82.6%	88.4%	95%
3 s	95.5%	96.5%	99%
4 s	98.2%	99.8%	99.9%
5 s	100%	100%	100%

Table 4: Verification performance of the attacked speech and music documents for different hash functions.

Methods	900 speech	650 music	1550 mixed
	FAR = FRR performance		
LSPE performance	99.08%	99.32%	97.1%
CPE performance	99.05%	99.73%	96.9%
SVD-MFCC performance	99.13%	99.73%	97.2%
	FAR = 1% performance		
LSPE performance	98.48%	99.46%	97.8%
CPE performance	98%	100%	97.7%
SVD-MFCC performance	99.18%	100%	98.1%

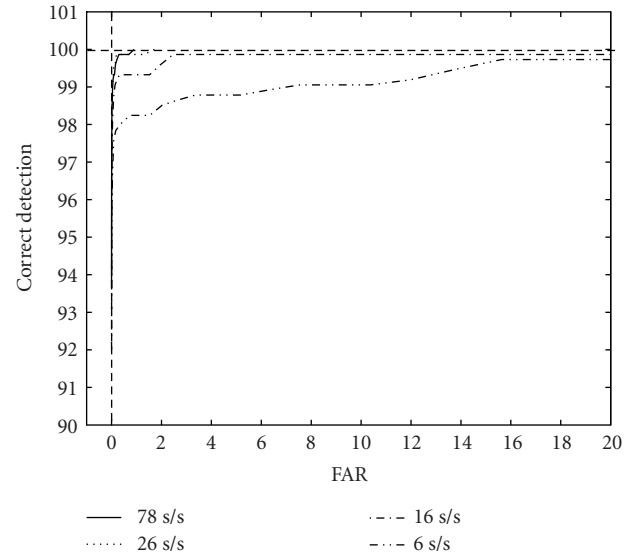


Figure 10: Receiver operating characteristics for different hash sizes in samples/s (s/s). 78 s/s: 3 SVDs, 25-millisecond frame length; 26 s/s: 1 SVD, 25-millisecond frame length; 16 s/s: 1 SVD, 40-millisecond; 6 s/s: 1 SVD, 100-millisecond frame length.

sizes equal to or longer than three seconds yield adequate performance. The SVD-MFCC method performs better than the two periodicity-based methods at all token sizes.

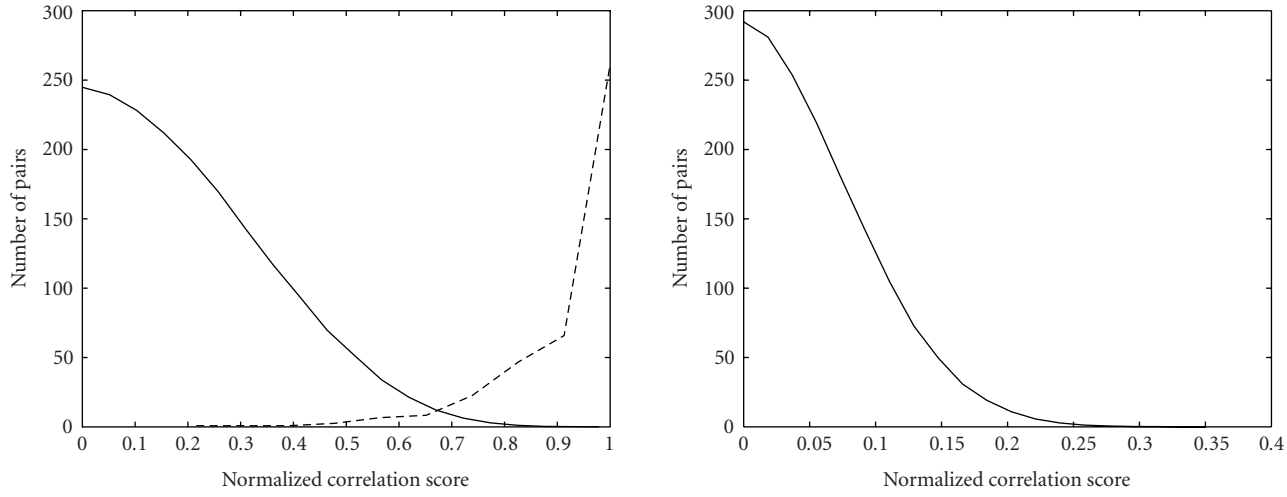
The conduct of the verification experiments can be deduced from the ROC curves. In these experiments, if the maximum similarity between a test hash and any other hash in the database (other than the test data in its original or altered forms) exceeds a predetermined threshold, then the test data is marked as a probable false detection. Conversely, one can present an "impostor" document, and see whether it ever matches our target document, that is, if their similarity score remains above a threshold. We gleaned from the ROC curves the results for both the equal error rate case (FRR equal to FAR), and for the FRR = 1% case. Table 4 summarizes the outcome of the verification experiments. The experiments show that in general SVD-MFCC performs better than the other two hash techniques. However, for a data set consisting only of music, the CPE performance was similar to that of SVD-MFCC.

4.5. Effect of the length of the hash function

We explored the effectiveness of the hash function as a function of its length. Thus we systematically reduced the hash size from 80 samples/s to 6 samples/s, by reducing the number of singular values considered and/or by varying the frame size. The receiver operating characteristics pictured in Figure 10 show that the system is quite insensitive to the size of the hash, and that its size can be reduced by more than an order of magnitude. For example, at 1% false acceptance rate, the probability of false rejection still remains under 2%.

4.6. Security aspects of the audio hash functions

The security of the hash extraction becomes important in audio authentication schemes. One common way to provide hash security is to devise a key-based scheme such that for two different keys, K_1 and K_2, the resulting hash functions become totally independent. Thus we minimize the probability of collision, that is, we want to guarantee that two distinct inputs yield different hash functions and that the hash sequences are mutually independent.



(a) (b)

Figure 11: Histograms of the difference of the hash functions with 900 speech records: (a) hashes of the different objects (solid line) and those of the attacked versions of the same object (dashed line); (b) hashes obtained from the same object with different keys.

Notice that secure fingerprinting requires that the pirate should not be capable of extracting the hash value of the content without knowledge of some secret key. This would, for example, allow him to change the content while preserving the hash, that is, find a collision which would circumvent any hash-based authentication mechanism being used. As another example, it could also enable him to manipulate the bits while preserving the content and yet change the hash. This would be done, for example, when a pirate may want to avoid being detected by a copyright controller for unauthorized use of some content.

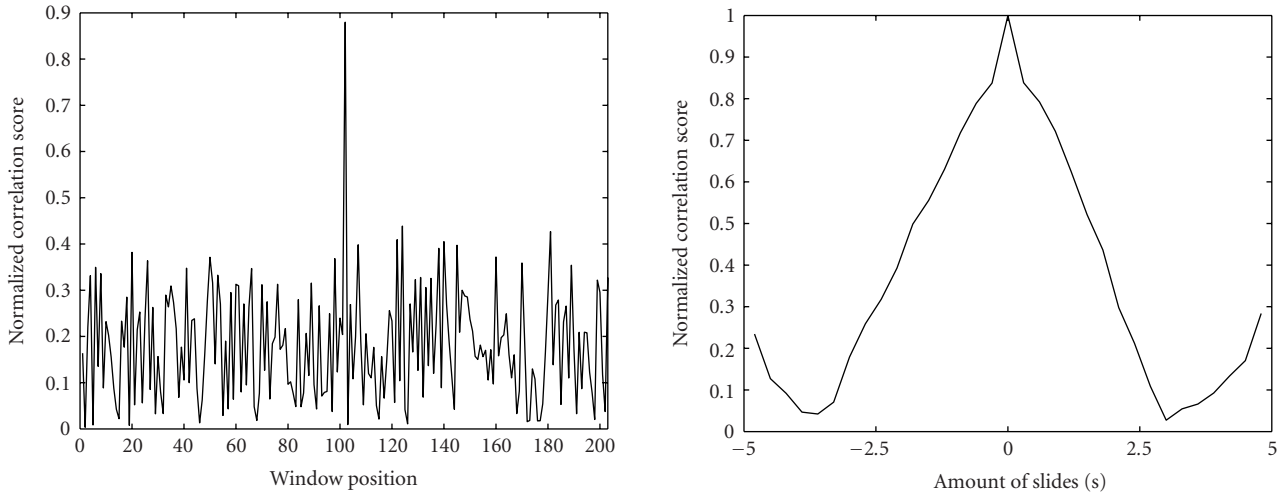
One way to arrive at a key-based hash function is to project the resulting hash sequences onto key-dependent random bases. Another scheme would be to subject the analog hash sequence to random quantization [20]. In this scheme, the hash sequence is quantized using a randomized quantizer, and the quantizer itself becomes the source of randomness in the hash function's output. A third scheme could be based on random permutation of the observation frames with possible overlaps. Thus we generate a key-based sequence of visiting positions and translate in saccades the frame window according to this sequence (recall that we used 25-millisecond windows with 50% overlap).

We have implemented such a key-instrumented hashing method with the LSPE-based technique. Robustness and uniqueness test results with keyed hash are shown in Figure 11a. We have generated 1000 hash values from an audio clip using different permutation matrices, and as before, the similarity of all possible pairs of the hash values (thus $1000 * 999/2 = 499500$ pairs) are calculated. The histogram is presented in Figure 11b. Similarity closer to zero indicates the amount of independence of keyed hashes. It can be deduced from the figure that the similarities between the hashes of the same object with different keys are as small as the similarity of distinct objects. Thus the hash values are significantly dependent on the key information.

4.7. Broadcast monitoring

One prominent use of audio and video hash functions is in the field of broadcast monitoring, that is, either to collect statistics about the instant and the number of times a specific content has been broadcasted, or alternately, to verify that a claimed broadcast content matches the reference content.

This can be done on-line in real time, as the hashing algorithms have number of operations that are an order-of-magnitude simpler than sophisticated audio compression algorithms. Suppose a T-second segment is being sought. As an audio window ($W > T$, possibly $W \gg T$) is captured and stored, it will be hashed in steps of ΔT interval, overall resulting in $W/\Delta T$ hash calculations and comparisons. To give an example, we have observed that scanning steps of 1.25 seconds suffice for records of 5 seconds (see Figure 12). Note that this search with a continuously sliding window over a continuous stream is reminiscent of the fade-over attack in video [21]. In the audio slide-over, one audio record gradually passes or becomes eclipsed by another one. Figure 12a shows the trace of hash correlation scores in the course of time. In this example, a 5-second excerpt is being sought within a 258.4-second-long record of audio ("Endless Love" by Mariah Carey). One can notice that the only peak above the threshold occurs at the exact point where the two records match. In Figure 12b, we expanded the peaking region and showed the sensitivity of the hash to the slide-over effect. As the window slides past the exact matching instance of the record and starts incorporating audio material from the future (past), the correlation of the hashes decreases gracefully. We have experimented with several window sizes T, from 1 second to 5 seconds to determine the width of the main lobe for a given threshold. This is defined as the amount by which the window can slide to cover new audio material while discarding a proportional amount of its past. We have observed that the main lobe is equal to $T/4$: thus for example,



(a) (b)

Figure 12: (a) Broadcast audio monitoring: trace of correlation scores when a 5-second audio excerpt is being continuously sought in a 258.4-second record. The scan step is $\Delta T = 1.25$ seconds. (b) Sensitivity of the hash to the slide-over effect against the scan step size when the length of the excerpt is 5 seconds.

when searching a record of 5 seconds, one can partially cover it, say, the first or the last $0.75T$ seconds and yet be able to identify it. In other words, hash remains invariant, provided the amount of overlap between the search window and the target record is at least 75%. The main lobe is important in determining the scanning step size for hashing in the broadcast monitoring application.

4.8. Intraspeaker and interspeaker performance

It is intriguing to investigate whether the hash scheme can survive variability due to speaker variability, whether the interspeaker or the intraspeaker type. The speaker variability emerges in the pitch, formant, and prosodic features, and overall they may cause sufficient changes to make a speech object look like another object. A caveat is that the hash function is not intended to be a tool for speaker or speech recognition, let alone a speaker-independent recognition algorithm. Since speech record identification would perform worse when different speakers were involved as compared to a single speaker case, we concentrated our attention on intraspeaker variability. For this purpose, we recorded 10 3-second sentences, each uttered 10 times, from 5 speakers, possibly with varying prosody. The hash correlation scores were only computed in the intramode, that is, between utterances of the same speaker. Figure 13a shows the hash correlation scores for three different speakers, where each plot displays the correlations of utterances by a single speaker (one utterance versus the other 9 utterances of the same speaker). One can see that the correlation scores (intraspeaker scores) vary significantly, and in fact can sometimes reach quite low values below the threshold. Another proof of the fact that content hunting (identification/verification) will not work very well even for different utterances of the same speaker is given in the hash distance histograms in Figure 13a. These histograms were obtained by calculating the 450 distances between 10 utterances of 10 sentences from 5 speakers. Superposed on these histograms is the plot of correlation without speaker variability taken from Figure 5. At the same threshold value of 0.5, the recognition rate drops down to 0.77 from 0.98 to 1.0. In summary, the perceptual hashing scheme does not operate well under the speaker variability "attack."

5. CONCLUSION

We have constructed three novel perceptual audio hash functions to enable content-oriented search in a database and/or as an instrument for security protection. We studied the verification and identification performance of the hash functions in a database composed of speech and music records. An important conclusion was that all three hash functions (LSPE, CPE, and SVD-MFCC), and in particular, the SVD-MFCC variety, perform satisfactorily in the identification and verification tasks. In fact, their performance resists a large variety of attacks, most of which have been pushed to their perceptually noticeable thresholds. A second conclusion is that these methods collapse the input audio file into a fingerprint stream of a much smaller size, typically from 16 kHz sampling rate to 26 samples per second, which represents reduction by a factor of more than 600. In fact, one need not even store the whole fingerprint from an audio document, but sub-fingerprints would suffice. For example, longer documents were identified from their much shorter fingerprint sections without significant performance degradation. The proposed hashing scheme can capture in real-time a target record by scanning longer records, like continuous broadcasting. It, however, cannot be applied in general when intra- and interspeaker variability is involved.

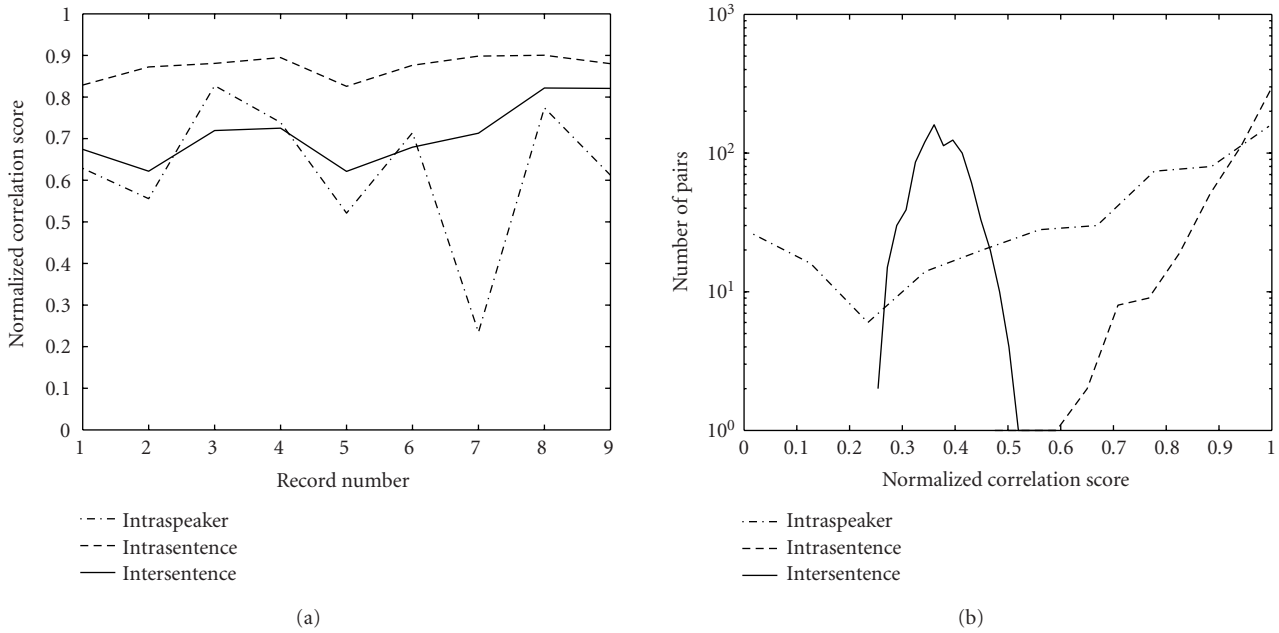


Figure 13: Typical hash correlations for speech: (a) three different sentences uttered by the same speaker 10-fold with possibly differing prosody; (b) histogram of hash correlations for speech (5 speakers, 10 sentences, 10 utterances). Intraspeaker graph: same speaker and same sentences, but different utterances; intersentence graph: same speaker but different sentences; intrasentence graph: distorted versions of the same record with attacks as in Table 1.

There are several avenues along which this research will proceed. Two of the immediate problems are the capacity assessment and binarization of the hash functions. Firstly, as the database climbs into tens of thousands of audio documents, the identification and verification capacity of the hash functions remains to be determined. Secondly, the hash functions need to be converted into a binary string. Various quantization strategies can be envisioned, such as random quantization [5] and median-based quantization [21] or an appropriate vector quantization, such as tree vector quantization for computational efficiency. The binarization is needed on one hand for both storage and more rapid search. A judicious binary representation, for example, coarse-to-fine content representation, can accelerate the database search.

REFERENCES

[1] J. S. Seo, J. Haitsma, T. Kalker, and C. D. Yoo, "A robust image fingerprinting system using the Radon transform," *Signal Processing: Image Communication*, vol. 19, no. 4, pp. 325–339, 2004.

[2] R. Radhakrishnan and N. Memon, "Audio content authentication based on psycho-acoustic model," in *Proc. Electronic Imaging 2002, Security and Watermarking of Multimedia Contents IV*, vol. 4675 of *Proceedings of SPIE*, pp. 110–117, San Jose, Calif, USA, January 2002.

[3] V. Roth and M. Arnold, "Improved key management for digital watermark monitoring," in *Proc. Electronic Imaging 2002, Security and Watermarking of Multimedia Contents IV*, P. W. Wong and E. J. Delp, Eds., vol. 4675 of *Proceedings of SPIE*, pp. 652–658, San Jose, Calif, USA, January 2002.

[4] T. Kalker, J. Haitsma, and J. Oostveen, "Robust audio hashing for content identification," in *Proc. International Workshop on Content Based Multimedia Indexing (CBMI '01)*, Brescia, Italy, September 2001.

[5] M. K. Mihçak and R. Venkatesan, "A perceptual audio hashing algorithm: a tool for robust audio identification and information hiding," in *Proc. Information Hiding*, pp. 51–65, Pittsburgh, Pa, USA, April 2001.

[6] C. J. Burges, J. C. Patt, and S. Jana, "Distortion discriminant analysis for audio fingerprinting," *IEEE Trans. Speech Audio Processing*, vol. 11, no. 3, pp. 165–174, 2003.

[7] F. Kurth and R. Scherzer, "Robust real-time identification of PCM audio sources," in *114th Convention of Audio Engineering Society*, Amsterdam, the Netherlands, March 2003.

[8] S. Sukittanon and L. E. Atlas, "Modulation frequency features for audio fingerprinting," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '02)*, vol. 2, pp. 1773–1776, Orlando, Fla, USA, May 2002.

[9] M. Gruhne, "Robust audio identification for commercial applications," *Fraunhofer IIS, AEMT*, 2003.

[10] L. Lu, H. Jiang, and H. J. Zhang, "A robust audio classification and segmentation method," in *Proc. ACM Multimedia (MM '01)*, pp. 203–211, Ottawa, Canada, September–October 2001.

[11] J. T. Foote, "Content-based retrieval of music and audio," in *Multimedia Storage and Archiving Systems II*, vol. 3229 of *Proceedings of SPIE*, pp. 138–147, Dallas, Tex, USA, November 1997.

[12] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. 1st International Symposium on Music Information Retrieval (ISMIR '00)*, Plymouth, Mass, USA, October 2000.

[13] T. Zhang and C. C. J. Kuo, "Hierarchical classification of audio data for archiving and retrieving," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '99)*, vol. 6, pp. 3001–3004, Phoenix, Ariz, USA, March 1999.

[14] R. Tucker, "Voice activity detection using a periodicity measure," *IEE Proceedings-I: Communications, Speech and Vision*, vol. 139, no. 4, pp. 377–380, 1992.

[15] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1978.

[16] J. Fitch and W. Shabana, "A wavelet-based pitch detector for musical signals," in *Proc. 2nd COST-G6 Workshop on Digital Audio Effects*, pp. 101–104, Trondheim, Norway, December 1999.

[17] M. J. Irwin, "Periodicity estimation in the presence of noise," in *Proc. Institute of Acoustics Conference*, Windemere, UK, November 1979.

[18] D. H. Friedman, "Pseudo-maximum-likelihood speech pitch extraction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, no. 3, pp. 213–221, 1977.

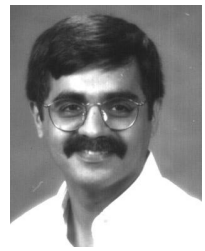
[19] D. Wu, D. Agrawal, and A. E. Abbadi, "Efficient retrieval for browsing large image databases," in *Proc. 5th International Conference on Knowledge Management*, pp. 11–18, Rockville, Md, USA, November 1996.

[20] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. International Conference on Image Processing (ICIP '00)*, vol. 3, pp. 664–666, Vancouver, British Columbia, Canada, September 2000.

[21] B. Coskun and B. Sankur, "Robust video hash extraction," in *Proc. European Signal Processing Conference (EUSIPCO '04)*, Vienna, Austria, September 2004.

Hamza Özer was born in Erzincan, Turkey, on August 10, 1973. He received the B.S., M.S., and Ph.D. degrees all in electrical and electronics engineering from Middle East Technical University, Başkent University, and Boğaziçi University, Turkey, in 1996, 1998, and 2005, respectively. From 1996 to 1999, he was with the Department of Electrical and Electronics Engineering, Başkent University, as a Research Assistant. Since June 1999, he has been a Senior Researcher at the National Research Institute of Electronics and Cryptology (UEKAE). His research interests are in signal processing and applications, data hiding, audio watermarking, robust audio hashing, time-frequency signal analysis, speech processing, image processing, development of test and measurement plan and setup.

Bülent Sankur has received his B.S. degree in electrical engineering from Robert College, Istanbul, and completed his M.S. and Ph.D. degrees at Rensselaer Polytechnic Institute, New York, USA. He has been teaching at Boğaziçi (Bosphorus) University in the Department of Electrical and Electronics Engineering. His research interests are in the areas of digital signal processing, image and video compression, biometry, cognition, and multimedia systems. Dr. Sankur has held visiting positions at the University of Ottawa, Technical University of Delft, and École Nationale Supérieure des Télécommications, Paris. He was the Chairman of ICT'96 (International Conference on Telecommunications) and EUSIPCO'05(The European Conference on Signal Processing) as well as the Technical Chairman of ICASSP'00.



Nasir Memon is a Professor in the Computer Science Department, Polytechnic University, New York. Professor Memon's research interests include data compression, computer and network security, and multimedia communication, computing, and security. He has published more than 200 articles in journals and conference proceedings. He was a Visiting Faculty at Hewlett-Packard Research Labs during the academic year 1997–1998. He has won several awards including the NSF CAREER Award and the Jacobs Excellence in Education Award. He was an Associate Editor for the IEEE Transactions on Image Processing from 1999 till 2002. He is currently an Associate Editor for the IEEE Transactions on Information Security and Forensics, ACM Multimedia Systems Journal, and the Journal of Electronic Imaging.

Emin Anarım received his B.S. (first-class honors) and the M.S. and Ph.D. degrees in electronics and electrical engineering from Boğaziçi University, Istanbul, Turkey, in 1981, 1983, and 1985, respectively. After spending 17 years in the Turkish Air Force Army as an officer, in January 1992, he joined the Department of Electrical and Electronics Engineering, Boğaziçi University, Istanbul, Turkey, where he is presently a Full-Time Professor. Dr. Emin Anarım is also presently an Adjunct Professor at George Washington University. He has given several courses to industry and government institutions on signal processing, video coding, secure telecommunications, mobile communications, lawful interception techniques in telecommunications, crytographic techniques, and network security.