On the comparison of audio fingerprints for extracting quality parameters of compressed audio

P.J.O. Doets, M. Menor Gisbert and R.L. Lagendijk

Information and Communication Theory Group, Faculty of EEMCS,
Delft University of Technology, P.O. Box 5031, 2600 GA Delft

ABSTRACT

Audio fingerprints can be seen as hashes of the perceptual content of an audio excerpt. Applications include linking metadata to unlabeled audio, watermark support, and broadcast monitoring. Existing systems identify a song by comparing its fingerprint to pre-computed fingerprints in a database. Small changes of the audio induce small differences in the fingerprint. The song is identified if these fingerprint differences are small enough. In addition, we found that distances between fingerprints of the original and a compressed version can be used to estimate the quality (bitrate) of the compressed version.

In this paper, we study the relationship between compression bit-rate and fingerprint differences. We present a comparative study of the response to compression using three fingerprint algorithms (each representative for a larger set of algorithms), developed at Philips, Polytechnic University of Milan, and Microsoft, respectively. We have conducted experiments both using the original algorithms and using versions modified to achieve similar operation conditions, i.e., the fingerprints use the same number of bits per second. Our study shows similar behavior for these three algorithms.

Keywords: audio fingerprint, audio hash, song identification, comparison, robustness, compression

1. INTRODUCTION

An audio fingerprint is a compact unique representation of an audio signal. It can be used to identify unlabeled audio based on the signal content. A fingerprinting system consists of two parts: fingerprint extraction and a matching algorithm. The fingerprints of a large number of songs are usually stored in a database, together with the metadata describing the content. A song is identified by comparing its fingerprint with the fingerprints in the database. The procedure for music identification using fingerprints is schematically shown in Figure 1.

Applications of audio fingerprinting include music identification using a cell phone, identification of songs/ commercials on the radio or television, and digital music library organization.[1] Snocap has recently attracted attention by using fingerprints for filtering in file sharing applications.[2] It's goal is to facilitate the use of Peer-to-Peer (P2P) networks for commercial music distribution. A similar idea was presented by Kalker et al. in their Music2Share paper.[3] Fingerprints can be used in a watermarking context to make watermarks that are content-dependent, to solve synchronization problems and to use watermarks to check whether audio content has been altered.[1,4]

Audio fingerprinting can be used to identify music. e.g. on the Internet. Usually this is done by searching through the metadata describing the music content. This metadata, however, is often incorrect, incoherent or incomplete. This problem can be avoided by using audio fingerprinting techniques to identify the audio based on its content instead of the metadata.

Songs on the Internet, however, are usually stored in a compressed data format such as MP3. Compression affects the perceptual quality of the content. The perceptual quality of a song compressed using MP3 at a bitrate of 32 kbps is totally different from the perceptual quality of the CD-recording of the same song. Therefore, a

Further author information: (Send correspondence to Peter Jan Doets)
Peter Jan Doets: E-mail: p.j.o.doets@tudelft.nl, Telephone: +31 15 2783635
Inald Lagendijk: E-mail: r.l.lagendijk@tudelft.nl, Telephone: +31 15 2783731

Security, Steganography, and Watermarking of Multimedia Contents VIII, edited by Edward J. Delp III, Ping Wah Wong,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6072, 60720L, © 2006 SPIE-IS&T · 0277-786X/06/$15

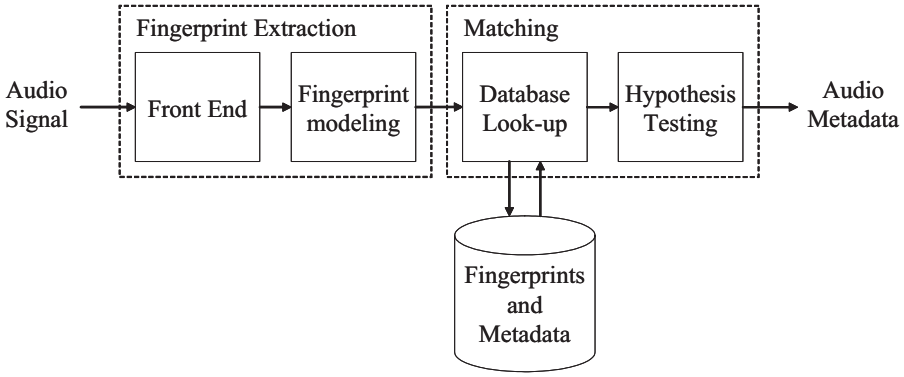


Figure 1. Music identification using audio fingerprinting.

content-based indication for the perceptual quality is needed. We intend to use fingerprinting to assess the perceptual quality of a compressed song after it has been identified.

Compression not only affects the perceptual quality of the audio content, but also induces small changes in the fingerprint. The differences between the fingerprint of the original content and the compressed content is dependent on the compression bitrate. Therefore, we are interested in the relation between compression bitrate and the differences between the fingerprints.

From previous work we know that the difference between the fingerprint of a song and its compressed version is related to the compression bitrate.[5,6] Figure 2(a) schematically illustrates this relation. We have shown that fingerprint differences can be used to indicate the perceptual quality of compressed content.[5,6] This implies that we need to use this relation between compression bitrate and fingerprint differences the other way around, i.e. given the fingerprint difference, what approximately was the compression bitrate, as shown in Figure 2(b). Therefore, the variance of the fingerprint difference for a given bitrate is very important for our intended application.

In our previous work we have focussed on one particular fingerprinting system only, developed by Philips.[7] We have worked on a statistical analysis of its fingerprint extraction process assuming uncorrelated signals.[5] In the Philips algorithm the difference between two fingerprints is expressed using the Bit Error Rate (BER). In previous work we have studied the differences between the Philips fingerprint of an uncorrelated signal and the Philips fingerprint of the same signal corrupted by additive noise as a function of Signal-to-Noise Ratio (SNR).[6] The relation between SNR and BER is given by:

$$\mathrm{BER} = \frac{1}{\pi} \arctan\left(\sqrt{\frac{\sigma_W^4}{\sigma_X^4} + 2\frac{\sigma_W^2}{\sigma_X^2}} \right), \tag{1}$$

where σ_W^2 and σ_X^2 denote the variance of the noise and the signal, respectively. SNR, of course, is defined as $20 \log_{10}(\sigma_X/\sigma_W)$. This relation has been confirmed experimentally. The shape of the curve also holds for the SNR-BER relations for real-life audio signals in the presence of noise, both additive and due to compression.

In this paper we extend our experimental results on the response to compression of the Philips algorithm[7] to two other algorithms, selected to be representative for a wide variety of existing fingerprinting algorithms. We focus on the fingerprint extraction only, and do not consider the problem of finding a matching fingerprint in a large database.

This paper is organized as follows. Section 2 gives an overview of existing audio fingerprinting algorithms, and selects three algorithms according to a set of criteria. Section 3 presents more details on the selected algorithms. Section 4 outlines adjustments to the algorithms presented in Section 3 such that a fair comparison can be made, and presents the experimental results. Section 5 draws conclusions.

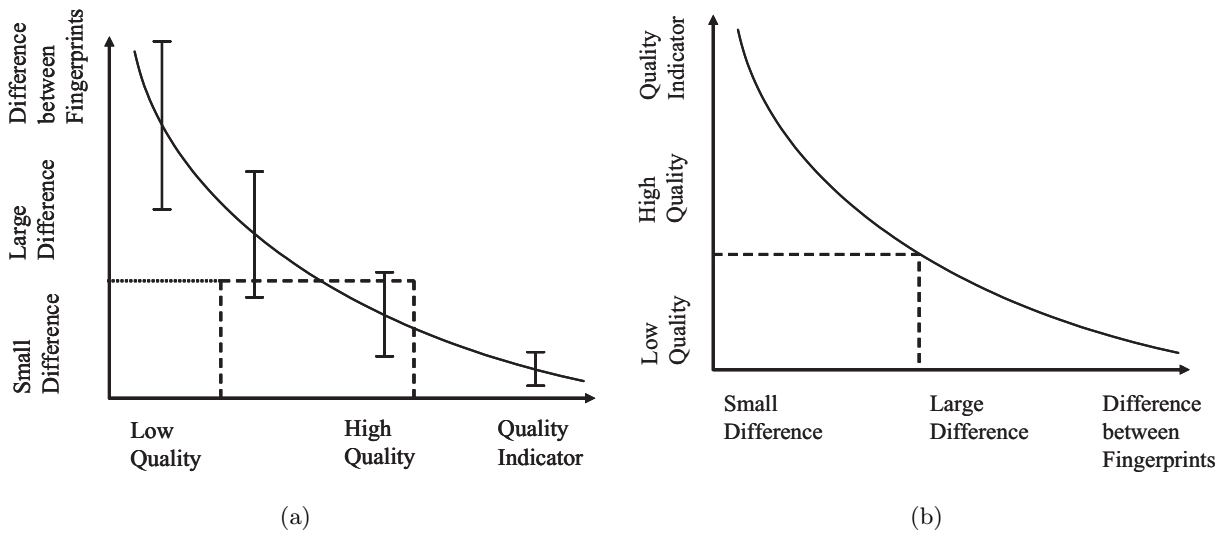


Figure 2. Illustration of (a) the measurable relation between quality of compressed music, e.g. compression bitrate, and fingerprint differences (b) our intended use: given the difference between fingerprints of an original recording and a compressed version.

2. AUDIO FINGERPRINTING ALGORITHMS

Cano et al. present a good survey of fingerprinting algorithms.[1] In their paper they identify a number of steps and procedures common to almost all audio fingerprinting systems. Figure 3 shows a schematic view of the steps in the fingerprint extraction process. In the pre-processing step the audio signal is usually converted to mono and downsampled to a lower sample rate. Then, the signal is divided into overlapping frames. Each frame is multiplied by a window before converting the frame to a spectral representation. Features are extracted from the time-frequency representation. Each feature is then represented by number of bits in the post-processing step.

In the last couple of years a lot of different fingerprinting systems have been developed by several institutions and companies. The most distinct differences between the algorithms found in literature are due to the (time-frequency) features that are used.

To cover the range of fingerprinting systems as good as possible in our comparison, we categorize the systems in to three groups. The features used in the first group are based on a combination of subband energies. The second group uses one subband to extract a feature. The third group uses a training procedure generate features which are optimized to use a combination of subbands. Within each group we have ranked the algorithms based on a number of criteria. The algorithm with the highest score within a group has been selected to represent that group of algorithms and is used in our experiments. We have ranked the algorithms using three criteria:

1. The algorithm is robust to compression, i.e. the algorithm is capable of identifying a song distorted by compression, while the distance measure reflects the effects of compression;
2. The algorithm is reported to be robust to common distortions.
3. The fingerprinting system is described well enough to be implementable;

Using these criteria, we have selected one algorithm to represent each group:

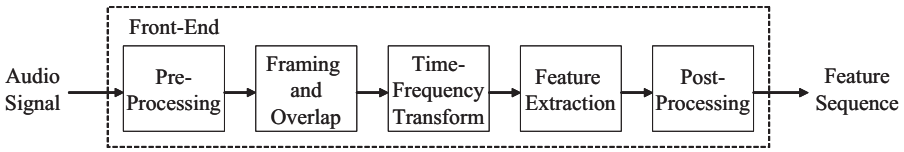


Figure 3. Fingerprint extraction procedure.

- Group 1: Systems that use features based on multiple subbands.
  Philips' Robust Hash (PRH) uses the sign of the difference between energies in Bark scaled frequency bands.[7] While it is reported to be highly robust against distortions,[7] the difference between fingerprints of original and compressed content does reflect compression artifacts.[6]

- Group 2: Systems that use features based on a single band
  Shazam has developed a fingerprinting algorithm to identify music using a cell phone.[9] It uses to peaks in the spectrogram to represent the fingerprint. The main principles are described in literature, but not detailed enough to be directly implementable. Furthermore, we expect the algorithm not to reflect the distortions related to compression, especially on medium and high bitrates. Özer et al. propose to use periodicity estimators and a Singular Value Decomposition of the Mel Frequency Cepstral Coefficient (MFCC) matrix.[10] Reported results are limited to speech and don't treat the robustness to audio compression. Sukkittanon and Atlas propose frequency modulation features.[11] The response to compression is not mentioned in the paper. Both Fraunhofer's AudioID and the algorithm developed by Mapelli et al. of Milan's Polytechnical University uses the Spectral Flatness Measure (SFM) and Spectral Crest Factor (SCF) to represent the fingerprint.[12, 13, 17] The latter algorithm is well-defined and the response to compression is discussed in literature. Based on its reported response to compression and its full description, we have selected the latter algorithm, to represent this category. In the remainder of this paper we will refer to this algorithm by the abbreviation SFCF (Spectral Flatness/Crest Factors). MusicDNA was developed by Cantametrix, Inc..[8] It uses global mean and standard deviation of the energies within 15 subbands of 15 seconds of music, thus creating a 30 dimensional vector. The effect of moderate compression is shown to be minimal.

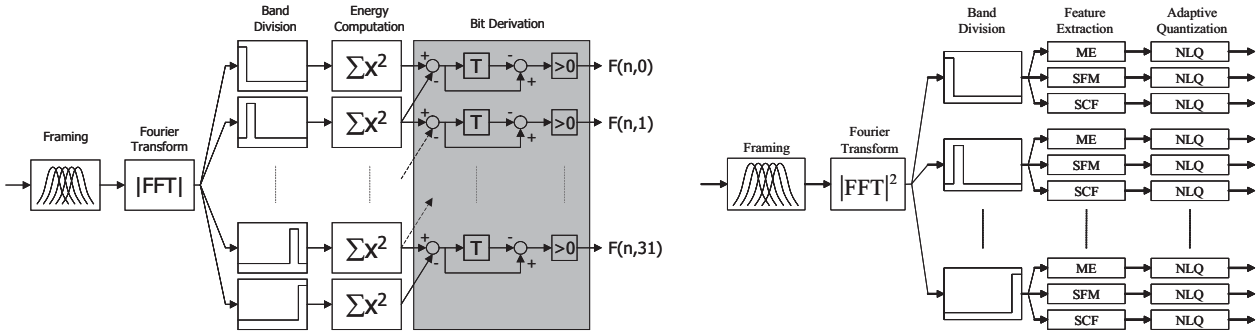
- Group 3: Systems using a combination of subbands or frames, which is optimized through training
  Batlle et al use Hidden Markov Models (HMMs) to describe their fingerprint.[14] The HMMs are trained based on audio examples. Identification is done using the Viterbi Algorithm. A second algorithm from UPF interprets the states sequences of the HMMs as 'Audio Genes' and uses techniques from bio-informatics to identify the audio.[15] Both systems use complex distance measures and implementation is far from straightforward. Microsoft Research uses dimensionality reduction techniques to extract the fingerprint in their Robust Audio Recognition Engine (RARE). The 2 stage dimension reduction is based on training using examples. Compression artifacts are reflected in the distances between fingerprints of the original and the compressed content. Therefore, we select Microsoft's RARE to represent the third category of algorithms.

3. SELECTED ALGORITHMS

In this section we will present the three selected fingerprinting algorithms into more detail. The algorithms developed by Philips, the Polytechnical University of Milan and Microsoft are discussed in Section 3.1, 3.2 and 3.3, respectively.

3.1. PRH

Figure 4(a) shows the fingerprint extraction of the Philips algorithm.[7] Like in most systems the audio signal is first converted to mono and downsampled to an appropriate lower sample frequency. The pre-processed signal is then divided into (strongly) overlapping frames, which are multiplied by a Hanning window, and transformed into a spectral representation using a periodogram estimator.



(a) (b)

Figure 4. Audio fingerprinting extraction for two algorithms (a) PRH.[7,16] T indicates a unit-time delay. (b) SFCF[13,17]

Within each frame, the energy within a number of frequency bands is computed. To match the properties of the Human Auditory System (HAS) the bandwidth of the frequency bands increases logarithmically with frequency, so to imitate the Bark scale. We denote the energy of frequency band m of frame n by $E(n,m)$. Energy differences are computed in time and frequency:

$$ED(n,m) \quad = \quad (E(n,m) - E(n,m+1)) - (E(n-1,m) - E(n-1,m+1)). \tag{2}$$

The bits of the sub-fingerprint are derived by

$$F(n,m) = \begin{cases} 1 & ED(n,m) > 0 \\ 0 & ED(n,m) \leq 0 \end{cases}, \tag{3}$$

where $F(n,m)$ denotes the m^{th} bit of sub-fingerprint n. Due to the strong frame-overlap there is strong correlation between the fingerprint bits along the temporal dimension.

3.2. SFCF

Figure 4(b) shows the fingerprinting algorithm proposed by Mapelli et al.[13,17] Like the Philips algorithm, they extract features from strongly overlapping periodograms. The extracted features are the mean energy (ME), the Spectral Flatness Measure (SFM) and the Spectral Crest Factor (SCF). The original algorithm uses no subbands and extracts all three features per frame. All features are based on the arithmetic and geometric means of (subband) energies. Define the arithmetic mean of signal $x(i), \quad i = 1, \ldots, N$, as:

$$M_a\left(x(i)|i = 1, \ldots, N\right) = \frac{1}{N}\sum_{i=1}^{N} x(i) \tag{4}$$

and the geometric mean as:

$$M_g\left(x(i)|i = 1, \ldots, N\right) = \sqrt[N]{\prod_{i=1}^{N} x(i)} \tag{5}$$

The ME, SFM and SCF features are defined as:

$$ME(n,m) \quad = \quad M_a\left(S(n,k)|k \in B_m\right) \tag{6}$$

$$\begin{aligned} SFM(n,m) &= 10\ {}^{10}\log\left(\frac{M_g\left(S(n,k)|k \in B_m\right)}{M_a\left(S(n,k)|k \in B_m\right)}\right) \\ &= 10\ M_a\left({}^{10}\log\left(S(n,k)|k \in B_m\right)\right) - 10\ {}^{10}\log\left(M_a\left(S(n,k)|k \in B_m\right)\right) \end{aligned} \tag{7}$$

$$\begin{aligned} SCF(n,m) &= 10\ {}^{10}\log\left(\frac{\max\left(S(n,k)|k \in B_m\right)}{M_a\left(S(n,k)|k \in B_m\right)}\right) \\ &= 10\ {}^{10}\log\left(\max\left(S(n,k)|k \in B_m\right)\right) - 10\ {}^{10}\log\left(M_a\left(S(n,k)|k \in B_m\right)\right), \end{aligned} \tag{8}$$

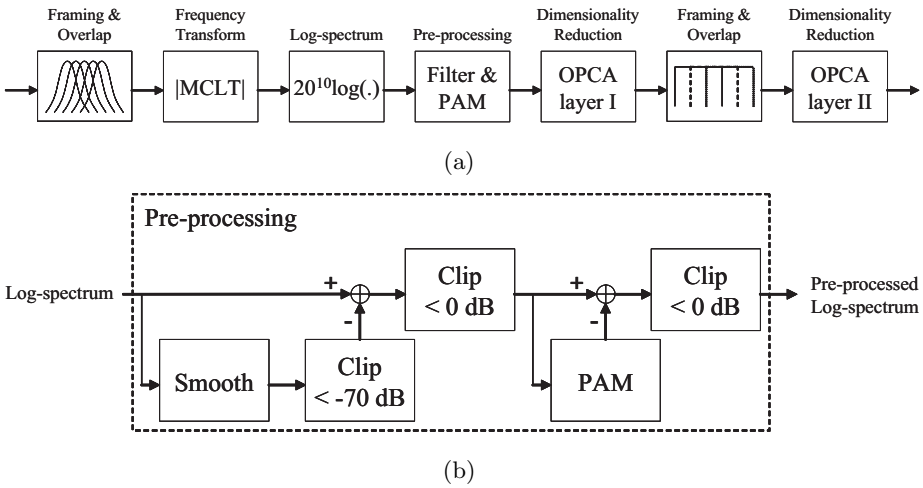


Figure 5. Microsoft's Robust Audio Recognition Engine (RARE)[18] (a) Fingerprint Extraction (b) Pre-processing.

where B_m defines subband m.

Within each band, each feature is quantized using a non-uniform quantizer. Mapelli et al. report that using a uniform quantizer, the quantization levels of a fingerprint are not robust enough when the signal is distorted. Therefore, the feature time series is quantized such that the quantization levels follow a uniform distribution. The original system uses 4 bits to quantize each feature. The Mean Square Error (MSE) is used as the distance measure between two fingerprints.

3.3. RARE

Also in Microsoft's RARE shown in Figure 5(a) the audio signal is first converted to mono, downsampled, and segmented into overlapping frames. The Modulated Complex Lapped Transform (MCLT) is used as the time-frequency representation of the data. Other than the previously described systems, the Microsoft system uses the log power spectrum.

Instead of extracting features from the spectral representation, it uses two projections to reduce the dimensionality of the audio data. Each projection is the result of Oriented Principle Component Analysis (OPCA) which uses both undistorted and distorted data for training. OPCA projects the data onto directions that maximizes the signal to noise ratio of the training data.

We will now describe the training procedure. Let the original signal be represented by L-dimensional column vectors $\underline{x}_i \in R^L$. Assume that for each vector $\underline{x}_i$ a set of Q distorted versions $\underline{\tilde{x}}_i^k \in R^L$ is available for training. These can be used to compute the difference between each original vector and each distortion:

$$\underline{z}_i^k \equiv \underline{\tilde{x}}_i^k - \underline{x}_i, \qquad i = 1, \ldots, P; \quad k = 1, \ldots, Q; \tag{9}$$

An OPCA layer is trained using the covariance matrices C_x and C_z:

$$C_x = \frac{1}{P} \sum_{i=1}^{P} \underline{x}_i \underline{x}_i' - \underline{\mu}_x \underline{\mu}_x' \tag{10}$$

$$C_z = \frac{1}{PQ} \sum_{i=1}^{P} \sum_{k=1}^{Q} \left(\underline{z}_i^k\right) \left(\underline{z}_i^k\right)' - \underline{\mu}_z \underline{\mu}_z' \tag{11}$$

to formulate the generalized eigenvalue problem:

$$C_x \underline{n} = \lambda C_z \underline{n}, \tag{12}$$

Table 1. System parameters for the modified versions of all system.

	PRH	SFCF	RARE
Frame overlap ratio	31/32	31/32	1/2
# Bits per feature	1	4	32
# Frequency bands	17	4	2048
Distance measure	BER	MSE	MSE

where $'$ denotes the transpose operation. The generalized eigenvectors represent the directions where the average Signal-to-Noise Ratio (SNR) of the training data ($\underline{x}_i$ and $\underline{z}_i^k$) is maximized. The full dimensionality reduction is done in a two stage process.

The first OPCA layer is trained using the covariance matrix of the pre-processed log spectra of the original signal, and the correlation matrix of the difference between the pre-processed log spectra of the original signal and its distorted versions. The resulting projection (RARE uses a projection onto 64 dimensions) is scaled such that the lower dimensional representation of the noise has unit variance in all dimensions and an offset is added such that the signal has zero mean in all dimensions. The resulting projection is then used to create a lower dimensional representation of the log-spectrum of each frame.

The second OPCA layer is trained using the lower dimensional representations of the signal and noise vectors. A number of output vectors of the first layer are concatenated before serving as input to the second OPCA layer.

Microsoft's RARE system uses the pre-processed log power spectrum to train the first OPCA layer. The pre-processing here consists of two steps shown in Figure 5(b). First, the difference is computed between the log magnitude spectrum and a smoothed version. All negative spectral coefficients are set to zero. This is done to remove the effects of amplitude scaling and equalization. Second, a simple Psycho-Acoustic Model is used to compute the masking threshold of the pre-processed spectrum. The difference between the pre-processed signal and the masking threshold is used as an input to the first OPCA layer. All negative values are set to zero.

4. ALGORITHMIC COMPARISON

For a fair comparison of the algorithms, they were slightly adjusted to let them operate under the same conditions. These conditions are o.a. the same false alarm rate and extraction of the same number of bits for a given segment. These modifications are described in Section 4.1. The actual comparison is done in three types of experiments. Section 4.2 describes how the systems deal with uncorrelated signals in the presence of additive noise. These experiments give an indication what to expect for the relation between noise due to compression and the difference in fingerprints, described in Section 4.3. Finally, Section 4.4 compares the fingerprinting systems on the aspect of fingerprint differences as a function of MP3 compression bitrate.

4.1. Enabling algorithmic comparison

The fingerprinting systems described in Section 2 not only use different features, but also have different sampling rates, granularity, etc. A fair comparison requires similar operating conditions. Therefore, we set the following parameters for all systems:

- Sampling rate of 5512.5 Hz
- Frequency bands between 300 and 2000 Hz for the PRH and SFCF system
- Fingerprint block length of about 3.1 seconds
- Framelength of 2048 samples (371.5 ms)
- Fingerprint block size of 4096 bits

Table 2. Comparison between system parameters for the original and modified version of the systems developed (a) by Philips and Polytechnic University of Milan, respectively; (b) Microsoft.

	PRH		SFCF	
	Original System	Modified System	Original System	Modified System
Sample rate [Hz]	5512.5	5512.5	44100	5512.5
Frequency Range [Hz]	300-2000	300-2000	300-3400	300-2000
Window length [ms]	371.5	371.5	743	371.5
Frame overlap ratio	31/32	31/32	63/64	31/32
# Bits er feature	1	1	4	4
# Frequency bands	33	17	1	4
# Features	1	1	3	1
# Frames per segment (sec.)	256 (3.1 s)	256 (3.1 s)	64 (1.5 s)	256 (3.1 s)

(a)

Microsoft	Original System	Modified System
Sample rate (Hz)	11025	5512.5
Window length (ms)	371.5	371.5
Frame overlap ratio	1/2	1/2
Overall OPCA reduction	$32 \times 2048 \rightarrow 64$	$16 \times 1024 \rightarrow 64$
Fingerprint block length (frames)	32 (6.2 s)	16 (3.1 s)
Overlap ratio in 2^{nd} OPCA layer	0	1/2

(b)

In order to achieve these settings, we can modify the frame overlap ratio, the number of frequency bands, the number of features, the number of bits to represent each feature. In addition we have changed the overlap ratio in the second OPCA layer of Microsoft's RARE system. Table 1 compares the settings for the different systems, Table 2 compares the original system with its modified counterpart.

We have used 275 song fragments of 40 seconds each; 100 of these fragments have been used for training Microsoft's RARE system. This is in the same order of magnitude as the number of songs mentioned in their paper. For each of these 100 song fragments we have generated 9 distorted versions. These distortions are mainly non-linear amplitude distortions and two pitch shifts. Compression is not one of the distortions.

We have used MP3 compression using the LAME codec.[19] The selected bitrates for MP3 compression range from 32-256 kilobit-per-second (kbps) using constant bitrate. For each system we have set a threshold for identification, such that all system operate under the same false alarm rate per fingerprint block, P_{fa}. The P_{fa} is based on a Gaussian approximation of the distances between fingerprint blocks of original, undistorted fragments. We have chosen $P_{fa} = 10^{-5}$. Although this is quite low for a practical fingerprinting system, it is achievable for all systems and we are interested in the relation between compression and fingerprint distance, given a fixed P_{fa}.

4.2. Signal-to-Noise vs. Fingerprint differences for uncorrelated data

Equation 1 analytically relates the (mean) distance between the PRH fingerprint of an uncorrelated signal and the PRH fingerprint of the same signal in the presence of additive noise to the SNR. For this paper, We have studied *experimentally* the relation between SNR and fingerprint differences due to additive noise for uncorrelated signals for the PRH, SFCF and RARE algorithms.

Figure 6 shows the experimental relation between SNR and fingerprint differences for the PRH and RARE algorithm and the features of the SFCF algorithm. The curves have been normalized to a common P_{fa} level, as discussed in the previous section.

For all curves we can distinguish two or three regions. For very low SNR levels (below 0 dB), the curves are approximately flat. There, the additive noise is dominant and the fingerprint is not related to the fingerprint of

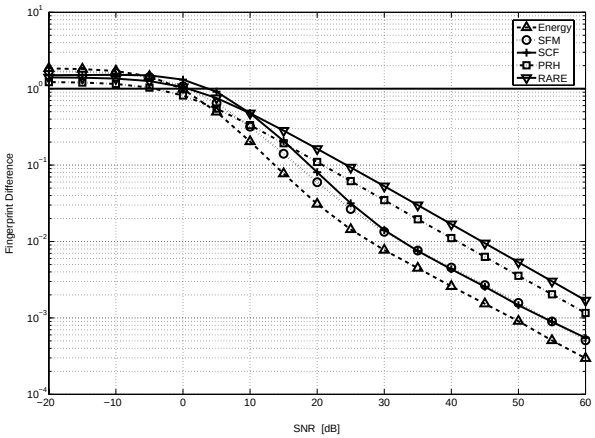


Figure 6. Comparison of fingerprinting features for uncorrelated signals as a function of SNR. SFCF features: Energy $(--\triangle)$, SCF $(-+)$, SFM $(\cdots\bigcirc)$; PRH features $(-.\square)$ and RARE features $(-\triangledown)$

.

the original signal. For high SNR levels all curves have the same steepness in the log-log plot. For high SNR levels, Equation 1 is approximated by:

$$\text{BER} \approx \frac{1}{\pi} \arctan\left(\sqrt{2}\ \frac{\sigma_W}{\sigma_X}\right). \tag{13}$$

Therefore, we can conclude that all curves for high SNR are approximately proportional to:

$$F_{diff} \propto \frac{\sigma_W}{\sigma_X}, \qquad \sigma_W \ll \sigma_X \tag{14}$$

In between these two regions, some of the features have a steeper angle. For the SCF and energy feature of the SFCF algorithm, the angle in the log-log plot is twice as steep. This implies that the fingerprint difference, F_{diff}, is proportional to the ratio σ_W^2/σ_X^2.

The overall conclusion of these curves is that we can expect the relationship between SNR-level and the fingerprint difference due to a certain compression bitrate is expected to result in straight lines in the log-log plots, with the same steepness.

4.3. Signal-to-Compression-Noise vs. Fingerprint differences

Figure 7 shows the Signal-to-Compression-Noise for the three algorithms. Figure 7(a)-7(c) compares the modified version with an implementation using settings described in literature.

The shading indicates the local spread in fingerprint differences of the curves. Due to the fact that in compression the bitrates are chosen, and the SNR levels are a result of the selected bitrate, it is not straightforward to indicate the spread in the curves. Since the points are not aligned on certain SNR levels, the shading indicates the 1/6-percentile and 5/6-percentile within an overlapping bin of SNR levels. The binning introduces the effect that the angle of the averaged curves changes slightly (becomes less steep). Curves for one single fragment show a clear relation between SNR and fingerprint difference: if the SNR is increased by 20 dB, the fingerprint difference becomes 10 times smaller.

After being normalized to achieve the common P_{fa}, some of the curves have been scaled, resulting in a vertical shift in the plot, to avoid overlap. The scaling factors are indicated in the caption of Figure 7. It is quite clear that all curves have approximately the same steepness in the SNR plots. This confirms the expected relation from the experiments in Section 4.2.

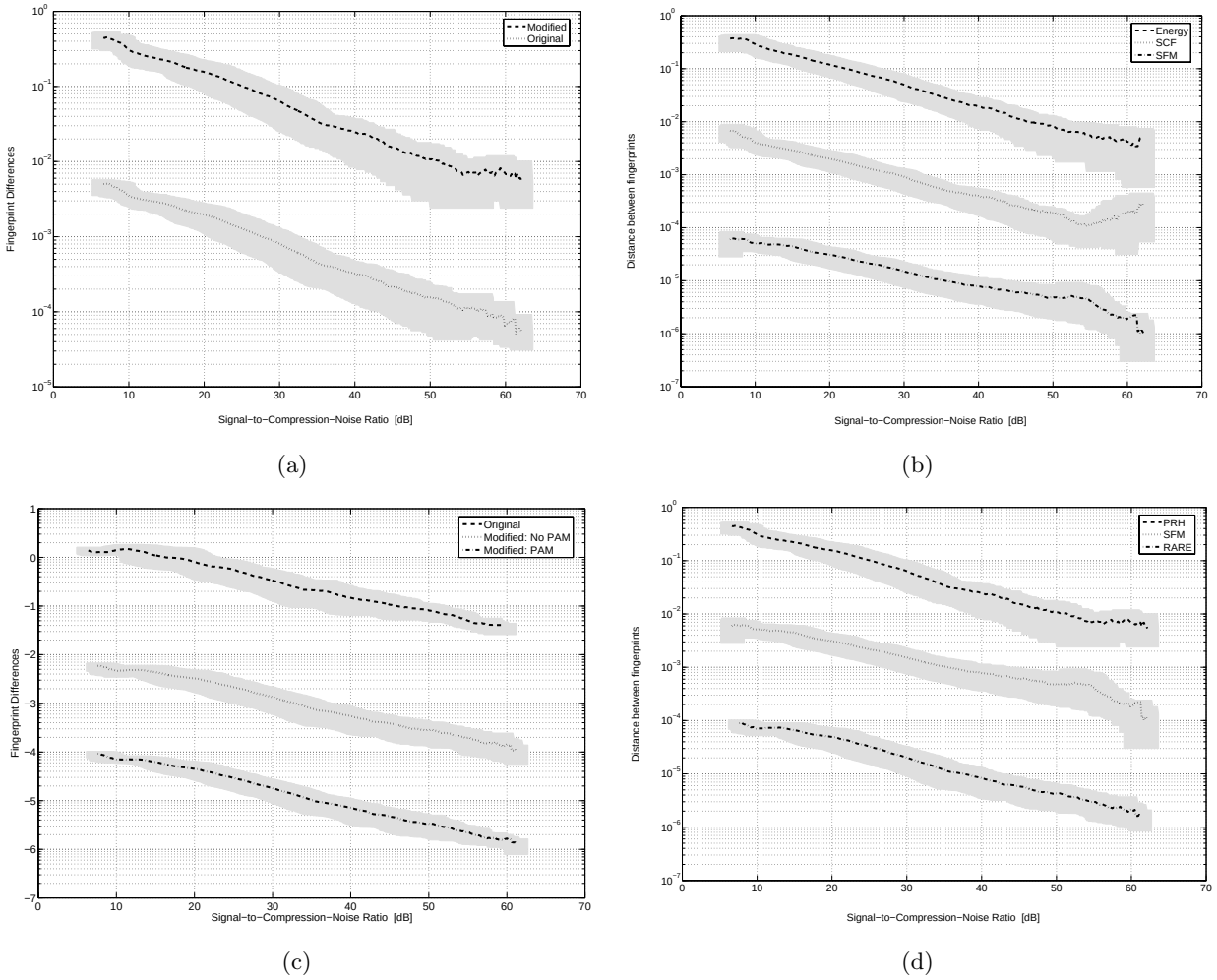


Figure 7. Compression bitrate vs. fingerprint differences. The curves have been scaled such that there is no overlap. (a) The features in the SFCF algorithm: From top to bottom: Energy (−−, not scaled), SCF (· · ·, scaled by factor 10^{-2}), SFM (−., scaled by factor 10^{-4}), (b) PRH: Modified (−−, not scaled), Original (· · ·, scaled by factor 10^{-2}), (c) RARE: Original (−−, not scaled), Modified, no Psycho-Acoustic Model (· · ·, scaled by factor 10^{-2}), Modified, using a Psycho-Acoustic Model (−., scaled by factor 10^{-4}) (d) Comparison between the modified versions of PRH (−−, not scaled), SFM (· · ·, scaled by factor 10^{-}), RARE (−., scaled by factor 10^{-4}).

4.4. Compression bitrate vs. Fingerprint differences

Figure 8 compares the relation between compression bitrate and fingerprint differences for the original algorithms with their modified counterparts. In general, the behavior of the modified algorithms is comparable to the algorithms using the original settings. Since the differences have been normalized such that the algorithms achieve a similar P_{fa}, the scale of the curves is related to the variance of the distribution of the fingerprints of the uncompressed songs.

All algorithms show similar behavior. There is no algorithm that has a significant lower ratio of standard deviation and mean.

5. CONCLUSION AND DISCUSSION

A wide variety of audio fingerprinting systems has been presented in literature over the last couple of years. Although each system is different, they share a number of steps and operations. The main difference between

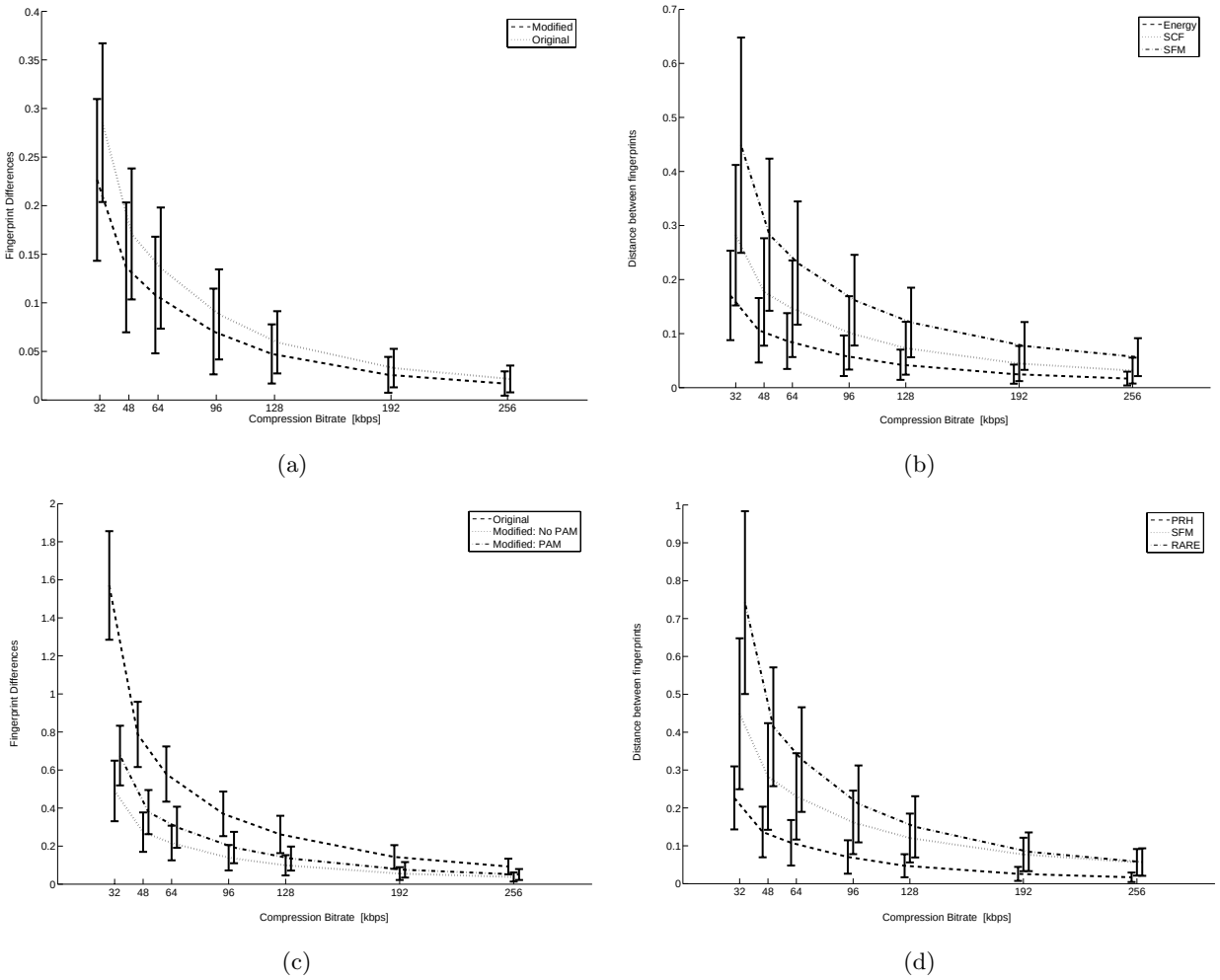


Figure 8. Compression bitrate vs. fingerprint differences. The curves have been scaled such that there is no overlap. (a) The features in the SFCF algorithm: From top to bottom: Energy (−−), SCF (· · ·), SFM (−.), (b) PRH: Modified (−−), Original (· · ·), (c) RARE: Original (−−), Modified, no Psycho-Acoustic Model (· · ·), Modified, using a Psycho-Acoustic Model (−.) (d) Comparison between the modified versions of PRH (−−), SFM (· · ·), RARE (−.).

the systems is the features that are used. In our comparison we are mainly concerned of the response to compression, i.e. the difference between the fingerprint of an original recording and a compressed version.

We have shown that although the features and projections that are used in the three systems that have been compared are very different, they behave in a comparable fashion. The differences are in the distribution of the differences between arbitrary fingerprints, the variance of the bitrate-fingerprint difference curves and in the steepness of the SNR-fingerprint difference curves. The model that relates the SNR to the BER for the PRH gives a good indication for the other algorithms as well.

The difference between fingerprints reflect the difference between an original recording and a compressed version and can be used to roughly estimate the quality of compressed content. The main obstacle for doing this is the large variance of the fingerprint difference for a given compression bit rate. All algorithms in our study suffer from a variance which is too large for our intended use. Furthermore, for comparing fingerprints to estimate the quality of compressed content it makes sense to use a psycho-acoustic model. In a file sharing application the amount of noise is very limited. If the music is very much corrupted by noise, e.g. heavily compressed, then it is useless to the average user anyway.

The fact that the systems behave more or less the same - the relation between compression bitrate and fingerprint differences and between noise and fingerprint differences have comparable shapes - gives the impression that there is more to fingerprinting than just extraction of robust features. There seems to be more common ground to behavior of the algorithms than the steps preceding the feature extraction. Therefore, it makes sense to analyze fingerprinting on a more abstract level, and to analyze the relation between compression and audio fingerprinting in general without looking at specific implementations or systems.

Besides this analysis, future work will include the reduction of the variance of the fingerprint differences for a given compression bitrate, the use of a proper psycho-acoustic model, such that main obstacles for using fingerprints to roughly estimate the quality of a song are removed.

REFERENCES

1. P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 169 – 173, December 2002. 4 pages.
2. Snocap, December 2005. http://www.snocap.com.
3. T. Kalker, D. Epema, P. Hartel, R. Lagendijk, and M. van Steen, "Music2share - copyright-compliant music sharing in p2p systems," *Proceedings of the IEEE* **92**(6), pp. 961 – 970, 2004. 10 pages.
4. S. Beauget, M. van der Veen, and A. Lemma, "Informed detection of audio watermark for resolving playback speed modifications," in *Workshop on Multimedia and Security (MM&Sec)*, pp. 117 – 123, 2004. 7 pages.
5. P. Doets and R. Lagendijk, "Stochastic model of a robust audio fingerprinting system," in *5th International Symposium on Music Information Retrieval (ISMIR)*, pp. 349 – 352, October 2004. 4 pages.
6. P. Doets and R. Lagendijk, "Extracting quality parameters for compressed audio from fingerprints," in *6th International Conference on Music Information Retrieval (ISMIR)*, pp. 498 – 503, September 2005. 6 pages.
7. J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *3rd International Symposium on Music Information Retrieval (ISMIR)*, October 2002. 9 pages.
8. V. Venkatachalam, L. Cazzanti, N. Dhillon, and M. Wells, "Automatic identification of sound recordings," *IEEE Signal Processing Magazine* **21**, pp. 92 – 99, March 2004. 8 pages.
9. A. Wang, "An industrial strength audio search algorithm," in *4th International Symposium on Music Information Retrieval (ISMIR)*, October 2003. 7 pages.
10. H. Ozer, B. Sankur, and N. Memon, "Robust audio hashing for audio identification," in *12th European Signal Processing Conference (EUSIPCO)*, September 2004. 4 pages.
11. S. Sukittanon, L. Atlas, and J. Pitton, "Modulation-scale analysis for content identification," *IEEE Transactions on Signal Processing* **52**, pp. 3023 – 3035, October 2004. 13 pages.
12. J. Herre, O. Hellmuth, and M. Cremer, "Scalable robust audio fingerprinting using mpeg-7 content," in *5th IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 165 – 168, October 2002. 4 pages.
13. F. Mapelli and R. Lancini, "Audio hashing technique for automatic song identification," in *International Conference on Information Technology: Research and Education (ITRE)*, Augustus 2003.
14. E. Batlle, J. Masip, and E. Guaus, "Automatic song identification in noisy broadcast audio," in *IASTED International Conference on Signal and Image Processing*, August 2002. 6 pages.
15. H. Neuschmied, H. Mayer, and E. Batlle, "Content-based identification of audio titles on internet," in *1st IEEE International Conference on Web Delivering of Music (WEDELMUSIC)*, pp. 96 – 100, November 2001. 5 pages.
16. J. Haitsma and T. Kalker, "Speed-change resistant audio fingerprinting using auto-correlation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 728 – 731, April 2003. 4 pages.
17. F. Mapelli, R. Pezzano, and R. Lancini, "Robust audio fingerprinting for song identification," in *12th European Signal Processing Conference (EUSIPCO)*, September 2004.
18. C. Burges, J. Platt, and S. Jana, "Distortion discriminant analysis for audio fingerprinting," *IEEE Transactions on Speech and Audio Processing* **11**, pp. 165 – 174, May 2003. 10 pages.
19. LAME, December 2005. http://lame.sourceforge.net.