Perceptual Hashing for the Identification of Telephone Speech

Gary Grutzek[1], *Julian Strobl*[1], *Bernhard Mainka*[1], *Frank Kurth*[2], *Christoph Pörschmann*[1], *Heiko Knospe*[1]

[1] Institute of Communications Engineering, Cologne University of Applied Sciences, 50679 Cologne, Germany
Email: {gary.grutzek,christoph.poerschmann,heiko.knospe}@fh-koeln.de
Web: http://viat.fh-koeln.de/

[2] Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, 53343 Wachtberg, Germany
Email: frank.kurth@fkie.fraunhofer.de
Web: http://www.fkie.fraunhofer.de/

Abstract

The hashing of audio content for the identification of specific recordings and their degradations has many applications. In particular music identification is well established. In this paper, the perceptual hashing of speech is investigated and applied to the content-based identification of telephone spam. Based on well-known audio fingerprinting methods, various modifications and extensions have been developed and compared. We explore index-based search methods in order to match sequences of feature vectors. We investigate the influence of the hash size on the recognition rate and in particular the search efficiency in a large and and constantly updated fingerprint database like in a telephone speech scenario. It is shown that two 32-bit hashes with a unique time-distance allow for an efficient identification of telephone speech within a large call database.

1 Introduction

Standard cryptographic hash functions have the property that minimal alterations of input data significantly change the resulting hash value (avalanche effect). In contrast, robust hashes are locality-sensitive [1] and similar input data result in identical or close hash values. In particular, the audio hashes should be invariant to (certain) content degradations and hence permit a perceptual audio comparison and an efficient identification of a specific recording. They should hence provide a compact, unique and robust description of audio material.

A number of different mechanisms for audio identification already exist which are mostly based on spectral audio features for short-time frames. The complete fingerprint is composed of short-time hashes and usually represented by a sequence of vectors. Depending on the algorithm, two audio pieces are identified if a certain number of hash vectors coincide or have a small distance. Audio hashes have already successfully been employed to match pieces of music in large song databases. In this article, different fingerprinting methods based on [2] are being investigated and efficient index-based techniques are used for fast searching in speech data.

The audio hash and the search algorithm described here have been developed in the context of the VIAT project [3]. A main objective of this project is to detect replayed spam calls. These calls use pre-recorded audio material which is distributed in large quantities by automated phone calling systems. Using fingerprinting methods, the replayed calls can be efficiently identified and, depending on the policy, subsequent calls from the same caller can be blocked.

This paper is organized as follows: the next section presents work related to fingerprinting for audio identification. Then, our construction of robust audio hashes is explained. The following section presents an index-based method for an efficient identification of matching sequences of sub-hashes within a large database. Finally, the recognition rate and the search efficiency of the proposed hashing method is analyzed.

2 Related Work

A number of robust audio hashing algorithms have been proposed in the last decade. In this section, we present several approaches which are related to our work.

A first basic method was *AudioID* [4] which used spectral flatness (SFM) and spectral crest factor (SCF) from the MPEG-7 standard to identify audio material. SFM and SCF describe the flatness and tonality of the audio signal's spectrum.

J. Haitsma and T. Kalker's [2] fingerprinting method is based on energy differences of spectral subbands. Here, three seconds of audio material are segmented into frames with an overlap factor of $31/32$. For each frame the spectral coefficients are extracted and filtered by a mel filter bank. The sign of the energy differences between the subbands yields binary data and forms a sub-hash vector. The complete fingerprint consists of 256 sub-hashes which are 32 bits long each. This approach turns out to be promising for the identification of speech in our scenario.

Index-based approaches are *audentify!* [5], *Shazam* [6, 7], and *audio matching* [8]. While *audentify!* uses temporal features based on code books, *Shazam* (hashes derived from spectrograms) and *audio matching* (quantized chroma features) use spectral features. Both provide an efficient search and allow for a possible time shift.

An approach using mel-frequency cepstral coefficients (MFCC) is described in [9]. The audio material is segmented into overlapping frames, from which the MFCC coefficients are extracted. The absolute values of the MFCC coefficients are arranged in an $M \times N$ - matrix. Then a non-negative matrix factorization (NMF) and a principal component analysis (PCA) are performed and yield the hash.

3 Robust Audio Hashing

As the audio hashes are used in a telephone environment they should be both robust against alterations due to codecs and channel influences like noise, delay and packet loss. The system should be able to handle hundreds of queries

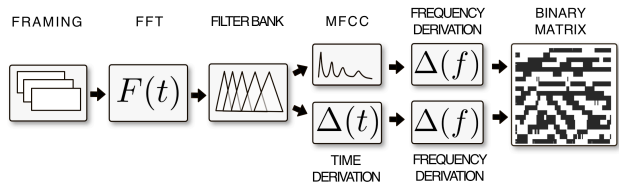

Figure 1: Feature extraction and hashing overview.

per second in a database with approximately one million calls. For this, the hash – besides being small in size and easy to compute – should support efficient search methods like the proposed index-based search.

As described in Section 4, one observes that the higher the number of (possible) hash values, the faster the search is. Unfortunately, increasing the number of feature classes also increases the bit-error rate and therefore deteriorates the robustness. The goal is to find a good balance between error rate and search speed. Another important aspect is to ensure that the amount of false positives is low in order not to incorrectly block regular callers.

3.1 Feature Extraction

The audio hash explained in the following is based on Haitsma and Kalker's [2] fingerprinting method. Taking their method as a starting point, further research has led to an adapted fingerprint. Besides the energy differences in frequency direction, an additional differentiation in time direction in combination with additional cepstral coefficients showed better results for speech signals. The feature extraction shown in Fig. 1 is done as follows:

A Fourier transform is applied to short overlapping frames of the spoken audio content. The spectral coefficients are filtered by a non-linear, mel filter bank to determine the energy in each subband. The resulting energy vector in the mel-spaced frequency domain is further processed to obtain MFCCs. The sequence of spectral values is differentiated in time and frequency direction whereas the sequence of cepstral coefficients is solely differentiated in frequency direction. Both series of vectors are then concatenated and for each concatenated vector a binary sub-hash vector H is computed (see below).

3.2 Implementation

The overlapping frames have a length of 370 ms and are extracted every 11.8 ms. The recorded audio samples have a length of 6 seconds and therefore result in 480 frames. The filter bank's bands are equally distributed on a logarithmic frequency axis between 300 Hz and 1.8 kHz. The resulting series of spectral and cepstral vectors is differentiated as described above.

For the test scenarios described in Section 5, the hash's bit length is varied. The number of spectral coefficients is directly specified by the number of bands of the filter bank, while some of the cepstral coefficients are omitted. The first cepstral coefficient is always excluded and the number of the following cepstral coefficients is varied between 8 and 24. For example, to obtain a 32 bit fingerprint, a filterbank with 21 bands is used and the cepstral coefficients from 2 to 14 were added. The differentiation results in 20 spectral and 12 cepstral coefficients, respectively.

To avoid false matches, only frames with an energy exceeding a certain threshold are processed. This also ensures that silence at the beginning of a call is not considered. Furthermore, in order to improve search speed only a maximum of 100 energy-rich frames form the complete audio hash.

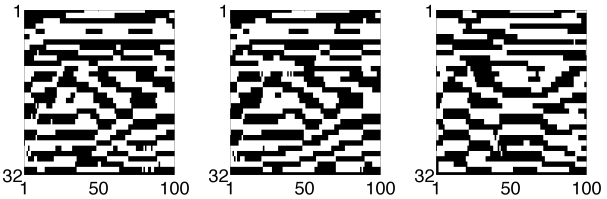

Figure 2: Binary hash matrix of three speech samples. Sample (a) is similar to (b), while (c) is dissimilar to both (a) and (b).

3.3 Hash

Since the search algorithm is based on the matching of integers, the resulting concatenated vectors are transformed into a sequence of integer values. This is achieved by only computing the signs of the differences while disregarding magnitudes:

$$H(n,m) = \begin{cases} 1 & \text{if } M(n,m)' \geq 0 \\ 0 & \text{if } M(n,m)' < 0. \end{cases} \quad (1)$$

An example for a resulting hash (as a binary matrix) is given in Fig. 2. The columns of the resulting binary matrix are *sub-hashes* and the vectors (or integers) correspond to one extracted audio frame.

4 Index Based Identification of Speech Samples

To efficiently identify short speech samples, an index-based technique, originally proposed in the domain of audio identification [5], can be suitably adapted. We first observe that by interpreting each column vector of the above binary feature matrices as an integer, the i-th audio fingerprint (sub-hash) can be interpreted as a set of features

$$D_i = \{[1,r_1],[2,r_2],\ldots,[m,r_m]\}.$$

A feature (t,r) consists of an integer r encoding the sub-fingerprint and an integer t specifying the features' temporal position. In our case this is not necessarily the column number within the matrix as frames with low energy are excluded. D_i is also called a feature document. By collecting feature documents $\mathscr{D} := (D_1,\ldots,D_N)$ corresponding to all available speech samples, we obtain a call fingerprint database. In a realistic scenario, we have to expect $N \geq 10^6$.

To identify an unknown fingerprint derived from a speech sample, the sample is first transformed to a feature document

$$Q = \{[t_1,\rho_1],\ldots,[t_k,\rho_k]\} =: \{q_1,\ldots,q_k\},$$

in a database setting called a *query*. By defining time-shifts $(t,\rho)+\tau := (t+\tau,\rho)$ of features and correspondingly of feature documents,

$$Q+\tau := \{[t,\rho]+\tau \mid [t,\rho] \in Q\},$$

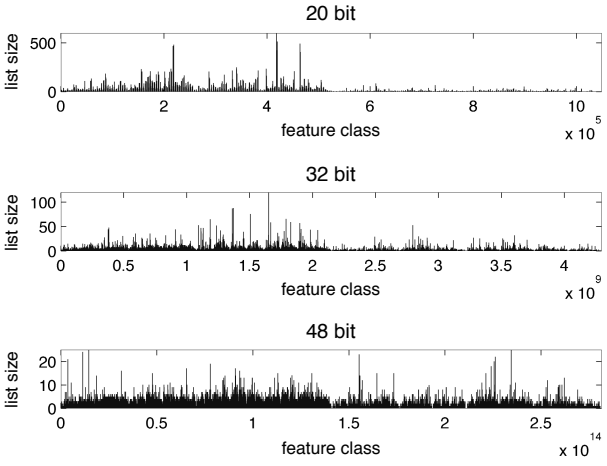

Figure 3: Distribution of $5200 \cdot 100$ real speech data sub-hashes at different bit lengths.

Q is contained in the i-th document of the call fingerprint database, if there exists a time-shift t such that $Q+t \subseteq D_i$. In this case, (i,t) is called an exact match. The set

$$H_{\mathscr{D}}(Q) := \{(i,t) \mid Q+t \subseteq D_i\}$$

of all exact matches can be obtained very efficiently by first observing that

$$H_{\mathscr{D}}(Q) = \bigcap_{q \in Q} H_{\mathscr{D}}(q)$$

where $H_{\mathscr{D}}(q) := H_{\mathscr{D}}(\{q\})$ is an *inverted list* containing all occurrences of the feature $q = [t,r]$ in the call fingerprint database. As $H_{\mathscr{D}}([t,r]) = H_{\mathscr{D}}(\{[0,r]\}) - t$ where subtraction is performed on the second component (time), it is sufficient to store a database index consisting of all inverted lists $H_{\mathscr{D}}([0,r])$.

It turns out that in order to reliably identify a given query, exact matches are generally not necessary. An n-match is a pair (i,t) such that

$$|(Q+t) \cap D_i| \geq n,$$

i.e. only n features of the query need to coincide with a particular position within the database. The set of all n-matches can be efficiently evaluated using a dynamic programming approach by iteratively defining score functions $S^j : \Gamma^j \to [1:j]$ with $S^1(\gamma) := 1$ for $\gamma \in \Gamma^1 := H_{\mathscr{D}}(q_1)$,

$$\Gamma^j := \Gamma^{j-1} \cup H_{\mathscr{D}}(q_j)\ ,$$

and

$$S^j(\gamma) := \begin{cases} S^{j-1}(\gamma)+1 & \text{if } \gamma \in \Gamma^{j-1} \cap H_{\mathscr{D}}(q_j), \\ S^{j-1}(\gamma) & \text{if } \gamma \in \Gamma^{j-1} \backslash H_{\mathscr{D}}(q_j), \\ 1 & \text{if } \gamma \in H_{\mathscr{D}}(q_j) \backslash \Gamma^{j-1}. \end{cases}$$

n-matches are elements $\gamma \in \Gamma^j$ with $S^j(\gamma) \geq n$.

Intuitively, a high number of different sub-fingerprints leads to more specific queries (and more, but shorter inverted files). In this case, we also expect to obtain reliable identification results when only considering n-matches for very small n. Moreover, the average number of 1-matches for a given number of $N = 10^6$ calls, an assumed length

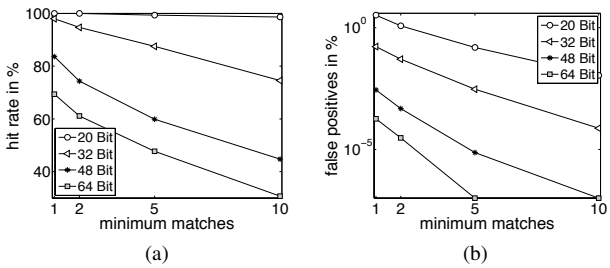

Figure 4: Hit rate (a) and false positives (b) at different bit lengths.

bit length	hit rate [%]	false positives [%]
20	99.89	1.20
32	94.67	0.05
48	74.33	$4.77 \cdot 10^{-4}$
64	61.11	$2.96 \cdot 10^{-5}$

Table 1: Hit rate and false positives for a 2-match search.

$m = 100$ of all feature documents and R different sub-fingerprints can be estimated as m^2N/R [10] (for a uniform distribution of features), resulting, e.g. in 9,537 1-matches for $R = 2^{20}$, reducing to only two 1-matches for $R = 2^{32}$. For a very large database with $N = 10^9$ calls, the average number of 1-matches would be 0.04 for $R = 2^{48}$, allowing for an efficient search also in that scenario. On the other hand, a higher number of different sub-fingerprints, i.e. higher bit lengths, leads to a lower robustness w.r.t. signal distortions.

5 Analysis

In order to test the efficiency of the hash to detect spam on the one hand and to avoid false matches of regular telephone calls on the other hand, a test corpus has been created. The test corpus is composed of material taken from two different sources. Regular phone calls have been gathered from the Verbmobil II corpus [11] of German telephone dialogs. Short utterances and sentences were concatenated with random pauses resulting in 5,000 audio files with a duration of around 10 seconds. The robocall part consists of 200 files and is based on 20 real telephone spam recordings. In order to test the robustness of the algorithm, the 20 spam files were intentionally altered by noise, audio- and telephone codecs.

More precisely, the following types of alterations and distortions were considered:

- mp3-codec 96 kbps and 32 kbps,
- GSM fullrate,
- G.726 codec 96 kbps and 32 kbps,
- 5 % and 10 % packet loss,
- white Noise, 20 dB SNR
- pink Noise, 20 dB SNR

5.1 Recognition Rate

In order to test the recognition rate, a search has been performed for each of the $N = 5{,}200$ files from the above corpus, resulting in $N(N-1)/2$ pairwise comparisons.

The bit length of the hash vectors, which corresponds to the number of feature classes, is varied in a range be-

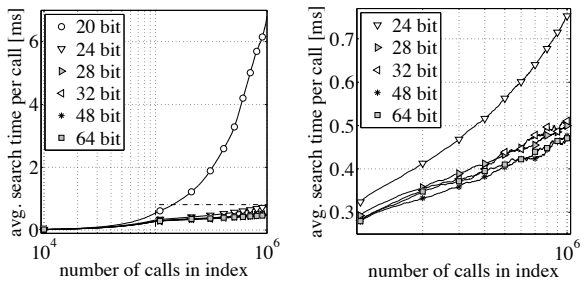

Figure 5: Search speed at different bit lengths. Overview (a) and detail view (b) for different numbers of indexed calls.

tween 20 bits and 64 bits. Different values for the minimum number n of matches are considered: $n = 1, 2, 5, 10$. A minimum match of only one sub-hash corresponds to a simple lookup. An n-match requires that at least n sub-hashes and their temporal offsets coincide. For increasing n, the number of false positive n-matches decreases.

The overall test results based on a 2-match search are shown in Fig. 4a and in Tab. 1. As expected, the hit rate decreases with an increasing number of feature classes. The hit rate of 1-matches decreases from 100 % at 20 bits to 69.4 % at 64 bits while on the other hand the false positives drop significantly from 3.3 % to 0.00018 %. 2-matches at 32 bits provide a good result with a 94.7 % hit rate and 0.05 % false positives. If lower hit rates are acceptable or the considered speech samples are of sufficient audio quality, 48 bits or even 64 bits are feasible. It has to be noted that the hit rate highly depends on the extent of the alterations but not on the source material itself. Furthermore, the same speaker or identical phrases do not show a significantly higher rate of false positives.

5.2 Search Speed

While the hit rate test is performed with a corpus of real speech data, the search speed test is done with randomly generated data. This second test addresses the influence of the number of feature classes on the search speed. The search speed is tested at 2^{20}, 2^{24}, 2^{28}, 2^{32}, 2^{48}, 2^{64} feature classes and a minimum requirement of a 2-match.

In contrast to the real speech corpus, the randomly generated data is uniformly distributed. If the hashes of the speech data were distributed as shown in Fig. 3, the mean search times would be slower by some constant factor. The search times in Fig. 5 are average values for one search in an index of one million calls with 100 sub-hashes each. It is obvious that the search times increase with a growing call database. As expected, the higher the number of feature classes the faster the search. Although the search speed does not improve with bit lengths of 48 or 64 bits in our scenario, it would allow an efficient identification with an even larger call database. Furthermore, it is expected that the non-uniform distribution of sub-hashes requires higher bit lengths for the same search efficiency.

6 Conclusion

The test results show that audio hashes are suitable to detect telephone spam, or more generally to match recorded speech. The bit length of the hash vectors (i.e. the number of feature classes) can be optimized to find a good balance between the recognition rate, the number of false positives and the search speed. Generally, higher bit lengths permit faster index-based searching but deteriorate the hit rate for similar speech data. However, for a fixed number of indexed hashes, no further speed-up is obtained above a certain bit length. The lowest recognition rate was observed for GSM-coded variants at higher bit lengths. When focusing on mildly degraded audio data, the hit rate should still be high enough even at 64 bits. In order to increase the recognition rate, it is also possible to extract a longer fingerprint (e.g. by using a longer time period of the audio file) at the cost of more false positives and higher search times. Nevertheless, in our current scenario a bit length of 32 permits a fast search with an acceptable hit rate.

Haitsma and Kalker [2] showed that their 32-bit spectral hash vectors are relatively robust against common alterations of music recordings. We investigated various bit-lengths, integrated mel frequency coefficients and described an efficient method to match sequences of hash vectors. This provides additional flexibility, robustness and search efficiency for the identification of speech signals in different fields of application.

Acknowledgments

This work has been conducted within the research project VIAT, which is supported by the German Federal Ministry of Education and Research (BMBF), reference 1736X09.

References

[1] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors [lecture notes]," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 128–131, 2008.

[2] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system.," in *ISMIR*, 2002.

[3] D. Lentzen, G. Grutzek, H. Knospe, and C. Pörschmann, "Content-based Detection and Prevention of Spam over IP Telephony - System Design, Prototype and First Results," *IEEE International Communications Conference (ICC) 2011*, June 2011.

[4] M. Cremer, B. Froba, O. Hellmuth, J. Herre, and E. Allamanche, "AudioID: Towards Content-Based Identification of Audio Material," in *Audio Engineering Society Convention 110*, May 2001.

[5] M. Clausen and F. Kurth, "A unified approach to content-based and fault-tolerant music recognition," *IEEE Transactions on Multimedia*, vol. 6, pp. 717–731, Oct. 2004.

[6] A. L.-C. Wang, "An Industrial-Strength Audio Search Algorithm," *ISMIR 2003, 4th Symposium Conference on Music Information Retrieval*, pp. 7–13, 2003.

[7] A. L.-C. Wang and J. O. Smith III, "Methods for recognizing unknown media samples using characteristics of known media samples," Mar. 2008.

[8] F. Kurth and M. Müller, "Efficient Index-Based Audio Matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 382–395, Feb. 2008.

[9] N. Chen, H.-D. Xiao, and W. Wan, "Audio hash function based on non-negative matrix factorisation of mel-frequency cepstral coefficients," *Information Security, IET*, vol. 5, pp. 19–25, Mar. 2011.

[10] K. Moravec and I. Cox, "A comparison of extended fingerprint hashing and locality sensitive hashing for binary audio fingerprints," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, p. 31, ACM, 2011.

[11] Bavarian Archive for Speech Signals, "Verbmobil II - VM2."