AUDIO SIGNATURE EXTRACTION BASED ON PROJECTIONS OF SPECTROGRAMS

Regunathan Radhakrishnan, Claus Bauer, Corey Cheng and Kent Terry

Dolby Laboratories Inc
100 Potrero Ave, San Francisco, CA
Email: {regu.r,cb,cnc,kbt}@dolby.com

ABSTRACT

Content-based signatures are designed to be a robust bitstream representation of the content so as to enable content identification even though the original content may go through various signal processing operations. In this paper, we propose a novel content-based audio signature extraction method that captures temporal evolution of the audio spectrum. The proposed method, first, divides the input audio into overlapping chunks and computes a spectrogram for each chunk. Then, it projects each of the spectrograms onto random basis vectors to create a signature that is a low-dimensional bitstream representation of the corresponding spectrogram. Our experimental results show the robustness and sensitivity of the proposed content-based audio signature extraction method for various signal processing operations on audio content.

1. INTRODUCTION

The Internet increasingly serves as a platform to publish and sell multimedia content. The piracy of multimedia content in the Internet creates a need for technologies that can identify copyrighted content. Content based signature extraction has been identified as a candidate technology. As pirates might publish multimedia content that has been processed in order to render a non-manual identification of pirated content more difficult, the extracted signature should largely remain the same as the content goes through various signal processing operations. Also, the signature should be compact for the efficiency and scalability of the identification algorithm that uses the signature.

Past work on content-based audio signature extraction generally consists of the following two steps: the first step extracts features that represent the underlying content and the second step generates a compact bitstream representation of the features. In [1], Burges et al propose a dimensionality reduction method called Distortion Discriminant analysis (DDA) to obtain a low-dimensional representation of the log spectrum of the audio signal. In [2], Chun-Shien Lu proposes a audio characterization method based on one dimensional wavelet transform coefficients. In [3], Batlle et al propose a signature for song identification from noisy broadcast audio based on MFCC feature extraction followed by Viterbi decoding from a sequence of pre-trained HMMs. In [4], Haitsma et al propose a robust audio signature based on the fact that the sign of energy differences (simultaneously along the time and frequency axes) is a property that is very robust to many kinds of processing.

Unlike prior approaches, our proposed method treats the whole spectrogram as a pattern and extracts signature bits to capture the temporal evolution of the audio spectrum. It thus avoids the additional step of extracting features from a spectrogram or spectrum and also can record more information about the spectrogram. It attempts to represent the underlying content as a sequence of spectrograms. The proposed method first divides the input audio into overlapping chunks and computes a spectrogram for each chunk. Then, it projects each of the spectrograms onto random basis vectors to create a signature that is a low-dimensional bitstream representation of the corresponding spectrogram. This method of projection onto random vectors was originally proposed for still image hashing in [5]. The concatenation of the sequence of extracted bits from the spectrogram of each chunk serves as the identifier for underlying content.

We present the proposed audio signature extraction method in section 2 and provide experimental results in section 3.

2. AUDIO SIGNATURE EXTRACTION

The audio signature extraction consists of the feature extraction and the calculation of the signature bits.

2.1. Audio Feature Extraction

The audio feature extraction attempts to represent the underlying content as a sequence of spectrograms. Note that the spectrogram itself captures the temporal evolution of the spectrum of the signal. Towards this end, we divide the input audio into overlapping chunks and create a spectrogram from each of the chunks. Then, we create a coarse spectrogram by averaging along both time and frequency. This operation provides robustness against small changes in the spectrogram along time and frequency. Note that the coarse spectrogram created could choose to emphasize certain parts of the spectrum more than others.

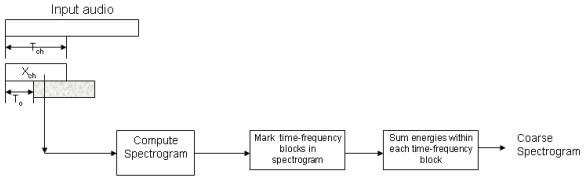

Fig. 1. Audio Feature Extraction

The generation of the coarse spectrogram itself is illustrated in detail in Figure 1. The input audio is first divided into chunks of duration T_{ch} with an overlap of T_o between adjacent chunks. For each chunk of audio data (X_{ch}) we compute a spectrogram with certain time resolution and frequency resolution. Then, we tile the computed spectrogram X_{ch} with time-frequency blocks. Finally, we sum up the magnitude of the spectrum within each of the time-frequency blocks to obtain a coarse representation of the spectrogram. Let us represent the spectrogram by S. We obtain a coarse representation (Q) of S by averaging the magnitude of frequency coefficients in time-frequency blocks of size $W_f \times W_t$. Here, W_f is the size of block along frequency and W_t is the size of block along time. Let F be the number of blocks along frequency axis and T be the number of blocks along time axis and hence Q is of size (F*T). Q is computed as given below:

$$Q(k,l) = \frac{1}{W_f * W_t} \sum_{i=(k-1)W_f}^{kW_f} \sum_{j=(l-1)W_t}^{lW_t} S(i,j)$$

$$k = 1, 2...F; l = 1, 2...T$$

Here, i and j represent the indices of frequency and time in the spectrogram and k and l represent the indices of the time-frequency blocks in which the averaging operation is performed.

Finally, we create a low-dimensional representation of a coarse spectrogram (Q) of the input audio frame by projecting the spectrogram onto random vectors, which can be thought of as basis vectors. The projection onto random vectors is explained in section 2.2.

2.2. Robust Hash

This block takes as input the matrix Q, and creates the signature by generating K hash bits. We use a robust hash function for this purpose as small perturbations in the audio features caused by signal processing operations such as compression, filtering, etc. would not change the hash bits drastically. The Robust hash function also serves to reduce the bit-rate of the signature stream. Let us represent the dimensions of the matrix Q by ($M \times N$). By using a robust hash, instead of sending ($M \times N$) values we only send few bits. A robust hash function is unlike a regular cryptographic hash function. A cryptographic hash function changes its output for every single bit change in the input. However, we would like our hash output to change slowly with small changes in features. This would enable us to allow for certain signal processing operations on the content which do not change the content but only slightly disturb the features. We use one such robust hash function proposed in [5] for generating the hash bits from the feature matrix, Q. We generate K random vectors each with the same dimensions as the matrix, Q ($M \times N$). The matrix entries are uniformly distributed random variables in $[0, 1]$. The state of the random number generator is set based on a key. Let us denote these random vectors by $P_1, P_2, ...P_K$ each of dimension ($M \times N$). We compute the mean of matrix P_i and subtract it from each matrix element in P_i (i goes from 1 to K). Then, the matrix Q is projected onto these K random vectors as shown below:

$$H_k = \sum_{i=1}^{M} \sum_{j=1}^{N} Q(i,j) * P_k(i,j) \quad (1)$$

Here H_k is the projection of the matrix Q onto the random vector P_k. Using the median of these projections (H_k, $k = 1, 2, ...K$) as a threshold, we generate K hash bits for the matrix Q. We generate a hash bit '1' for k^{th} hash bit if the projection H_k is greater than the threshold. Otherwise, we generate a hash bit of '0'.

3. EXPERIMENTAL RESULTS

3.1. Test Content

The test content used for the performance assessment of the proposed audio signature extraction consisted of 36 one-minute audio clips of a variety of content types in various formats. The audio content included stereo at a sampling rate of 48khz. Each of the test clips were processed in a variety of ways to simulate typical signal processing operations. We considered the following processing operations: Downmixing, Gain modification by +/-15db, Equalization, Dynamic Range Compression, Dolby Digital Encoding/Decoding (at 192 and 128kbps for stereo), MPEG1 layer II encoding and decoding at 128 and 256kbps for stereo. Overall, we created 645 test cases for audio.

3.2. Parameter Settings

The signature extraction parameters for audio were set as described below. The time-frequency resolution for the spectrogram was set to be 10×20 ($T = 10, F = 20$). The number of random vectors for the hash was chosen to be 18. The chunk size (T_{ch}) was chosen to be 90ms and the step size (T_o) was set to be 512 samples at 48khz sampling rate. The comparison of two audio signatures was based on the hamming distance. These parameters were chosen empirically based on performance on a wide variety of content.

3.3. Robustness of proposed audio signature

In this section, we present experimental results to show the robustness of the proposed audio signature to various signal

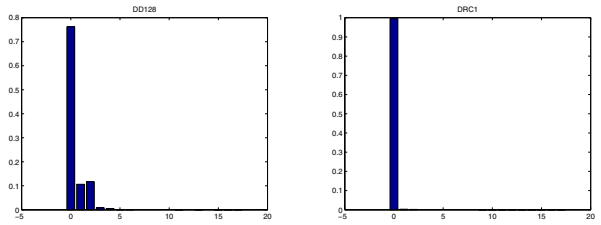

Fig. 2. Left:Robustness against compression Right: Robustness against DRC

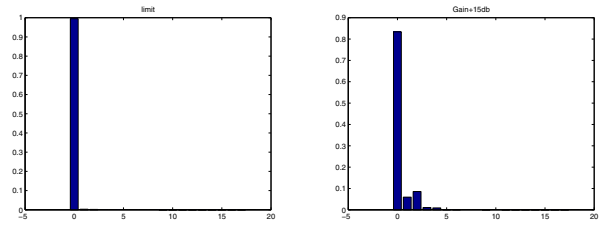

Fig. 3. Left:Robustness against Limiting Right: Robustness against Volume Change

processing operations on the original content. Towards that end, we first extract the proposed audio signature from original audio content and then compare that against the signature extracted from processed (attacked) audio content using hamming distance measure. Finally, we look at the histogram of hamming distance for each of the following signal processing operations. A robust signature would have a distribution that is heavily skewed near hamming distance value of zero.

3.3.1. Audio Compression

We performed the following five compression attacks on the original content: Dolby Digital Encoding and Decoding cycle at 192kbps (DD192) and at 128kbps (DD128) and MPEG1 Layer II Encoding and Decoding cycle at 256kbps (MP1LII256) and at 128kbps (MP1LII128) and Dolby E encoding and decoding (DE20). Figure 2 shows the distribution of hamming distances for DD128 compression attack. From the figure, it can be observed that 1 or 2 bit flips occur for almost 20% of the time with DD128 attack. Also, from Table 1, one can observe that MP1LII128 compression attack causes most number of bits to be flipped in the signature and DD192 causes the least number of bits to be flipped in the signature. One can also conclude that as one allocates more bits for compression, fewer signature bits change. Another interesting observation is that MPEG1 LayerII compression at 256kbps (MP1LII256) causes more bitflips in the signature than Dolby Digital Encoding at 192kbps.

3.3.2. Dynamic Range Compression

Dynamic range compression attacks were evaluated by applying multi-band compression methods to the original content (DRC1 and DRC2). These represented moderate to aggressive compression of dynamic range. Figure 2 shows the distribution of hamming distances for one of the dynamic range compression attacks (DRC1). The distribution has a weight of close to 1 for a hamming distance value of zero. Also, from table 1, observe that both DRC1 and DRC2 cause only 0.04% and 0.05% of the bits to be flipped.

3.3.3. Equalization

An equalization attack was tested by applying an equalization method (consisting of +/-6dB equalization processing across various frequency bands) to the source content. Figure 4 shows the distribution of hamming distances for the equalization attack. From the figure, one can observe that close 40% of the time there can 1 or 2 bit flips in the extracted signature if equalization is performed. Also, from table 1, observe that equalization causes 3.87% of the bits to be flipped which is one of the highest among the test cases. Since the spectrum is shaped during equalization, it causes some of the values in the coarse spectrogram to change. This consequently causes more percentage of the bits to be flipped.

3.3.4. Gain Change

A volume change attack was evaluated by applying a +/-15dB gain to the original content (Gain+ and Gain-). Figure 3 shows the distribution of hamming distances for volume change attacks. Also, from table 1, observe that positive gain change by 15db causes 1.7% of the bits to be flipped whereas negative gain change causes only 0.35% of the bits to be flipped. We observed that a volume change of +15db results in clipping of the sample values and which in turn can affect the spectrum and result in more bit changes in the signature than in the case of volume change by -15db.

3.3.5. Limit

We increased the gain on original audio by 6db first and then passed it through a limiter and finally reduce the gain by 6db again. Figure 3 shows the distribution of hamming distances for this processing (limit). As is shown in table 1, this processing causes only 0.04% of the bits to be flipped.

3.3.6. Addition of Noise

A noise attack was evaluated by adding a relatively high level of noise (-25dB) to the source content. Figure 4 shows the distribution of hamming distances for addition of noise attack. Also, from table 1, observe that addition of noise causes 4.74% of the bits to be flipped which is the highest among all attacks. The reason for this is the following. Recall that the

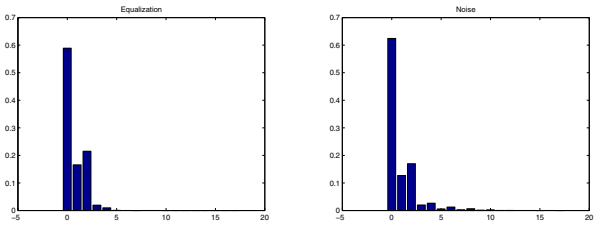

Fig. 4. Left:Robustness against equalization Right: Robustness against noise

Attack	B	C	D
DD128	53730	135950	0.0219
DD192	24692	135950	0.0100
DE20	23036	135924	0.0094
DRC1	1070	135990	0.0004
DRC2	1316	135990	0.0005
Equalization	90594	129756	0.0387
Gain-	8792	135990	0.0035
Gain+	37716	122694	0.0170
Limit	1069	135990	0.0004
MP1LII128	79314	135999	0.0324
MP1LII256	45945	135999	0.0187
Noise	116134	135978	0.0474

Table 1. Robustness of Proposed Audio Signature for various Signal Processing Operations; B: $Num.BitErrors$ C: $Num.Frames$ D:$BER = \frac{B}{(Sig*C)}$, $Sig = 18$

spectrogram was computed from 100ms chunks of audio. If the audio source contains lots of silence periods like speech pauses, the spectrograms in these regions change considerably when noise is added. This results in bit errors in those parts of the signal. Note, from Figure 4, that the hamming distance between the original and attacked signature can be as high as 10 for this reason. We are basically comparing the fingerprints of the noise signal to silent segments in the original which results in such a high value of hamming distance.

For the aforementioned attacks, we have seen the robustness of the proposed signature for various signal processing operations in terms of percentage number of bit flips for each kind of processing (see Table 1). Now, we would like to see how sensitive (unique) the proposed signature is to the underlying content it represents. Figure 5 shows the histogram of hamming distances between signatures of an audio (say clip A) and its various processed versions (clip A compressed and decompressed, clip A through an equalizer, clip A through dynamic range compression etc). Note that this histogram is skewed close to 0 showing the robustness to various processing. The figure also shows the histogram of hamming distances between signatures of two different audio files (say clip A and another clip B). This histogram is centered around

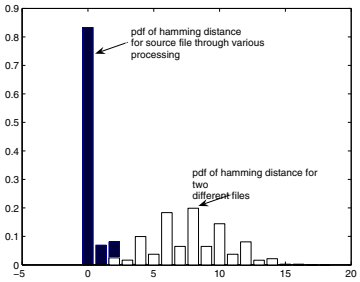

Fig. 5. Comparison of hamming distance histograms

the hamming distance value 8, and has very little overlap with the other. This means that the proposed signature is sensitive(unique) to the underlying content it represents and can serve as a robust audio content identifier.

4. CONCLUSION

We proposed a novel audio signature extraction method that is robust to various signal processing operations. The proposed method divides the input audio into overlapping chunks and creates a spectrogram from each chunk to capture the temporal evolution of the spectrum. Then, the extracted spectrograms are projected onto random basis vectors to generate signature bits. Thus, the underlying audio content is represented compactly as a sequence of spectrograms. Our experimental results show that common signal processing operations including compression, equalization etc cause only less than 5% of the signature bits to flip. We also show the sensitivity of the signature by comparing the signatures of two different files. In our future work, we will study the tradeoff between signature bit rate and robustness. We would also study the effect of time scale modification attacks on the signatures.

5. REFERENCES

[1] Burges, C.J.C. Platt, J.C. and Jana, S, "Distortion discriminant analysis for audio fingerprinting," *IEEE Transactions on Speech and Audio Processing*, May 2003.

[2] Chun-Shien Lu, "Audio fingerprinting based on analyzing time-frequency localization of signals," *Proc. of MMSP*, May 2002.

[3] E.Batlle, J.Masip and E. Guaus, "Automatic song identification in noisy broadcast audio," *Proc. of SIP*, Aug 2002.

[4] J.Haitsma and T.Kalker, "A highly robust audio fingerprinting system," *Proc. of ISMIR*, 2002.

[5] J.Fridrich and M.Goljan, "Robust hash functions for digital watermarking," *Proc. of ITCC*, May 2000.