Audio Engineering Society

Convention Paper

Presented at the 125th Convention
2008 October 2–5 San Francisco, CA, USA

EFFICIENT DETECTION OF EXACT REDUNDANCIES IN AUDIO SIGNALS

José R. Zapata G.[1] and Ricardo A. Garcia[2]

[1] Universidad Pontificia Bolivariana, Medellín, Antioquia, Colombia
Joser.zapata @ upb.edu.co

[2] Kurzweil Music Systems, Waltham, MA, 02451, USA
rago @ ycrdi.com

ABSTRACT

An efficient method to identify bitwise identical long-time redundant segments in audio signals is presented. It uses audio segmentation with simple time domain features to identify long term candidates for similar segments, and low level sample accurate metrics for the final matching. Applications in compression (lossy and lossless) of music signals (monophonic and multichannel) are discussed.

1. INTRODUCTION

Most of the compression systems are based on signal analysis using short time windows (2 to 50 ms) due to the computational cost associated to the handling of much larger windows. They also work with Inter-channel redundancy analysis (like joint stereo) to increase the compression ratio.

The main assumption is that there are multi channel audio files where the information is repeated exactly at long intervals. Most of the popular music (Ex: electronic music, Pop, Rap, Hip-hop, rock, among others) contain inter or intra channel repetitive elements [1] which may have durations up to 30 seconds [2] (E.g.: The chorus in a song). This is true for some types of music because, for example, the live music or recordings made before the digital age have little redundancy and is primarily of the psycho-acoustic kind, ex: appear similar to human perception.

This repetitive behavior stems from the fact that often, during the recording sessions, the copy – paste technique is commonly used in different audio segments (inter or intra channel) to achieve a final song (nowadays the artists records the chorus just once and then this is copied in the places where the song is repeated over). Another factor that must be considered is that each kind of music is different, which implies a specific analysis: the redundancy found in electronic music can be as twice the one found in jazz or classical music.

If you want to find the exact redundancy of two segments in an audio file, it could be done through brute force: by segmenting the audio signal, comparing each segment to the others, then increasing the size of the segments and do the complete process all over again to finally determine the length of the redundancy. As it shows, this is computationally expensive, that's why here is proposed an implementable method to detect and eliminate long-time redundancy in an audio file, such as pre-coding process looking for achieve an increased compression rate.

2. PREVIOUS APPROACHES AND PROPOSAL

Vishweshwara, Rao [3] presents a compression method based on the "musical structure" [4] of the song, comparing 4 different features of the audio signal: mfcc (mel-frequency cepstral coeffiecients), chroma, pitch and critical band scale rate. With these features and the similarity matrix [5], the similar audio segments are identified, then those who are significantly similar to others are eliminated downsizing the file and increasing the compression rate. Of the 4 features used to determine the similarity, the best results were presented by the critical band scale rate. It is important to clarify that this method only eliminates similar segments and does not determine their exact similarity. Besides, since these measures are in the frequency domain it implies a heavy computational cost.

This work describes the general operation and testing of an algorithm (ADROTA) that allows to locate the exact long-time redundancy in audio signals, with long duration repetitive patterns (T > 10 sec) and achieves an improvement the compression process (lossless or lossy).

3. LONG SEGMENT REPETITION DETECTION

As can be seen in Figure 1, the algorithm ADROTA basically works as follows: Divide the song in segments, the size of the segments is determined by a Beat tracking algorithm [6], then for each data segment the mean, median, variance, zero crossings, maximum and minimum feature is calculated. The similarity matrix for each feature is built using the calculated values.

This matrix allows the detection of the size and position of the repeated section, which is then eliminated. Afterwards, the algorithm proceeds to store the resulting audio file and the information needed to rebuild the original file again, which is the beginning of the repeated section, the length and the position where it is repeated.

3.1. Segmentation

Before starting to process the audio signal, it needs to be divided into segments. Each segment is processed individually. Although this may seem trivial is a very important step in the detection of similarity, where the segment's size plays a crucial role. If the segments size is too small it results in a very high time resolution. But unfortunately, it means an increase of the computational cost. If the segment's size is too long, the time resolution would be too small to extract accurate information. In previous investigations [6] it was determined that an optimal segment size at CD sampling rate of 44.1KHz would be between 50 milliseconds (2205 samples) to 100 milliseconds (4410 samples) long.

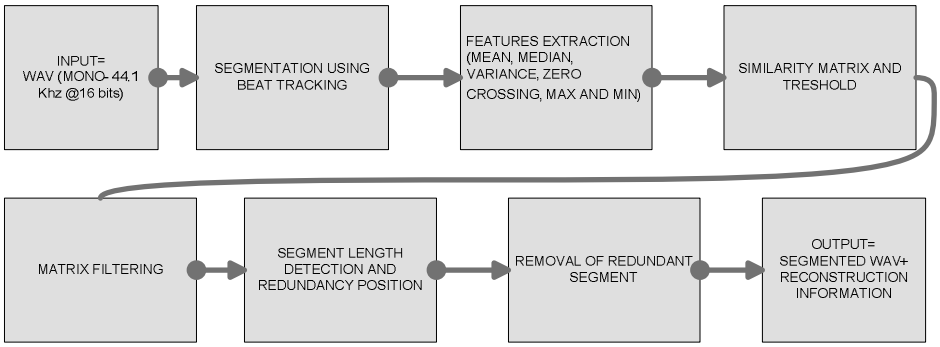

Figure 1 Basic algorithm function

An optional way to calculate the segment's size is suggested by Mildner, Bartsch and Wakefield [6], they proposed to use a segment's size which is a multiple of the beat or tempo of the song. Typically, the choruses and verses begin "in the beat", therefore the probability of detecting the similarity between various segments will increase.

To find the segment's size a Beat tracking system is used, this algorithm calculates the number of beats per minute (bpm) in a song. A very simple and robust method for extracting the song's beat, if the song doesn't contains strong or very marked drums beating, is developed by Scheirer [7]

3.2. Feature Extraction

Once the audio signal is divided into equal length segments, different features are calculated for all of them. As mentioned earlier, previous research have been focused on finding acoustical similar segments, using for this purpose different measures, such as the fundamental frequency, MFCC (mel-frequency cepstral coefficients), the Chroma, and the Critical Band Scale Rate, obtaining the best results whit this last one. To calculate these features requires an exceptional computing capability, in addition to the great amount of memory that is used in each of the processes. This paper shows a way to find exact repetitions (bit to bit) through the length of a song, using different features, considered more objectives and that fulfill the following conditions:

- The computational cost required to calculate the features must be low, to increase the detection's speed.
- The feature's data must be reduced, intending not to compromise the performance of the processor's memory while calculating them.

The features of each of the segments that were chosen for the tests were: Arithmetic Mean, Median, Variance, Number of zero crossings, Minimum value and Maximum value.

- Arithmetic Mean:

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N} \quad (1)$$

- Median:

$$Median = P(X \le m) \le 0.5 \le P(X \ge m) \quad (2)$$

- Variance:

$$S^2(x) = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}{n-1} \quad (3)$$

- Number of zero crossings:

$$Zerocross\sin gs = \sum_{i=0}^{N} I\{X_i * X_{i-1} < 0\} \quad (4)$$

$I\{A\}$ is equal to 1 if A is true or equal to 0 if A is false

- Minimum value:

$$X_{\min} = \min imum\{X_i\} \quad (5)$$

- Maximum value:

$$X_{\max} = \max imum\{X_i\} \quad (6)$$

3.3. Similarity Matrix

The similarity matrix proposed by Foote [8], is a method to visualize the music's structure by his acoustic likeness or unlikeness over time, rather than the absolute characteristics or events of the audio signal. As its name indicates, the purpose of the similarity matrix is to display the similarities between all the segments of the song, by comparing the calculated features of each segment by an Euclidian distance of their values.

If there are similar segments in the audio signals, they can be seen as parallel lines to the main diagonal of the matrix (figure 2). Unwanted data are also presented, to get just the data from the diagonals, the matrix is filtered with a diagonal Gaussian kernel (figure 3), emphasizing only the diagonals.

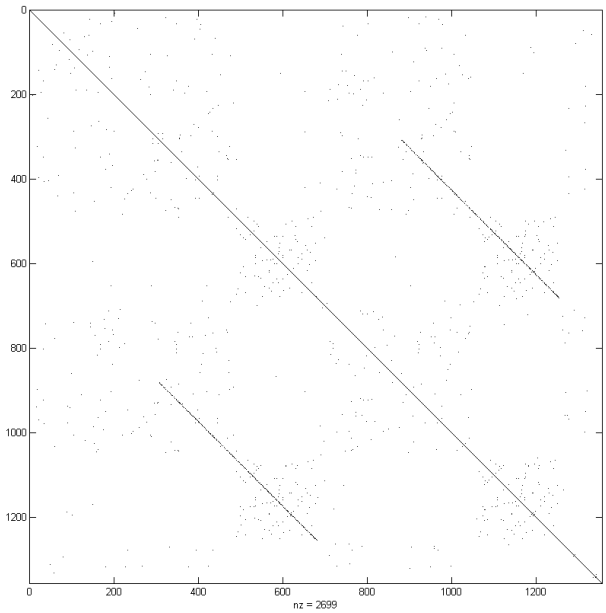

Figure 2 Similarity matrix example

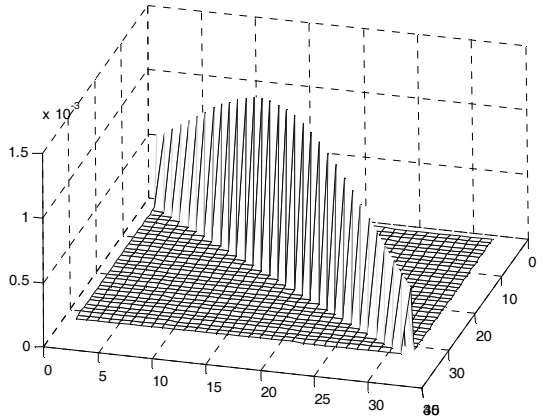

Figure 3 Diagonal Gaussian Kernel

3.4. Repetition detection

The diagonals are extracted from the filtered similarity matrix (the matrix is symmetric, figure 2), each one represents the presence of a redundant segment. The initial point of each diagonal shows approximate information on the beginning of the redundant data, and its length shows an approximation of the redundant segment's length.

To determine the exact start of the redundancy, a cross-correlation between the samples of the original and repeated segment is done, the length of the segment is obtained with the length of the diagonal. By making the correlation, an exact point where two segments are more similar is obtained (correlation = 1). Using this data, it's possible to find the exact position where the repeated segment is just like the original segment.

3.5. Repetition Removal

Following the detection of the repeated data in the audio signal, the next step is to remove it from the audio file and store the data to reconstruct the song later on. The stored data is: the time of the beginning and the end of the repeated segment, and the time where the segment is repeated in the song.

Finally the audio data is encoded by a lossless coder, reducing the size of the song to be stored.

4. RESULTS AND COMPARATION

The ADROTA algorithm is conformed by the detection and elimination of the long-time redundancies, and a basic lossless coder. The performance in compression of the algorithm was evaluated by a controlled series of test, to have more control over the experiment, 5 audio files were created based on actual songs, and modified (The audio files sounded musically correct, when people hear them, they don't notice the differences in the verse / verse union) to comply the following specifications: average length; a minute and a half, and minimum 10 seconds of exact redundancy. All of the audio signals are mono, sampled at 44.1 kHz @ 16-bit but the algorithm can be use to find inter channel redundancy in multi-channel audio files.

The comparison measure of compression was the space savings. A greater number indicates a higher degree of compression.

$$R\% = (1 - \frac{\text{Final\_size}}{\text{Original\_size}}) x 100\% \qquad (7)$$

	Original (Mbytes)	R% Space savings									
		Monkey's Audio .APE		OptimFrog .ofr		LA .la		TTA .TTA		FLAC .FLAC	
		Mbytes	R%	Mbytes	R%	Mbytes	R%	Mbytes	R%	Mbytes	R%
File 1	9,57	5,94	37,93%	6,03	36,99%	6,17	35,53%	6,21	35,11%	6,34	33,75%
File 2	10,295	6,36	38,22%	6,46	37,25%	6,6	35,89%	6,7	34,92%	6,83	33,66%
File 3	10,721	7,14	33,40%	7,24	32,47%	7,39	31,07%	7,53	29,76%	7,64	28,74%
File 4	11,67	7,755	33,55%	7,671	34,27%	7,548	35,32%	7,856	32,68%	8,1	30,59%
File 5	9,575	6,229	34,95%	6,215	35,09%	6,1	36,29%	6,336	33,83%	6,771	29,28%

Table 1 Compression results of Monkeys audio, Optimfrog, LA, TTA and FLAC

	Original (Mbytes)	R% Space savings									
		SHORTEN .SHT		ADROTA .JRZ		ADROTA + FLAC .FLAC		WINRAR .RAR		WINZIP .ZIP	
		Mbytes	R%	Mbytes	R%	Mbytes	R%	Mbytes	R%	Mbytes	R%
File 1	9,57	6,63	30,72%	4,626	51,66%	4,449	53,51%	6,71	29,89%	9,04	5,54%
File 2	10,295	7,16	30,45%	5,543	46,16%	5,307	48,45%	7,218	29,89%	9,766	5,14%
File 3	10,721	7,87	26,59%	6,3	41,24%	6,029	43,76%	7,934	26,00%	10,4	2,99%
File 4	11,67	8,371	28,27%	7,423	36,39%	7,15	38,73%	8,43	27,76%	10,85	7,03%
File 5	9,575	6,714	29,88%	4,643	51,51%	4,5	53,00%	6,82	28,77%	9	6,01%

Table 2 Compression results of Shorten, ADROTA, ADROTA + FLAC, Winrar and Winzip

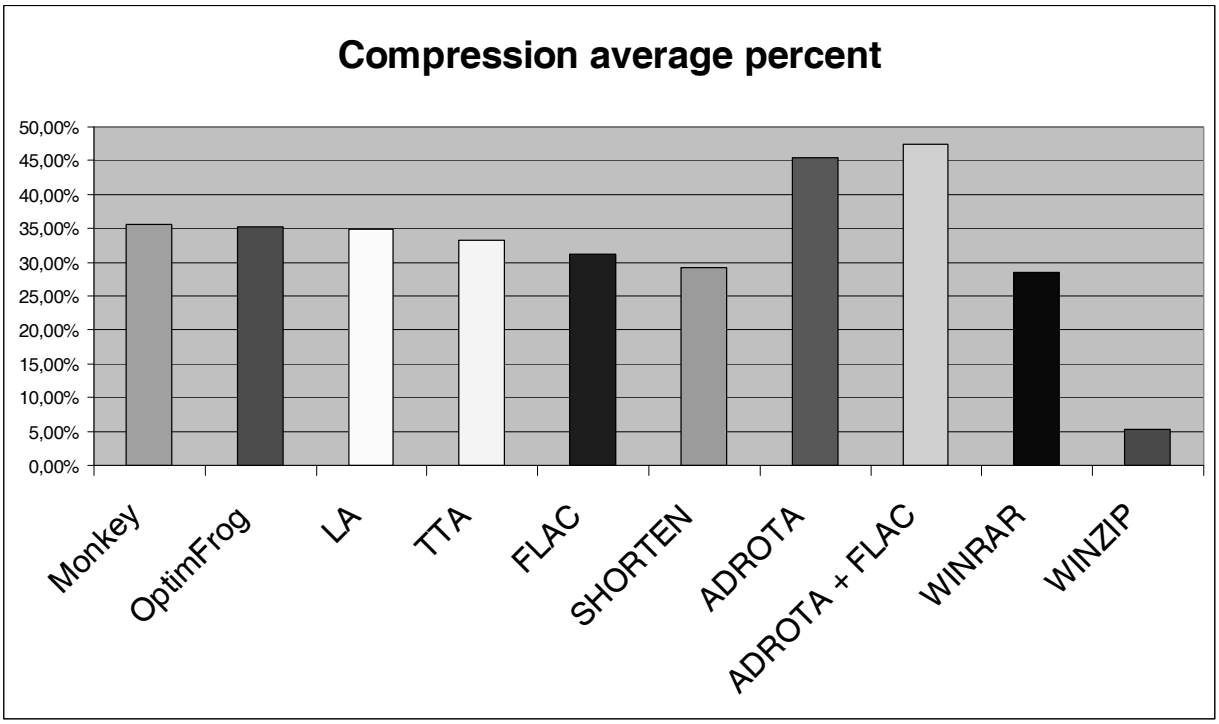

Figure 4 Space savings average

5. DISCUSSION

The results of controlled testing show an average improvement of 9.78% in the compression, with higher encoding times due to the fact that the implementation was done in Matlab, using only ADROTA (detection and compression). Using ADROTA for the redundancy detection and elimination, and FLAC compression, the average compression improves to 11.88%. Those percentages were obtained comparing with the following encoders: FLAC, TTA, OptimFROG, LA, and SHORTEN Monkey's Audio.

The data compression encoder winrar, without being specifically for audio signals, it generates a good compression, which is very close to that achieved with SHORTEN, the first known lossless audio encoder. Given this results, it is desirable to analyze the winrar encoder's operation to improve its performance in the lossless audio compression.

The exact redundancy analysis through the features of variance, median, arithmetic mean and number zero crossings are not recommended because the difference between segments is not very noticeable.

Future studies can analyze the exact redundancy in shorter time intervals, between 250 ms and 10 sec, and adjacent redundancies in the audio file, because in the similarity matrix is not possible to identify precisely when these events occur.

The search algorithm can be improve by analyzing consecutive redundancies of approximately 1 second, for example the ones that can be found in the so-called electronic music.

6. CONCLUSIONS

Comparing the achieved performance using the different features of the signal, the maximum and minimum value of the segment, showed the best results. Visually, with the other measures (mean, median, variance and crosses by zero) similarities can be observed, but the range of similarity is small, and makes it difficult to discriminate when two segments are equal.

The maximum and minimum values of a segment allow to find exact repetitions in audio signals, fast and with low computational cost, but they are not good measures for determining how similar segments are, frequency features are recommended, as seen analyzing previous research.

The long time redundancies search and elimination process is independent and transparent to the subsequent codification, making it possible to be implemented as a pre-process for either lossless or lossy audio encodings.

According to the evidence obtained in the current investigations, it was determined that the lossless encoders do not take into account the long time redundancies. Using ADROTA detection, the compression ratio can be increased.

7. ACKNOWLEDGEMENTS

This work was supported by Universidad Pontificia Bolivaria and GIDATI.

Thanks to Ricardo Garcia and Roberto Hincapié for their good recommendations, Maria C Juanita for her friendship and support. And my family for their love.

8. REFERENCES

[1] AUCOUTURIER, Jean-Julien. Finding repeating patterns in acoustic musical signals: applications for audio thumbnailing, AES 22 International Conference on Virtual, Synthetic and Entertainment Audio, 2002.

[2] CHAI Wei, Barry Vercoe, Structural analysis of musical signals for indexing and thumbnailing, Proceedings of ACM/IEEE Joint Conference on Digital Libraries, 2003.

[3] RAO, Vishweshwara. Audio compression using repetitive structures in music, florida, EEUU. Mayo 2004.

[4] DANNENBERG R. & Hu N., "Pattern discovery techniques for music audio" ISMIR Conference proceedings: Third international conference on music information retrieval, pp. 63-70, 2002.

[5] COOPER M. & Foote J., "Summarizing popular music via structural similarity analysis", IEEE workshop on applications of signal processing to audio and acoustics, New Paltz, New York, pp. 127-130, Oct 19-22, 2003.

[6] MILDNER V., Klenner P. & Kammeyer K., "Chorus detection in songs of pop music",Elektronische Sprachsignalverabeitung (ESSV 2003), Karlsruhe, Germany, September, 24th-26th 2003. Poli, A. Picialli, S. T. Pope, and C.

Roads Eds. Swets & Zeitlinger, Lisse, Switzerland, 1996.

[7] SCHEIRER E., “Tempo and beat analysis of acoustic musical signals.” Journal of the Acoustical Society of America, vol.103, no. 1, pp. 588-601, 1998.

[8] FOOTE J., "Automatic audio segmentation using a measure of audio novelty." Proc. Of IEEE International Conference on Multimedia and Expo, vol. I, pp. 452-455, 2000.