2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Self-Synchronous Time-Domain Audio Watermarking Based on Coded-Watermarks

Raúl Martínez-Noriega
Grad. School of Electro-Comms.
University of Electro-Comms.
Tokyo, Japan
raul@ice.uec.ac.jp

Mariko Nakano
Research Sec. and Grad. School
National Polytechnic Inst.
Mexico City, Mexico
mnakano@ipn.mx

Kazuhiko Yamaguchi
Grad. School of Informatics and Eng.
University of Electro-Comms.
Tokyo, Japan
yama@ice.uec.ac.jp

***Abstract*—A time-domain audio watermarking algorithm based on low-frequency amplitude modification was proposed by Lie in 2006. The algorithm has acceptable robustness against common attacks and it also achieves synchronization. However Lie's method has two disadvantages: the coded watermark was produced by convolutional codes which are not the best option for very noisy channels like watermarking ones and also, the payload is decremented because the synchronization code. We propose an improvement to Lie's algorithm which solves the problems abovementioned. Our improvement replaces both synchronization code and convolutional code for only a self-synchronous decoding algorithm which uses low-density parity-check (LDPC) codes. With this modification, higher payload is obtained because synchronization codes are avoided, and the watermark is more robust due to the LDPC code. The self-synchronous decoding is based on the good cyclically permutable (CP) characteristics of LDPC codes. Our improvement is capable to kept in sync with combined attacks, e.g. MP3 and clipping attack or low-pass filtering and clipping attack.**

***Keywords*-synchronization; audio-watermarking; time-domain; LDPC codes**

I. Introduction

Copyright protection is one of the main concerns for multimedia companies. Potential solutions have been proposed and some of them even implemented in real life but with a discreet success. Copyright protection is a demanding task because the protection must be unbreakable, low-complex and it should produce small distortion in the host signal.

Watermarking has been investigated as potential solution to copyright protection of digital media [1]. Research on watermarking for images is enough developed to producing commercial services. However, its counterpart for audio is still under development. The applications of audio watermarking go further than copyright protection. Extension of narrow bands, data-base indexing and advertising, just to mention some of them, are potential applications of watermarking.

Generally, watermarking in frequency domain produces robust watermarks. Nevertheless two problems arise, desynchronization and high computational cost. Desynchronization is because the host media is clipped in time-domain and therefore is difficult to recognize the clipped part in frequency. On the other hand, even nowadays there are algorithms like the fast Fourier transform, there is not lower complexity than the native domain, e.g. time-domain.

Proposals for audio watermarking in time-domain have been described in [2] and [3]. The former is a proposal for digital instruments, e.g. synthesizers, and even the technique shows interesting results is not suitable for a wider kind of music. The latter is a proposal from Lie *et ál.* which produces robust watermarks against the most common manipulations or attacks, e.g. MPEG-1 Layer 3 (MP3) and low-pass filtering. Due to the algorithm was developed for time-domain, it has low computational complexity which could be good enough for practical applications. Lie's algorithm is robust to clipping attack by using synchronization codes and also, the watermark was protected with convolutional coding to prevent errors due to the noise.

However, there are two disadvantages in Lie's method: the convolutional code and the synchronization code. Watermarking channels tend to obtain high bit error rate (BER) which means than powerful coding techniques are desirable. Common techniques like BCH codes and convolutional codes practically do not improve the watermark robustness [4]. The use of synchronization codes reduces the payload because the synchronization code does not include any information about the watermark at all.

Low-density parity-check (LDPC) codes are classified like forward error correction techniques. LDPC were discovered in the early 60's [5] but they were almost forgotten until 90's because the early decoding algorithms were complex. These codes use iterative decoding which is more powerful than conventional one. LDPC's performance increases exponentially to the code length but, unlike other codes, its computation cost increases only linearly to the code length.

We have introduced a self-synchronous decoding algorithm using LDPC codes in [6] and [7]. This algorithm does not need extra-bits, e.g. synchronization codes, to keep the synchrony between the encoder and the decoder. The synchronization capability is due to the good cyclically permutable (CP) characteristics of the LDPC codes. The synchronized codeword is searched in a region which is as large as almost twice the code length. The algorithm finds the synchronized codeword after applying only one iteration of sum-product [8] algorithm on each possible sequence.

978-0-7695-4222-5/10 $26.00 © 2010 IEEE
DOI 10.1109/IIHMSP.2010.41

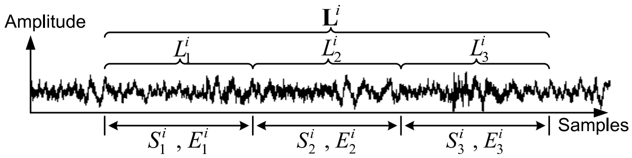

Figure 1. GOS^i is composed of three different sections S_1^i, S_2^i and S_3^i.

In this paper, we propose an improvement to Lie's method [3] using our self-synchronous algorithm [6]. The improvement produces watermarks which are more robust. The payload is increased because synchronization codes are not needed. The experimental results show that the watermark can be correctly retrieved after combination of attacks, e.g. MP3 compression and clipping attack.

II. Watermarking Scheme

Time-domain audio watermarking can be categorized in: algorithms that modify independently each audio sample and those which modify the audio in groups of samples. The former obtains high embedding rates, however the algorithms are not robust against intentional attacks. The latter, besides producing robust watermarks, controls better the distortion in the audio file.

In this Section, the algorithm proposed by Lie *et ál.* [3] is explained. The algorithm scales the audio amplitudes in a grouping and consistent manner while embedding the watermark information. Hence, the audio envelope is almost preserved and good audio quality can be achieved. Before the embedding, the watermark is encoded with a half-rate convolutional code and, the synchrony is kept with synchronization codes which are concatenated at the beginning of a certain block of bits.

Definition 1 (GOS): Group of samples (GOS) is defined as an audio segment, in time, of consecutive $\mathbf{L}$ samples. Each GOS^i is composed of three sections, S_1^i, S_2^i and S_3^i, of length $\mathbf{L}^i = L_1^i + L_2^i + L_3^i$ respectively, as shown in Fig. 1.

The audio file is divided into consecutive and non-overlapped GOS. Each bit will be embedded in a different GOS by modifying the average of absolute amplitudes (AOAA) of sections S_1, S_2 and S_3. The AOAA E_1^i, E_2^i, E_3^i of the GOS^i is computed as:

$$E_1^i = \frac{1}{L_1^i} \sum_{j=0}^{L_1^i - 1} |x_{(\mathbf{L}\cdot i + j)}|,$$

$$E_2^i = \frac{1}{L_2^i} \sum_{j=L_1^i}^{L_1^i + L_2^i - 1} |x_{(\mathbf{L}\cdot i + j)}|,$$

$$E_3^i = \frac{1}{L_3^i} \sum_{j=L_1^i + L_2^i}^{\mathbf{L}^i - 1} |x_{(\mathbf{L}\cdot i + j)}|.$$

E_1, E_2, E_3 are sorted and classified in $E_{min}, E_{mid}, E_{max}$ respectively. The differences, A and B, are computed:

$$A = E_{max} - E_{mid}, \quad (1)$$

$$B = E_{mid} - E_{min}. \quad (2)$$

For a certain GOS^i, the watermark bit "1" will be represented when $A^i \geq B^i$ otherwise that GOS^i contains a "0".

The embedding is conveyed according to the next rules:

- Watermark bit "1": **If** $(A - B \geq \text{Thd})$, then no operation is performed.
  Else, increase E_{max} and decrease E_{mid} by the same amount $\delta_{(1)}$ so that the above condition is satisfied.
- Watermark bit "0": **If** $(B - A \geq \text{Thd})$, then no operation is performed.
  Else, increase E_{mid} and decrease E_{min} by the same amount $\delta_{(0)}$ so that the above condition is satisfied.

The threshold Thd is computed with:

$$\text{Thd} = (E_{max} + 2E_{mid} + E_{min}) \cdot d,$$

where d is a parameter that controls the robustness-distortion trade-off. Thus, δ is computed depending on the bit to be embedded:

$$\delta_{(1)} = \frac{\text{Thd} - (A - B)}{3},$$

$$\delta_{(0)} = \frac{\text{Thd} - (B - A)}{3}.$$

Finally, E_{min}, E_{mid}, E_{max} can be modified by scaling (up or down) sample amplitudes in the corresponding GOS; that is, the watermarked audio will be $\bar{\mathbf{x}} = w \cdot \mathbf{x}$, where $\mathbf{x} = x_1, x_2, ... \in S_j$, $j = 1, 2$ or 3 and w is computed with $w_{up} = 1 + (\delta/E)$ and otherwise, to decrease we have $w_{down} = 1 - (\delta/E)$.

Initially d is set to $d = 0.05$ for each GOS^i, however not all the GOS has the same masking characteristics and therefore the watermark can be audible in some GOS. Thus, d has to be independently tunned for each GOS^i according to the psychoacoustic model. If the distortion with the initial conditions is high, then d is decreased by $\Delta d = .01$ and the whole embedding process is repeated until the audio quality constraint is satisfied. For more details refer to [3].

The watermark extraction is done by computing A' and B' from the watermarked audio, in the same fashion as in the embedding phase using (1) and (2). Then, if $A' \geq B'$ the bit "1" is retrieved otherwise "0" is recovered.

A. Coded Watermark

The watermark is protected with a half-rate convolutional code (2,1,3) which takes one input bit and produces 2 output bits. The output bits depend on the last 3 input bits. At the decoder, Viterbi algorithm is used to recover the watermark.

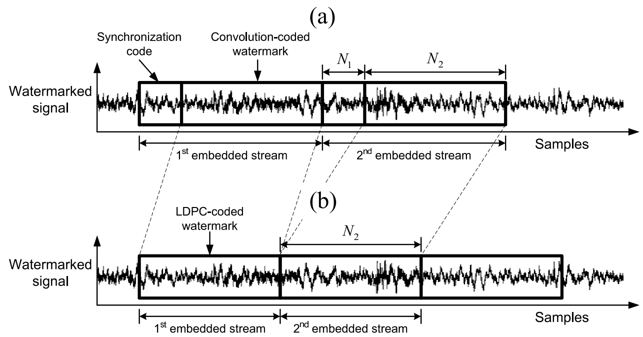

Figure 2. Embedding: (a) Lie's method, concatenation of convolution-coded watermark and synchronization code. (b) Our proposal, LDPC-coded watermark with self-synchronization capabilities.

B. Synchronization

Lie's algorithm is also robust to desynchronization produced by clipping attack. The solution is to concatenate a synchronization code at the beginning of each bit stream as shown in Fig. 2a. The synchronization code is a special vector of 20 bits. This vector is unique and it does not appear inside the coded watermark stream.

The decoder searches for the synchronization code in an area of $\mathbf{L} \cdot (N_1 + N_2)$ samples, where N_1 and N_2 are the numbers of bits in the synchronization code and the convolutional-coded watermark respectively. The decoder takes a window with the first $\mathbf{L} \cdot N_1$ samples and retrieves the embedded bits, those bits are compared with the synchronization code pattern. If they do not match, the window is shifted one sample to the right and the embedded bits are retrieved again until match with the synchronization code.

III. Self-synchronous Audio Watermarking

Lie's method, described in [3] and summarized in Sec. II, has low-complexity and acceptable robustness e.g. against MP3 at 128 kbps. However, it has two main disadvantages: first, the synchronization code reduces the payload and second, watermarking channels tends to be very noisy and therefore convolutional codes are not the best selection.

We introduced a novel self-synchronous decoding algorithm for common transmission in [6] and [7]. In this Section, we present an improvement of Lie's method by developing a practical implementation of our self-synchronous algorithm. The improved algorithm obtains higher payload than the original because it does not need synchronization codes. The robustness is also increased due to LDPC codes are more powerful codes than convolutional codes.

A. Watermark Embedding

The binary watermark $\mathbf{w} = w_1, w_2, \ldots$ is encoded with a LDPC code. The coded watermark $\bar{\mathbf{w}} = \bar{w}_1, \bar{w}_2, \ldots$ is embedded in the audio file $\mathbf{x} = x_1, x_2, \ldots$ following the procedure of Sec. II. However, Sec. II-A and Sec. II-B are omitted; that is, the convolutional code is not used and the synchronization code is not concatenated either. Fig. 2b pictures this idea.

B. Watermark Detection

The watermarked audio $\bar{\mathbf{x}}$ is susceptible to suffer distortions due to attacks. Therefore $\mathbf{y} = \bar{\mathbf{x}} + \mathbf{v}$ is defined as the watermarked audio with attacks or noise $\mathbf{v}$.

When the decoder receives $\mathbf{y}$, it does not have any knowledge about where the embedded watermark begins. Therefore, synchronization is needed. Let us assume that every coded watermark bit $\bar{w}_i$ was embedded in a GOS^i. The code length of the LDPC code is N_2. Also assume that the coded watermark was embedded continuously, that is one codeword after another. Therefore the synchronized codeword $\tilde{C}^{syn} = c_1, c_2, \ldots, c_n$ exists inside a segment $\mathbf{P}$ of $(2N_2 - 1) \cdot \mathbf{L}$ samples.

The audio segment $\mathbf{P}$ is divided into overlapped blocks $\mathbf{b}_j$ of $\mathbf{L}N_2$ samples. Every block $\mathbf{b}_j$ has an overlap of $\mathbf{L}N_2 - 5$ with the previous one, that is the difference between the blocks $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$ is only 5 samples shifted to the right. From each block $\mathbf{b}_j$, N_2 bits are retrieved using the extraction algorithm of Sec. II. Those bits represent a potential codeword $\tilde{C}^j$. Also, for each block $\mathbf{b}_j$ the log-likelihood ratio is computed with:

$$\mathbf{llr}_j = \frac{2\mathbf{y}_j \mu}{\sigma^2}, \tag{3}$$

where $\mathbf{y}_j = A' - B'$ is a vector with soft-information from the block $\mathbf{b}_j$, σ^2 is the noise variance and μ is the mean of $|A - B|$.

Each $\mathbf{llr}_j$ is decoded with only one iteration of sum-product algorithm, thus, an updated codeword $\hat{C}^j$ and its syndrome Z^j are obtained. The metric $m_j = d_H(\tilde{C}^j, \hat{C}^j) + Z^j$ is used to decide whether a codeword $\tilde{C}^j$ is synchronized, where $d_H(\tilde{C}^j, \hat{C}^j)$ is the hamming distance between $\tilde{C}^j$ and $\hat{C}^j$. Therefore the codeword $\tilde{C}^j$ that minimize the metric m_j is selected as the synchronized codeword $\tilde{C}^{syn}$, and it is decoded using full-iterative sum-product decoding.

This decoding algorithm works because LDPC are codes with good CP-characteristics. Let us consider any two codewords $\{C^i, C^j\} \in \mathscr{C}$, where $\mathscr{C}$ is the codebook. In a code with good CP-characteristics there is not codeword or just a very few codewords C^i that can turn into C^j by cyclically permuting C^i. That is, in principal $\mathscr{C}$ does not include cyclic permutations of C^i.

Inside the audio segment $\mathbf{P}$, all the potential codewords $\tilde{C}^j$ are cyclic equivalent codewords of $\tilde{C}^{syn}$ or they have small Hamming distance with a cyclic permutation of $\tilde{C}^{syn}$. Therefore, it is very likely that from all $\tilde{C}^j$ only the synchronized codeword $\tilde{C}^{syn}$ exists in $\mathscr{C}$ which means that the Hamming distance $d_H(\tilde{C}^j, \hat{C}^j)$ for $j = syn$ and its syndrome have the lowest value among all $\tilde{C}^j$.

Table I
BER RESULTS AGAINST MP3 COMPRESSION AND LOW-PASS FILTERING.

Attacks		Lie's method [3] (BER)		Proposed method (BER)
		uncoded	*coded*	*coded*
MP3	160 kbps	0.0006	0.0000	0.0000
	128 kbps	0.0104	0.0001	0.0000
	64 kbps	0.0945	0.0259	0.0000
Low-pass	10 kHz	0.0000	0.0000	0.0000
	4 kHz	0.0280	0.0007	0.0000

IV. RESULTS AND COMPARISONS

The watermark was embedded in random segments of 30 seconds taken from three different audio files: *Egmont Op. 84* (7 min.), *Billie Jean* by M. Jackson (4 min.) and *A change of seasons* by Dream Theater (21 min.). Those audio files are mono in WAVE format, sampled at 44.1 kHz/16 bits. The LDPC code is a random half-rate code with code length 96.

The parameters that control the embedding phase were set to the same value used in [3]. $L_1 = L_2 = L_3 = 340$, d was initially set to be 0.05 and the audio quality was controlled with the psychoacoustic model in silence using a quality of 85%.

Table I shows the BER for Lie's method [3] and our algorithm after MP3 and low-pass filtering. In general, Lie's method is robust against several audio manipulations however with strong attacks, e.g. MP3 at 64 kbps, the BER is not low enough. Our contribution to Lie's method produces more robust watermarks that does not need synchronization codes. The improvement in robustness might be small but the payload is increased as well. In [3] is not mentioned the length of the convolutional code N_2, therefore is difficult to quantify the payload benefit. However, in Fig. 2 is clear that our method has a benefit of 20 bits per each block of embedded information.

The results of our method, in Table I, were a combination of two attacks: MP3 or low-pass filtering with clipping attack. The number of continuous clipped samples were chosen randomly in each iteration from an uniform distribution between $[1, \mathbf{L}N_2]$. In case of Lie's method, we assumed that the system was in perfect sync and only attacks like compression or low-pass filtering were implemented.

It is expected that certain audio segments have stronger watermarks because the watermark's strength depends on the relation of the audio and the psychoacoustic model. Therefore, the watermark reliability varies depending on the audio file. That explains why different BER can be obtained with the same method but different audio files. In our comparison both methods, Lie's and ours, were tested with the same audio files.

Due to the good error correction capability of LDPC codes is not necessary to search the watermark in every audio sample. Searching the watermark every 5 samples is enough to decode the correct information, this explains the length of the overlap in Sec. III-B. Clearly, if the search is done every 5 samples instead of every sample, the complexity is reduced. However, perfect alignment can not be produced. In our opinion, perfect alignment is not important if the embedded watermark is retrieved correctly.

LDPC codes increase its performance with larger code length. Therefore, if larger LDPC codes were used the robustness would be increased. Small codes were used in this proposal because computational complexity restrictions. However, if smarter searches were implemented then better LDPC codes could be used.

V. CONCLUSIONS

In this paper we proposed a modification to Lie's audio watermarking method in time-domain. The modification produces a more robust algorithm because LDPC codes are used instead of convolutional codes. The payload is also increased because synchronization codes are avoided but the synchrony is still kept, even after combination of attacks.

Our self-synchronous decoding algorithm is based on the good CP-characteristics of LDPC codes. The algorithm is not capable to produces prefect alignment, however the watermark is perfectly recover. This misalignment is less than 10 audio samples, the same as in Lie's algorithm.

The synchronized codeword is searched in audio segments with size depending on the code length. Therefore small LDPC codes were used in order to keep the computational complexity low. However if the code length is increased our proposal becomes more robust. As future work, we are planning to implement smart searches which allow us to use larger codes with acceptable complexity.

REFERENCES

[1] J.-S. Pan, H.-C. Huang, L. C. Jain, and W.-C. Fang, *Intelligent Multimedia Data Hiding: New Directions*. Springer Publishing Company, Incorporated, 2007.

[2] K. Yamamoto and M. Iwakiri, "Real-time audio watermarking based on characteristics of PCM in digital instrument," *Inf. Hiding and Multimedia Sig. Process. (JIH-MSP)*, vol. 1, no. 2, pp. 59–71, 2010.

[3] W. N. Lie and L. C. Chang, "Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modificacion," *IEEE Trans. on Multimedia*, vol. 8, no. 1, pp. 46–59, 2006.

[4] L. Gu, Y. Fang, and J. Huang, "Revaluation of error correcting coding in watermarking channel*," *Lecture Notes in Computer Science*, vol. 3810, pp. 274–287, 2005.

[5] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Inf. Theory*, vol. 8, no. 10, pp. 21–28, 1962.

[6] R. Martínez-Noriega, I. Abe, and K. Yamaguchi, "Self-synchronizable decoding algorithms for transmission with redundant information," *IEICE Trans. on Fundamental on Electronics, Comms and Comp. Sciences.*, 2010.

[7] R. Martínez-Noriega, K. Yamaguchi, and K. Kobayashi, "Recovering synchronization with iterative decoders: LDPC codes," *Proc. of Int. Symp. on Inf. Theory and its Apps.*, 2010.

[8] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399–431, 1999.