High capacity FFT-based audio watermarking

Mehdi Fallahpour and David Megías

Estudis d'Informàtica, Multimèdia i Telecomunicació
Internet Interdisciplinary Institute (IN3)
Universitat Oberta de Catalunya- Rambla del Poblenou, 156
08018 Barcelona, Spain
Tel: (+34) 933 263 600, Fax: (+34) 933 568 822
E-mail: {mfallahpour,dmegias}@uoc.edu

Abstract. This paper proposes a novel high capacity audio watermarking algorithm to embed data and extract it in a bit-exact manner based on changing the magnitudes of the FFT spectrum. The key idea is to divide the FFT spectrum into short frames and change the magnitude value of the FFT samples based on the average of the samples of each frame. Using the average of FFT magnitudes makes it possible to improve the robustness, since the average is more stable against changes compared with single samples. In addition to good capacity, transparency and robustness, this scheme has three parameters which facilitate the regulation of these properties. The experimental results show that the method has a high capacity (0.5 to 4 kbps), without significant perceptual distortion (ODG is about –1) and provides robustness against common audio signal processing such as added noise, filtering and MPEG compression (MP3).

Keywords: audio watermarking, multimedia security

1 Introduction

The growth of the Internet, sudden production of low-cost and reliable storage devices, digital media production and editing technologies have led to widespread forgeries of digital documents and unauthorized sharing of digital data. As a result, the music industry alone claims multi-billion illegal music downloads on the Internet every year. Thus, it is vital to develop robust technologies to protect copyrighted digital media from illegal sharing and tampering.

Considering the embedding domain, audio watermarking techniques can be classified into time domain and frequency domain methods. In frequency domain watermarking [1-7], after taking one of the usual transforms such as the Discrete/Fast Fourier Transform (DFT/FFT) [4-6], the Modified Discrete Cosine Transform (MDCT) or the Wavelet Transform (WT) from the signal [7, 9], the hidden bits are embedded into the resulting transform coefficients. In [4-6], which were proposed by the authors of this paper, the FFT domain is selected to embed watermarks for making use of the translation-invariant property of the FFT coefficients to resist small distortions in the time domain. In fact, using methods based on transforms provides better perceptual quality and robustness against common attacks at the price of increasing the computational complexity.

In the algorithm suggested in this paper, we select the middle frequency band of the FFT spectrum (4–12 kHz) for embedding the secret bits. The selected band is divided into short frames and a single secret bit is embedded into each frame. Based on corresponding secret bit, all samples in each frame should be changed by the average of all samples or the average multiplied by a factor. If the secret bit is “0”, all FFT magnitudes should be changed by the average of all FFT magnitudes in the frame. If the secret bit is “1”, we divide the FFT samples into two groups based on the sequence and, then, we change the magnitude of the first group using a scale factor, α, multiplying the average of all samples and the magnitude of second group multiplying (2 – α) by the average. These changes either in embedding “0” or “1”, keep the average of the frame unchanged after embedding. Using the average of a frame is very useful to increase the robustness against attacks, whereas embedding a secret bit into a single sample is usually fragile. Using FFT magnitudes, $\sqrt{\mathrm{real}^2 + \mathrm{imag}^2}$, results in better robustness against attacks compared to using the real or the imaginary parts only.

The experimental results show that this method achieves a high capacity (about 0.5 to 4 kbps), provides robustness against common signal processing attacks and entails very low perceptual distortion.

The rest of the paper is organized as follows. In Section 2, the proposed method is presented. In Section 3, the experimental results are shown. Finally, Section 4 summarizes the most relevant conclusions of this research.

2 Proposed scheme

In this scheme, we use the following method to embed a bit stream (secret bits) into the FFT magnitudes. First, based on the desired capacity, transparency and robustness, the frequency band, frame size and scale factor should be selected. The selected band is divided into short frames and the average of the FFT magnitudes of each frame is calculated. Each single secret bit of the stream is embedded in a frame. The average of the FFT magnitudes of a frame plays a key role in the embedding and extracting processes. In the embedding process, all FFT samples in a frame are changed with a value related to the average which depends on the secret bit. In the extracting process, the secret bits are detected based on the average of the frame in the decoder. Using the average of FFT magnitudes improves robustness, since the average of the frame is more stable than FFT samples. In addition, we have chosen the FFT domain to embed the hidden data in order to exploit the translation-invariant property of the FFT transform such that small distortions in the time domain can be resisted. Compared to other schemes, such as quantization or odd/even modulation, keeping the relationship of FFT coefficients is a more realistic approach under several distortions.

An extensive work has been performed over the years in understanding the characteristics of the human auditory system (HAS) and applying this knowledge to audio compression and audio watermarking. Human beings tend to be more sensitive towards frequencies in the range from 1 to 4 kHz. Based on the HAS, the human ear sensitivity in higher frequencies is lower than in low frequencies. It is thus clear that, by embedding data in the middle and high frequency bands, which is used in the proposed scheme, the distortion will be mostly inaudible and thus more transparency can be achieved.

2.1 Embedding the secret bits

The frequency band, the scaling factor (α) and the frame size (d) are three required parameters in the embedding process which have to be adjusted according to the requirements. In this section, for simplicity, we do not consider the regulation of these parameters and just use them as fixed. The effect of these parameters is analyzed in Section 3.

The embedding steps are described below.

1. Calculate the FFT of the audio signal. We can use the whole file (for short clips, *e.g.* with less than one minute) or blocks of a given length (*e.g.* 10 seconds) for longer files.
2. Divide the FFT samples in the selected frequency band into frames of size d.
3. Calculate the average of magnitudes of FFT samples in each frame by using Equation (I).

$$m_i = \frac{1}{d} \sum_{j=(i-1)d+1}^{id} |f_j| \qquad \text{(I)}$$

Where $\{f_j\}$ are the FFT samples of the selected frequency band, d is the frame size and m_i is the average of the i–th frame.

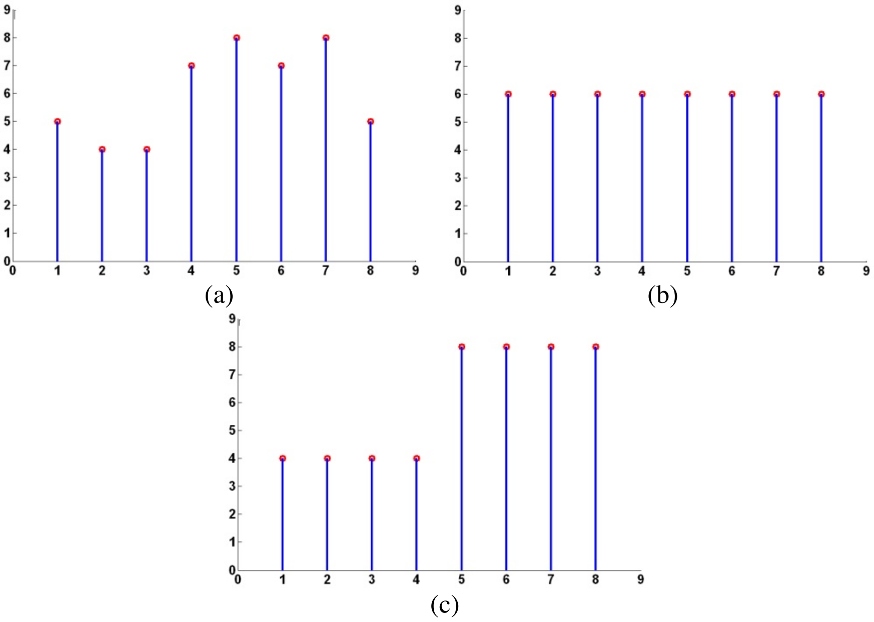
4. The marked FFT samples $\{f_j^{\square}\}$ are obtained by using Equation (II).

$$f_j^{\square} = \begin{cases} \alpha m_i & if \ \operatorname{mod}(j,d) < d/2\,,\ w_l = 1 \\ (2-\alpha) m_i & if \ \operatorname{mod}(j,d) \geq d/2,\ w_l = 1 \\ m_i & \qquad w_l = 0 \end{cases} \qquad \text{(II)}$$

Where $i = \lfloor j/d \rfloor + 1$, w_l is the l-th bit of the secret stream, $0 < \alpha < 1$ is a scaling factor and mod denotes the residual function. Each secret bit is embedded in a suitable frame and thus, after embedding the bit, the index l is incremented and the next secret bit is embedded into the next suitable frame.

5. In the previous embedding steps, the FFT phases are not altered. The marked audio signal in the time domain is obtained by applying the inverse FFT with the new magnitudes and the original FFT phases. For simplicity the embedding process is proposed for even frame size, *d.* However if we want to use an odd frame size, the embedding process for zero secret bit embedding is the same as above and, for embedding one, we need to change the middle sample to the average, m_i. Fig. 1 shows the FFT samples of a frame of size 8. Fig. 1(a) shows the original FFT samples before modification. Fig. 1(b) depicts embedding “0”, where all samples are changed by the average of all samples, which is 6 in this case. Fig. 1(c) illustrates that, to embed “1”,

the first four samples are changed by αm_i and the last four samples (second part) are changed by $(2-\alpha)m_i$. In this example, α equals to 0.5.



(a)

(b)

(c)

Fig 1. (a) Original FFT samples. (b) marked FFT samples after embedding "0". (c) marked FFT samples after embedding "1".

2.2 Extracting the secret bits

The watermark extraction is performed by using the FFT transform and the parameters, which can be considered as side information. The scale factor, frame size and the frequency band can be transmitted in a secure way to the decoder or they could be embedded using some fixed settings. For example, we could use default parameters to embed only the value of the adjusted parameters. Then, in the decoder, the adjusted parameters would be extracted by using the default parameters and the secret bits would be obtained using the extracted adjusted parameters. Since the host audio signal is not required in the detection process, the detector is blind. The detection process can be summarized in the following steps:

1. Calculate the FFT of the marked audio signal.
2. Divide the FFT samples in the selected frequency band into frames of size d.
3. Compute the average of magnitudes of marked FFT samples in each frame by equation (III)

$$m_i^{\square} = \frac{1}{d}\sum_{j=(i-1)d+1}^{id} \left|f_j^{\square}\right| \qquad \text{(III)}$$

4. To detect a secret bit in a frame, each sample should be examined to check if it is a zero frame ("0" embedded) or a one frame ("1" embedded). Then, depending on the evaluation for all samples in a current frame, a secret bit can be extracted. To determine the frame type, we define a threshold as a function of the average of corresponding sample to detect when "0" is embedded and "1" is embedded. This threshold for samples in the first part is $(1+\alpha)m_i/2$ and for second part of the frame is $(3-\alpha)m_i/2$.

I.e. if a sample in the first part of the frame is less than threshold, $(1+\alpha)m_i/2$, it means this sample belongs to the frame which "1" was embedded in and if it is equal or larger than this threshold, "0" was embedded in the frame. If a sample in the second part is equal or larger than threshold, $(3-\alpha)m_i/2$, it indicates this sample belongs to the frame which "1" was embedded in and if it is less than threshold "0" was embedded in the frame. After getting information about all samples, based on the number of samples which present "0" or "1" (voting scheme) a secret bit can be detected. If the number of samples identified as "0" is equal to or larger than the half of frame size secret bit is "0", otherwise it is "1".

To increase security, pseudo-random number generators (PRNG) can be used to change the secret bit stream to another stream which makes it more difficult for an attacker to extract the secret information. For example, the embedded bitstream can be constructed as the XOR sum of the real watermark and a pseudo-random bit stream. The seed of the PRNG would be required as a secret key both at the embedder and the detector [15].

3 Experimental results

To evaluate the performance of the proposed method, horn in *horn23_2*, and Violoncello in *vioo10_2* have been selected from the Sound Quality Assessment Material (SQAM) [10] which is popular for evaluation of properties of watermarking schemes. Also, to consider the applicability of the scheme in a real scenario, the songs "Citizen, Go Back to Sleep" (1:57) and "Do You Know Where Your Children Are" (2:31) included in the album *Rust* by No, Really [11] have been selected. All audio clips are sampled at 44.1 kHz with 16 bits per sample and two channels. The experiments have been performed for each channel of the audio signals separately.

Considering a trade-off between capacity, transparency and robustness is the main challenge for audio watermarking applications. The following conditions can be assumed to obtain different capacity, transparency and robustness:

1. No robustness: in this case, very high capacity and transparency can be achieved.

2. Semi-robustness: robustness against MP3 compression and common attacks is demanded. In this case, more distortion should be accepted, compared with Condition 1.

3. Robustness against many attacks with wide range of changes is desirable. This is more complicated than the previous conditions since we need robustness against most varied attacks. Thus, according to the trade-off between capacity, transparency and robustness, a sacrifice in capacity and transparency is required.

The Objective Difference Grade (ODG) has been used to evaluate the transparency of the proposed algorithm. The ODG is one of the output values of the ITU-R BS.1387 PEAQ [12] standard, where ODG = 0 means no degradation and ODG = –4 means a very annoying distortion. Additionally, the OPERA software [13] based on the ITU-R BS.1387 has been used to compute this objective measure of quality.

Tables I and II show the perceptual distortion, payload and BER under the MP3 compression attack with different bit rates. Note that different values for parameters are used to achieve a different trade-off between capacity, transparency and robustness, as usual for all watermarking systems. For example, for "Citizen, Go Back to Sleep", by changing the frame size from 4 to 8 we can get robustness against MP3-64, which is difficult with a frame size equal to 4. However, to obtain that robustness, we should accept less capacity. Also by using the same frame size and scaling factor and just changing the frequency band, better transparency and robustness is achieved and, as a trade-off, less capacity is obtained.

In this scheme, we have three parameters and each audio watermarking scheme has three main properties. Thus, we have three inputs and three outputs for a nonlinear system which works based on the human auditory system. Finding linear functions to adjust the requirements is extremely difficult and sometimes impossible. We can just use the different loops and conditions to get better results. In this scheme, we have general tuning rules which help us to reach the requirements or to get close to them very quickly. The frame size has more effect on robustness, whereas the scaling factor and frequency band have more effect on transparency and capacity. In other words, by increasing the frame size better robustness is achieved. In addition, increasing the frequency band leads to better capacity. Finally, with a scaling factor near one, better transparency can be achieved.

Note that these parameters allow to regulate the ODG between 0 (not perceptible) and –1 (not annoying), with about 1 kbps to 2 kbps capacity allowing robustness against MP3-128, which are typical requirements.

Table III illustrates the effect of several common attacks, provided by the Stirmark Benchmark for Audio (SMBA) v1.0 [14], on ODG and BER for the four selected audio test files. The parameters were selected for each signal, then the embedding method was applied, the Stirmark Benchmark for Audio (SMBA) software was used to attack the marked files and, finally, the detection method was applied for the attacked files. The ODG in Table III is calculated between the marked and the attacked-marked files. The parameters of the attacks are defined based on SMBA web site [14]. For example, in ADDFFTNoise, 2-4 shows the FFT size and 0-20 shows the strength. This row illustrates that any value in the range 0–20 for the strength and 2-4 for the FFT size could be used without any significant change in BER. In fact, this table provides the average results for the test signals based on BER and, in the case with the same BER, based on the limitation of the parameters. It can be seen that the proposed scheme produces excellent robustness against all these attacks (BER close to zero) even if the attacks significantly distort the audio files (even for ODG lower than –3).

Table I. Results of 2 mono one instrument signals (robust against Table III attacks)

Audio File		Horn			Violoncello		
Time (m:sec)		0:31			0:37		
Type		Fragile	Semi-robustness	Robustness	Fragile	Semi-robustness	Robustness
Factor (α)		0.9	0.75	0.6	0.9	0.75	0.6
Frame size (d)		2	4	8	2	4	8
Frequency band (kHz)		4–12	4–11	4–11	4–12	4–11	6–10
SNR (dB)		45	42	40	46	43	41
MP3 Attack	***Rate (kbps)***	192	128	64	192	128	64
	BER	0.14	0.04	0.04	0.07	0.05	0.11
ODG of marked		– 0.1	– 0.4	– 0.6	– 0.3	– 0.6	– 0.7
Payload (bps)		4025	1762	881	2025	1012	506

Table II. Results of 2 real song signals (robust against table III attacks)

Audio File		Citizen, Go Back to Sleep				Do You Know Where Your Children Are		
Time (m:sec)		1:57				2:31		
Type		Fragile	Semi robustness	Semi robustness	Robustness	Fragile	Semi robustness	Robustness
Factor (α)		0.9	0.75	0.75	0.6	0.9	0.75	0.6
Frame size (d)		2	4	4	8	2	4	8
Frequency band (kHz)		4–12	4–11	6–10	4–11	6–10	6–10	6–10
SNR (dB)		29	31	37	33	40	37	35
MP3 Attack	***Rate (kbps)***	192	128	128	64	192	128	64
	BER	0.03	0.03	0.02	0.07	0.03	0.03	0.08
ODG of marked		– 1.4	– 1.4	– 1.0	– 1.5	–0.8	–0.9	–0.9
Payload (bps)		4025	1762	1012	881	2025	1012	506

Table III. Robustness test results

Attack name	**ODG of attacked file**	*Parameters*	*BER*
AddDynNoise	–1.3 to –1.7	1-10	0.0 to 0.07
ADDFFTNoise	– 0.2 to –0.8	2 to 4, 1 to 20	0.01 to 0.11
Addnoise	– 0.8 to –1.3	1 to 500	0.0 to 0.03
AddSinus	–3.1 to –2.5	1 to 4000, 1 to 7000	0.0
Amplify	–0.2 to –0.0	10 to 120	0.0 to 0.09
Invert	–3.6 to –2.8	–	0.0
LSBZero	– 0.2 to 0.0	–	0.0
RC_HighPass	–3.7 to –2.9	1kHz to 20 kHz	0.0 to 0.03
RC_LowPass	–3.5 to –0.4	1kHz to 20 kHz	0.0 to 0.03

In order to reduce computation time and memory usage, songs can be divided into small clips, e.g. 10 seconds each. Then, the synchronization method described in [9] and the embedding algorithm described in this paper was applied for each clip separately. A more recent synchronization scheme with better transparency properties is presented in [16].

The method proposed in this paper has been compared with several recent audio watermarking strategies. Almost all the audio data hiding schemes which produce very high capacity are fragile against signal processing attacks. Because of this, it is not possible to establish a comparison of the proposed scheme with other audio watermarking schemes which are similar to it as capacity is concerned. Hence, we have chosen a few recent and relevant audio watermarking schemes in the literature. In Table IV, we compare the performance of the proposed watermarking algorithm and several recent audio watermarking strategies robust against the MP3 attack.

[1, 2, 8] have low capacity but are robust against common attacks. [3] Evaluates distortion by mean opinion score (MOS), which is a subjective measurement, and achieves transparency between imperceptible and perceptible but not annoying (MOS = 4.7).

Capacity, robustness and transparency are the three main properties of an audio watermarking scheme. Considering a trade-off between these properties is necessary. E.g. [1] proposed a very robust, low capacity and high distortion scheme. However [3] and the proposed scheme lead to high capacity and low distortion but they are not as robust as the low-capacity method described in [1]. The scheme presented in [4], which was also proposed by the authors of this paper, has good properties, but the scheme proposed in this paper can manage the needed properties better since there are three useful adjustable parameters. For example, in the proposed scheme by using frame size of d = 8 getting robustness against MP3–64 is easy. On the other hand, in [4], low bit rate MP3 compression was not considered.

Table IV. Comparison of different watermarking algorithms

Algorithm	Capacity (bps)	Imperceptibility SNR (dB)	Imperceptibility (ODG)
[1]	2	42.8 to 44.4	–1.66 to –1.88
[8]	2.3	Not reported	Not reported
[2]	4.3	29.5	Not reported
[4]	2996	30.55	–0.6
[3]	689	Not reported	Not reported
Proposed	506 to 4025	29 to 46	–0.1 to –1.5

This comparison shows the superiority in both capacity and imperceptibility of the suggested method with respect to other schemes in the literature. This is particularly relevant, since the proposed scheme is able of embedding much more information and, at the same time, introduces less distortion in the marked file.

4 Conclusions

In this paper, we present a high-capacity watermarking algorithm for digital audio which is robust against common audio signal processing. A scaling factor, the frame size and the selected frequency band are the three adjustable parameters of this method which regulate the capacity, the perceptual distortion and the robustness of the scheme accurately. Furthermore, the suggested scheme is blind, since it does not need the original signal for extracting the hidden bits. The experimental results show that this scheme has a high capacity (0.5 to 4 kbps) without significant perceptual distortion and provides robustness against common signal processing attacks such as added noise, filtering or MPEG compression (MP3).

Acknowledgement

This work is partially supported by the Spanish Ministry of Science and Innovation and the FEDER funds under the grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER-INGENIO 2010 CSD2007-00004 ARES.

References

1. Xiang S., Kim H.J., Huang J., "Audio watermarking robust against time-scale modification and MP3 compression," *Signal Processing*, Vol. 88 No.10, pp.2372-2387, October, 2008.

2. Mansour M. and Tewfik A., “Data embedding in audio using time-scale modification,” *IEEE Trans. Speech Audio Process.*, Vol. 13, No. 3, pp. 432–440, 2005.
3. Garcia-Hernandez J. J., Nakano-Miyatake M. and Perez-Meana H., “Data hiding in audio signal using Rational Dither Modulation”, *IEICE Electron. Express*, Vol. 5, No. 7, pp. 217-222, 2008.
4. Fallahpour M., Megías D., “High capacity audio watermarking using FFTamplitude interpolation” *IEICE Electron. Express*, Vol. 6, No. 14, pp. 1057-1063, 2009.
5. Fallahpour M., Megías D., “High Capacity Method for Real-Time Audio Data Hiding Using the FFT Transform”, Advances in Information Security and Its Application, Springer-Verlag. pp. 91-97, 2009.
6. Fallahpour M., Megías D., “Robust high-capacity audio watermarking based on FFT amplitude modification” *IEICE Trans on Information and Systems,* Vol.E93-D, No.01, pp.87-93, Jan. 2010.
7. Fallahpour M., Megías D., “DWT–based high capacity audio watermarking”, *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences,* Vol.E93-A, No.01, pp.331-335, Jan. 2010.
8. Li W., Xue X., “Content based localized robust audio watermarking robust against time scale modification” *IEEE Trans. Multimedia*, Vol. 8, No. 1, pp. 60-69, Feb. 2006.
9. Wang X.-Y. and Zhao H.. “A novel synchronization invariant audio watermarking scheme based on DWT and DCT”. *IEEE Trans. on Signal Processing*, Vol. 54, No 12, pp. 4835–4840, December 2006.
10. SQAM Sound Quality Assessment Material, http://www-sipl.technion.ac.il/Info/Downloads_DataBases_Audio_Quality_Assessment_Readme_e.shtml
11. No, Really, “Rust”. http://www.jamendo.com/en/album/7365. (Last checked on March 15th, 2011).
12. Thiede T., Treurniet W. C., Bitto R., Schmidmer C., Sporer T., Beerens J. G., Colomes C., Keyhl M., Stoll G., Brandenburg K., and Feiten B., “PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality,” *Journal of the AES*, Vol. 48 No.1/2, pp. 3–29, 2000.
13. OPTICOM OPERA software site. http://www.opticom.de/products/opera.html
14. Stirmark Benchmark for Audio. http://wwwiti.cs.uni-magdeburg.de/~alang/smba.php. (Last checked on March 15th, 2011).
15. Megías D., Herrera-Joancomartí J., and Minguillón J.. “Total disclosure of the embedding and detection algorithms for a secure digital watermarking scheme for audio”. *Proceedings of the Seventh International Conference on Information and Communication Security*, pp. 427-440, Beijing, China, December 2005.
16. Megías D., Serra-Ruiz J. and Fallahpour M., “Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification”. *Signal Procesing*. Vol. 90, No. 12, pp. 3078-3092. December 2010.