Article

Autocorrelation Modulation-Based Audio Blind Watermarking Robust Against High Efficiency Advanced Audio Coding

Yiyu Hong ¹ and Jongweon Kim ^{2,*} ¹ Department of Copyright Protection, Sangmyung University, Seoul 03016, Korea² Department of Electronics Engineering, Sangmyung University, Seoul 03016, Korea

* Correspondence: jwkim@smu.ac.kr

Received: 6 June 2019; Accepted: 8 July 2019; Published: 10 July 2019

Abstract: High Efficiency Advanced Audio Coding (HE-AAC) is a lossy compression method for digital audio data which supplies high-quality audio at a very low bit rate. In this paper, the audio blind watermarking algorithm, on the basis of autocorrelation modulation, is introduced to maximize the robustness against low-bit rate HE-AAC. The watermark is embedded by modulating the normalized correlation of the original signal as well as its delayed version. The signal-to-noise ratio of before and after HE-AAC compression decides the strength of the embedding watermark. The watermarking embedding strength is guaranteed by the feedback process. The effectiveness of the proposed method is proven using the Perceptual Evaluation of Audio Quality algorithm and bit error rate of recovered watermarks under HE-AAC compression on mono, stereo and 5.1 channel audio. Experimental results show that the proposed method provides good performance in terms of imperceptibility, robustness and data payload compared with some recent state-of-the-art watermarking methods under an MPEG-2 Audio Layer III (MP3) compression attack.

Keywords: audio blind watermarking; autocorrelation modulation; high efficiency advanced audio coding

1. Introduction

Over the last couple of years, High Efficiency Advanced Audio Coding (HE-AAC) [1–3] has become one of the most important enabling technologies for state-of-the-art multimedia systems. It delivers compact disc (CD) quality stereo at 48 kbps and 5.1 channel surround sound at 128 kbps. This efficiency level is the optimal choice for current internet content delivery. At the same time, it has basically realized novel applications in mobile digital broadcasting and mobile markets. It also enhances audio services while enhancing video services, such as digital television. Without reducing the quality of the audio signal, more bits can be distributed to video signal by coupling HE-AAC with MPEG-4 video.

Though HE-AAC is popular in the field of multimedia systems, as far as we know, there is no audio watermarking related scientific paper that have developed a watermarking algorithm for HE-AAC or have evaluated performance under it. Therefore, the objective of this study was to design an audio watermarking algorithm to maximize the robustness against low-bit rate HE-AAC compression. This paper is an expansion of our previously published conference paper [4]. We illustrate this algorithm in more detail through more experiments and visualization. The parameters of the proposed algorithm are optimized by thorough experiments, and its performance on multi-channel audio is also added. Moreover, synchronization design is discussed.

The rapid growth of computer multimedia technology as well as the broad utilization of the internet have promoted the distribution as well as transmission of digital multimedia content. Digital

watermarking [5] refers to the procedure of embedding data into digital multimedia content such as audio, video, and image. It was initially utilized for security-related purposes like copyright protection and source tracking, but nowadays it is dedicated to various non-security-oriented applications [6], such as broadcast monitoring and automatic content recognition.

An audio watermarking method must comply with the following requirements [7]. (1) Imperceptibility: The perception of the original audio signal is similar to that of the watermarked audio signal. (2) Security: Watermarks can only be detected by authorized personnel. (3) Payload: It is a number of bits in which audio can be inserted in a unit of time. The watermarking data payload needs to be greater than 20 bps (bit/second). (4) Robustness: This means that the watermark should be resistant to common malicious attacks as well as signal processing. However, there is no algorithm that can meet all the demands mentioned above. The goal of the watermarking algorithm is to achieve an appropriate trade-off between requirements. Security-oriented applications may require high robustness and security, because it is more likely to receive malicious attacks. By contrast, non-security-oriented applications may require high payload and a certain degree of robustness against one or two specific attacks that are known in advance.

The organization of the rest of the thesis is as follows. In the next part, we give a simple instruct of HE-AAC and audio watermarking algorithms. In the third part, we introduce the audio watermarking algorithm in detail. Experimental results are discussed in Section 4, and these evaluate the function of the algorithm in the data payload, robustness under compression attack, and imperceptibility aspects. Finally, Section 5 summarizes and analyzes the future direction of development.

2. Related Works

2.1. HE-AAC

HE-AAC is an extension of low complexity AAC (AAC-LC), which provided an optimization for those applications of low bit rate like streaming audio. It has the characteristics of standardization and has become a configuration file for the MPEG-4 audio standard. Two versions of HE-AAC are available: HE-AAC v1 and HE-AAC v2. HE-AAC v1 includes two principal technologies, spectral band replication (SBR) and AAC-LC. In contrast to version 1, version 2 additionally uses a technique named parametric stereo (PS) that compresses stereo signals more effectively. Figure 1 shows the HE-AAC techniques family.

Figure 1. HE-AAC techniques

AAC-LC is a commonly used audio compress codec. It removes the audio signal by utilizing a psychoacoustic model and only keep auditory information. Though it has good audio quality at a bit rate of 128 kbps for mono audio, the audio quality begins to significantly decline below this bit rate. In order to achieve good audio quality as well as a low compression bit rate, two complementary technologies, SBR and PS, are exploited.

SBR can improve compression efficiency in the frequency domain. At low frequencies, there is a lot of important audio information. SBR utilizes the strong relevance between high frequency and low frequency audio signals to reconstruct the high frequency signals through approximation as well as the transposition of low frequency signals rather than transfer the data of high frequency audio.

The compression efficiency of stereo signal is improved by PS. PS technology mixes stereo signals downward into mono-channel signals, extracts parametric stereo data, and describes the differences as well as similarities of two channels. When decoding, the original stereo signal is reconstructed using parametric stereo data and a mono-channel signal.

2.2. Audio Watermarking Algorithms

Over the past few years, most proposed audio watermarking algorithms [8–24] can be grouped into two categories: Spread spectrum (SS) modulation [8–10] and quantization index modulation (QIM) [11–14]. All these methods utilize either a time domain [15,16] or a transform domain such as fast Fourier transform [17], discrete cosine transform [18], or discrete wavelet transform [19]. From the watermark extraction mode, digital watermarking algorithms can be divided into three schemes: Blind, semi-blind and non-blind. Among them, blind methods, which extract watermarks without knowing the host audio signal, are highly desirable, because semi-blind and non-blind methods are not applicable to most practical applications.

The SS-based watermarking scheme embeds a watermark bit into a host audio segment by using pseudorandom number sequences which are shaped to fit under the masking threshold of the audio signal. At the extractor, the watermarks are extracted by a sliding correlator that correlates the received signal to the predefined spread spectrum template. One of the main drawbacks of the existing SS-based audio watermarking methods is their low embedding capacity. Hence, their main task is to increase embedding capacity while maintaining high imperceptibility and robustness.

The QIM approach uses a set of quantizers to quantify the host signal features to embed watermarking data, and each quantizer is associated with different information. In most cases, to obtain high robustness, quantization is implemented to some coefficients in the transform domain rather than to signal samples. Though the QIM approach usually has merits of low complexity and high embedding capacity, it has low robustness.

Another audio watermarking algorithm is autocorrelation modulation [20–22]. Its basic idea is that inserting a delayed or advanced version of the host signal itself can modify autocorrelation. In other words, it uses a delayed and modulated version of the host signal as a watermark signal (the difference between the host and watermarked signal), which is then inserted back to the host signal to make the final watermarked signal. A common watermarking approach is to introduce the watermark signal as noise, but a drawback to these approaches is that lossy audio compression algorithms tend to remove most imperceptible artifacts, including typical low dB noise. Autocorrelation modulation introduces changes to the host signal that are characteristic of environmental conditions rather than random noise. Therefore, using a host signal as a watermark signal can be robust against lossy audio compression. In this paper, we present an autocorrelation modulation-based audio blind watermarking algorithm which maximizes robustness against low-bit rate HE-AAC compression. In the next section, the detail of the proposed audio watermarking method is described.

3. Proposed Audio Watermarking Method

This section first provides the following definitions:

- The host audio signal $x(t)$ is frequency bandpass filtered to outputs the filtered audio signal $s(t)$ which is further modulated to be a watermark signal. There are two purposes for the frequency bandpass filtering process. One is to acquire an amount of the host signal, which will reduce disturbance to the audio signal. The other is to avoid embedding watermarks around a high frequency, as the HE-AAC will entirely remove the signals at the range of high frequency on encoding process. The lower and upper cutoff frequencies are denoted as F_{lc} and F_{uc} , respectively.
- Then, the filtered audio signal is divided into successive frames, each of which has length T_f and contains two non-overlapping sections of samples. These two sections have equal length, and we call them as the front subframe and the back subframe in this paper.

- One piece of watermark information is represented as one binary bit of value '0' or '1,' which is embedded in one frame.
- The normalized correlation of the original signal and its delayed version (NCOD) was selected as the characteristic of each subframe. The embedding and extracting of watermark bits are decided by the difference of NCOD relations between the front subframe and back subframe. The NCOD item is calculated as follows:

$$C_{i1} = \sum_{t=0}^T [s(T_f \cdot i + t) \cdot s(T_f \cdot i + t + \tau)] \quad (1)$$

$$N_{i1} = \sum_{t=0}^{T_f/2} s(T_f \cdot i + t)^2 \quad (2)$$

$$NC_{i1} = C_{i1} / N_{i1} \quad (3)$$

$$C_{i2} = \sum_{t=T_f/2}^{T_f/2+T} [s(T_f \cdot i + t) \cdot s(T_f \cdot i + t + \tau)] \quad (4)$$

$$N_{i2} = \sum_{t=T_f/2}^{T_f} s(T_f \cdot i + t)^2 \quad (5)$$

$$NC_{i2} = C_{i2} / N_{i2} \quad (6)$$

where i ($i = 0, 1, 2, \dots$) represents the frame index. The integration time T (in samples) should be $T \leq T_f - \tau$ to avoid intersymbol interference. For a frame, the NC_{i1} and NC_{i2} are the NCOD values of the front subframe and back subframe, respectively. Since the correlation value is normalized, $NC \in [-1, 1]$.

- The difference between NC_{i1} and NC_{i2} is computed to obtain:

$$D_i = NC_{i1} - NC_{i2} \quad (7)$$

If the watermark bit is '1,' D_i should larger than 0 ($D_i > 0$). If the watermark bit is '0,' D_i should smaller than 0 ($D_i < 0$). Basically, raising $|D_i|$ enhances robustness. On the basis of the value of D_i , one bit of watermark information is embedded and extracted from each frame.

3.1. Watermark Embedding Scheme

Figure 2 presents the process of the watermark embedding. Firstly, bandpass filtering is applied to the host signal $x(t)$ to get the filtered signal $s(t)$. Then, the filtered signal is separated into successive frames. After that, they are divided into the front subframes and back subframes. We compute a gain based on the scale factor ξ_i , original D_i as well as watermark bit w_i , and then multiply the gain to the front and back subframe with mutually opposite signs to generate the watermark signal. In the end, the watermark signal is added back into the host signal with a specified delay τ to modify the NCOD values.

Figure 2. Watermark embedding procedure.

When modulated subframes are added back to host signal, there are sudden changes at the boundaries which cause audible noise. To avoid this, amplitude of the boundaries of the modulated subframes are attenuated. In our implementation, the window function is calculated by following equation:

$$f_w(t) = \begin{cases} (t-1) \cdot \Delta k, & 1 \leq t < l_a \\ 1, & l_a \leq t \leq T-l_a \\ (T-t) \cdot \Delta k, & T-l_a < t \leq T \\ l_a \cdot |\Delta k| = 1, & 1 < l_a < T/2 \end{cases} \quad (8)$$

where Δk is attenuation interval and l_a is attenuation length at the boundary. We set $|\Delta k| = 1/l_a$. Figure 3 shows the window.

Figure 3. Window function.

The watermark embedding procedure of each frame is illustrated by pseudo code in Algorithm 1. The b_i is the bipolar term of w_i , $w_i \in \{0, 1\} \Rightarrow b_i \in \{-1, +1\}$. The TH_i is threshold for determining a gap of NCOD relation between the front subframe and back subframe.

Algorithm 1. Watermark embedding process

- 1: $b_i = w_i \cdot 2 - 1$
 - 2: $D_i = NC_{i1} - NC_{i2}$
 - 3: $TH_i = \xi_i \cdot b_i$
 - 4: **if** ($w_i == 1$ and $D_i > TH_i$) or ($w_i == 0$ and $D_i < TH_i$)
 - 5: **return** // return without performing any action
 - 6: **else**
 - 7: $g_i = TH_i - D_i$
 - 8: **end**
 - 9: front subframe = front subframe $\cdot g_i \cdot f_w$
 - 10: back subframe = back subframe $\cdot (-g_i) \cdot f_w$
-

If D_i has already met the requirements of the watermark embedding condition, then no further modulating operation to the frame is required. Otherwise, a gain value g_i is calculated by TH_i minus D_i , and this gain is multiplied to the front subframe. In contrast, the back subframe is multiplied by opposite sign of g_i . These subframes are further multiplied by the f_w window function.

In other words, the natural NCOD is firstly calculated, and then the necessary modifications are determined. To get a well visual effect, Figure 4 illustrates the procedure of the watermark embedding of two frames.

Figure 4. Watermark embedding diagram.

The scale factor ξ_i is utilized to balance requirements between robustness and transparency. The main purpose of the proposed audio watermarking algorithm is to be robust against HE-AAC compression attack, so we designed the algorithm to adaptively adjust ξ_i by using the signal-to-noise ratio (SNR) between host signal and the HE-AAC compressed signal. For convenience, we call the SNR as CSNR in this paper. Figure 5 presents the calculation process of ξ_i .

Figure 5. Scale factor ξ_i calculation.

We can calculate the CSNR of each frame by utilizing following formula:

$$SNR_i = 20 \cdot \log_{10} \frac{\sum_{t=T_{f,i}}^{T_{f,i+1}} |x(t)|}{\sum_{t=T_{f,i}}^{T_{f,i+1}} |x(t) - y(t)|} \quad (9)$$

where $x(t)$ is host signal and $y(t)$ is its compressed version. It can be indicated that the lower the CSNR value, the greater the compression is performed to the frame, which means lots of redundant signals are removed. Thus, a frame with a lower CSNR value would get a larger ξ_i value to guarantee robustness and vice versa.

3.2. Watermark Extraction Scheme

Figure 6 shows the proposed watermark extraction procedure, where the $|\cdot|$ symbol means the absolute function. The procedure needs to know the embedding parameters in advance: The passband frequency $[F_{lc}, F_{uc}]$, the frame length T_f , and the delay τ . The watermark extraction procedure can be regarded as a part of the embedding procedure. Assuming that the start point of watermark embedding has been found, the watermarked signal $\bar{x}(t)$ is firstly filtered to produce the filtered watermarked signal $\bar{s}(t)$. After then, the formula (1)–(7) is utilized to calculate the $\bar{D}_i$ of each frame. If $\bar{D}_i > 0$, the detected bit is '1.' If $\bar{D}_i < 0$, the detected bit is '0.' Note that the proposed method does not need the original signal in the watermark extraction process, which indicates that we proposed a blind audio watermarking method.

Figure 6. Watermark extraction procedure.

3.3. Feedback Process

A feedback process was added to the watermark embedding procedure (Figure 1) to ensure that the watermark is embedded with enough strength. As illustrated in Figure 7, after embedding the watermark, the generated watermarked signal $\bar{x}(t)$ is immediately input into the watermark extractor to calculate the $\bar{D}_i$ values. By comparing the extracted $\bar{D}_i$ and TH_i , we can know whether the desired watermarking strength is achieved. If the strength is not enough, the gain g is increased by a small quantity α , and then it is put back into the watermark embedder to generate a stronger watermark signal. The feedback process works in an iterative manner.

Figure 7. Feedback process.

4. Experimental Results

In this section, we present the results of our many experiments to present the performance of the proposed audio watermarking algorithm. First, there were parameter optimization experiments, followed by a comparison of the results with some state-of-the-art audio watermarking algorithms. We also show experiment results on 5.1 channel and stereo audio. Finally, synchronization is discussed.

Like other watermarking algorithms, we used bit error rate (BER) as a metric to evaluate the robustness of the watermarking algorithm. It is defined as:

$$BER = \frac{\text{Number of incorrect bits}}{\text{Number of total bits}} \times 100\% \quad (10)$$

The objective difference grade (ODG) was utilized to measure imperceptibility, as it is one of the output values acquired according to the Perceptual Evaluation of Audio Quality (PEAQ) measurement technique prescribed by the ITU-R BS.1387 standard [25]. The ODG values were between 0.0 and −4.0 (imperceptible to very annoying), and they are shown in Table 1.

Table 1. Objective difference grades (ODG) for audio quality evaluations.

Impairment Description	ODG	Quality
Imperceptible	0.0	Excellent
Perceptible, but not annoying	−1.0	Good
Slightly annoying	−2.0	Fair
Annoying	−3.0	Poor
Very annoying	−4.0	Bad

In the rest of this part of the paper, if there is no specific notice of parameters, experiments were performed with following default parameters: Frame length $T_f = 500$ samples (96 bps), delay $\tau = 45$, integration time $T = 500/2 - 45 = 205$ samples, $l_a = 10$ samples, $\Delta k = 0.1$, $\alpha = 0.005$, and passband frequency $[F_{lc}, F_{uc}] = [2500 \sim 5500 \text{ Hz}]$. CSNR was calculated using 24 kbps HE-AAC v1 compression. We terminated the feedback process of a frame if it was conducted more than five times and it resulted in average of 2.13 feedbacks per frame.

The ODG was measured by basic PEAQ model, which was implemented from the Telecommunications & Signal Processing Laboratory of McGill University [26]. The “fdkaac.exe” software [27], which was developed by Fraunhofer, was used to conduct HE-AAC compression. The “ffmpeg.exe” [28] was used to conduct MP3 compression and decode the compressed audio.

4.1. Experiments on Mono Audio

We used seven audio clips belonging to seven different genres as host audio signals to illustrate the performance of our watermarking algorithm on mono-channel audio. They were “Drama,” “Debate,” “Sports Commentary,” “Classical Music,” “Jazz Music,” “Pop Music,” and “Rock Music.” All audio files were in the WAVE format, mono, sampled at 48 kHz, quantized with 16 bits, and 30 s long.

The scale factor ξ_i was adaptively adjusted by CSNR, as we mentioned in Section 3.1. Here, we first showed CSNR with 24 kbps HE-AAC v1 compression for two test audios and the sum of CSNR histograms of all seven test audio signals in Figure 8. From the histogram graph, we can see that most CSNR values were distributed from 0 to 15. According to the histogram distribution, we found that calculating ξ_i by following Equation (11) well balanced the trade-off between robustness and imperceptibility in our implementation.

$$\xi_i = \begin{cases} 0.2, & 14 \leq \text{CSNR}_i \\ -0.05 \cdot \text{CSNR}_i + 0.9, & 0 \leq \text{CSNR}_i < 14 \\ 0.9, & \text{CSNR}_i < 0 \end{cases} \quad (11)$$

Figure 8. Calculated CSNR of test audio (a) CSNR of “Sports Commentary” audio. (b) CSNR of “Classical Music” audio. (c) The sum of CSNR histograms of seven test audio signals.

The data payload refers to the number of bits that can be embedded into the audio signal within a unit of time and is measured in the unit of bps (bits per second). The data payload of our algorithm was determined by frame length T_f , where one bit watermark is embedded. For example, if $T_f = 500$ and the sampling rate of the host audio is 48,000 Hz, we can embed $48,000/500 = 96$ bits per second in the audio. As the frame length will affect the imperceptibility and robustness of our algorithm, in Table 2 and Figure 9, we show how the change of data payload influenced ODG and BER, where BER was calculated after the HE-AAC v1 24 kbps compression. From the table and graph, we can see that BER was under 2% when the data payload was lower than 96, but it rose rapidly from 1.77 to 8.77% when the data payload was 96–192 bps ($T_f = 250$). With the increase of data payload, the ODG was slightly increased and larger than -1 , which indicates the original and watermarked audio signals were perceptually similar and not annoying.

Besides the frame length, the other important parameter of the proposed algorithm is passband frequency. If the selected passband frequency is too low or too high, it is likely to yield bad results for BER, as the audio compression will remove most of signals in that frequency range based on the psychoacoustic model. Moreover, HE-AAC applies SBR to directly cutoff high frequencies, which will result in worse BER results. Table 3 and Figure 10 show how the selected passband frequency influences the ODG and BER, where BER was calculated after the HE-AAC v1 24 kbps compression. As shown in the table and graph, we observed that, when using a frequency higher than 3.5 kHz, BER was over 5% and rose rapidly. When using a passband frequency of 0.5~3.5 kHz, the ODG was lower than -1 , the other frequency ranges were all larger than -1 , and the BER was also worse than when using 1.5~4.5 kHz and 2.5~5.5 kHz. Therefore, we can conclude that the 1.5~5.5 kHz frequency range is a good choice for embedding watermarks with the proposed algorithm. Note that this range of frequency bands is very sensitive to the human hearing system according to the absolute threshold of hearing. Almost all audio watermarking methods were designed to avoid this range of frequency to meet the imperceptibility requirement, though our method achieved high imperceptibility on the 1.5~5.5 kHz frequency band while maintaining a low BER.

Table 2. ODG and bit error rate (BER) values under various data payloads.

Data Payload (bps)	ODG	BER (%)
24	−0.82	0.47
48	−0.66	0.61
72	−0.66	0.83
96	−0.66	1.77
144	−0.62	4.55
192	−0.57	8.77

Figure 9. ODG and BER values under various data payloads.**Table 3.** ODG and BER values under various passband frequencies.

Passband Frequency (kHz)	ODG	BER (%)
0.5~3.5	−1.17	2.96
1.5~4.5	−0.86	1.22
2.5~5.5	−0.65	1.77
3.5~6.5	−0.77	5.34
4.5~7.5	−0.76	15.06
5.5~8.5	−0.68	45.76

Figure 10. ODG and BER values under various passband frequencies.

Delay is also a significant factor which influences the performance of the proposed algorithm. If the delay is too large, the accumulation samples for calculating autocorrelation will decrease, which will cause an increase in the BER. The results in Table 4 and Figure 11 show how the delay had an effect on the ODG and BER. We can see that the delay around 45 samples gave good result where the

ODG was over -1 and the BER was under 3%. Though the ODG got better when delays were over 100 samples, BER rose over 3%

Table 4. ODG and BER values under various delays.

Delay	ODG	BER (%)
15	-0.77	2.65
45	-0.66	1.77
75	-0.56	2.1
100	-0.44	3.29
150	-0.23	7.23
200	-0.03	21.12

Figure 11. ODG and BER values under various delays.

Figure 12 and Table 5 show how the attenuation length at the boundary of the window influenced the BER and ODG. It can be observed that the ODG steadily rose as the attenuation length increased and the BER decreased. An attenuation length of 10–25 is preferable, as the BER increased quickly over 25, and the ODG was relatively low at 5.

Figure 12. ODG and BER values under various attenuation lengths of the window.

Table 5. ODG and BER values under various attenuation lengths of the window.

Attenuation Length of the Window	ODG	BER (%)
5	−0.75	1.69
15	−0.61	1.83
25	−0.54	2.26
35	−0.48	2.89

In Table 6, we illustrate our comparison results. Our comparison was based on reported results of recently published papers [8,9,12,13,23,24] and was given for the data payload, ODG, and BER under MP3 compression (32 kbps, 64 kbps, 96 kbps, 128 kbps). We also listed the BER under various HE-AAC v1 compression bitrates (16 kbps, 24 kbps, 32 kbps, 64 kbps) with various data payloads. As we can see from the table, the proposed algorithm is competitive compared with other methods at similar data payloads under MP3 compression, indicating that our algorithm is also able to be robust against MP3 compression. Comparing the BER at the same bit rate (32 kbps, 64 kbps) on MP3 and HE-AAC v1 compressions of our algorithm, we found that the BER under HE-AAC v1 was nearly half of the MP3. From the BER results at various HE-AAC v1 compression bitrates, we can see that our algorithm was able to be robust against HE-AAC v1 compression where the bit rate was higher than 24 kbps with a low BER. Unfortunately, we could not have a comparison result under HE-AAC v1, as there is no paper which has evaluated its robustness under HE-AAC v1 compression.

Table 6. Performance comparison results with state-of-the-art algorithms under MP3 compression, sorted by data payload. (“—” means not reported).

Reference	Data Payload (bps)	ODG	MP3 Compression BER (%)				HE-AAC v1 Compression BER (%)			
			32 kbps	64 kbps	96 kbps	128 kbps	16 kbps	24 kbps	32 kbps	64 kbps
[23]	43	−0.57	3.00	0.00	0.00	0.00	—	—	—	—
[8]	43	−0.55	—	0.22	0.02	—	—	—	—	—
[9]	84	−0.70	—	—	0.33	0.00	—	—	—	—
[12]	172	−1.05	6.13	—	—	—	—	—	—	—
[13]	172	−0.51	—	2.86	—	0.02	—	—	—	—
[24]	4000	−0.91	—	19	15	—	—	—	—	—
Ours	48	−0.66	0.46	0.21	0.21	0.17	9.15	0.61	0.24	0.20
	96	−0.65	1.29	0.31	0.15	0.18	18.02	1.77	0.50	0.16
	192	−0.58	6.50	2.60	1.43	1.23	30	8.77	3.56	1.50

4.2. Experiments on 5.1 Channel and Stereo Audio

Multichannel audio systems have become more and more popular in home entertainment environments. As HE-AAC is one of the most efficient audio codecs for multichannel audio, here we present experimental results under HE-AAC v1 and v2 compression by applying our watermarking algorithm on 5.1 channel audio and stereo audio, respectively. The CSNR for 5.1 channel audio was applied to HE-AAC v1 128 kbps compression, and, for stereo audio, it was applied to HE-AAC v2 48 kbps compression.

We used the twelve 5.1 channel audio types, which are listed in Table 7. All 5.1 channel audio files were in WAVE format, sampled at 48 kHz, quantized with 16 bits, and 10~20 s long. Figure 13 shows the time domain waveforms of the “Bach organ” 5.1 channel audio.

Table 7. 5.1 channel host audio used in the experiment.

No.	Audio Name	Description
1	Bach organ	Church organ; lots of stops out
2	Brass	Orchestral; lots of brass instruments
3	Harpsichord	Solo harpsichord; isolated notes
4	Mouth harp	Mouth organ, bass guitar, percussion
5	Sax Piano	Saxophone and piano
6	Trumpet	Orchestral piece
7	Applause	Applause with distinct clapping sounds
8	Chants	Small choir; large church; Gregorian chant
9	Classical	Orchestral piece; open sound
10	Radio drama	Clarinet, orchestra, male speaker, tenor, ambience
11	Sedambonjou	Atmospheric performance of Latin-American music
12	Moonriver	Mouth organ and string orchestra

Figure 13. Time domain waveforms of the “Bach organ” 5.1 channel audio.

The channels from, top to bottom, are front left (FL), front right (FR), center (C), low-frequency effects (LFE), surround left (SL), and surround right (SR). We independently embedded a watermark on each channel except LFE, as the channel showed nearly no signals in our test audio. Table 8 provides the BER of each channel of 5.1 channel audio under HE-AAC v1 128 kbps compression, as well as the corresponding ODG of each audio. The ODG was first calculated for each channel except LFE and averaged to report in the table for each audio. From the table, we can observe that there was a big difference of the BER of each channel of an audio. For example, the BER of the center channel of the No.1 audio was only 0.18%, but its front right channel BER was 5% higher than center channel, which was 6.21%. Regarding the ODG, all test audio yielded larger than -1 ODG values, except for three test audios that were slightly lower than -1 . The average BER of the twelve test audios was 2.53%, and the average ODG was -0.84 .

For testing stereo audio, we down-mixed the 5.1 channel audio by the following formula, $FL = FL + 0.707 \times FC + 0.5 \times SL$; $FR = FR + 0.707 \times FC + 0.5 \times SR$, to make test stereo audio. Figure 14 shows the time domain waveforms of the “Bach organ” stereo audio, which was made by the corresponding 5.1 channel audio. For the experiment on 5.1 channel audio, we independently embedded a watermark on each channel. Table 9 provides the BER of each channel of stereo audio under HE-AAC v2 48 kbps compression, as well as the corresponding ODG of each audio type. From the table, we can see that average BER of the twelve test audios was 1.28%, and the average ODG was -0.78 .

Table 8. BER (%) and ODG of each channel of 5.1 channel audio under HE-AAC v1 128 kbps compression.

No.	FL	FR	C	LFE	SL	SR	Avg BER	ODG
1	2.93	6.21	0.18	—	3.87	5.21	3.68	−0.70
2	3.95	2.44	0.29	—	5.03	1.94	2.73	−0.84
3	0.88	1.24	0.12	—	1.18	0.65	0.81	−1.06
4	1.87	1.74	0.37	—	1.55	1.99	1.51	−0.73
5	3.83	3.32	0.22	—	0.30	0.30	1.59	−0.54
6	3.81	1.57	0.00	—	3.47	2.35	2.24	−0.86
7	0.06	0.24	0.77	—	1.42	4.56	1.41	−0.48
8	0.59	1.00	2.17	—	0.06	1.00	0.96	−0.86
9	4.90	3.74	0.27	—	2.76	3.48	3.03	−1.14
10	7.84	8.67	0.46	—	8.39	9.03	6.88	−0.93
11	2.78	2.83	0.64	—	2.95	2.78	2.39	−1.04
12	3.47	4.02	0.64	—	3.29	4.48	3.18	−0.95
Avg	—	—	—	—	—	—	2.53	−0.84

Figure 14. Time domain waveforms of the “Bach organ” stereo audio.**Table 9.** BER (%) and ODG of each channel of stereo audio under HE-AAC v2 48 kbps compression.

No.	Left Channel	Right Channel	Avg BER	ODG
1	0.70	1.70	1.20	−0.42
2	0.57	0.93	0.75	−0.78
3	0.35	0.88	0.62	−0.98
4	0.56	0.12	0.34	−0.62
5	1.62	0.81	1.22	−0.73
6	1.01	0.78	0.90	−0.83
7	0.06	0.06	0.06	−0.45
8	0.18	0.64	0.33	−0.83
9	1.78	0.45	1.12	−1.07
10	3.19	3.01	3.10	−0.87
11	4.80	4.40	4.60	−0.87
12	1.10	1.19	1.15	−0.89
Avg	—	—	1.28	−0.78

Furthermore, we present the BER of the multichannel audio under various bit rate HE-AAC compressions in Table 10. It can be seen that our watermarking algorithm was able to be robust against HE-AAC v1 and HE-AAC v2 compression when the bit rate was higher than 128 and 32 kbps, respectively, for 5.1 channel and stereo audio, as the BER was under 3% at a data payload of 96 bps, and the ODG was larger than −1. However, our algorithm showed a bad robustness for extremely low bit rate compression, like 5.1 channel audio with HE-AAC v1 64 kbps and stereo audio with HE-AAC v2 16 kbps, where the BER was over 20%.

Table 10. Averaged BER (%) of test multichannel audio under various bit rate HE-AAC compressions.

5.1 Channel				Stereo			
64 kbps	96 kbps	128 kbps	160 kbps	16 kbps	24 kbps	32 kbps	48 kbps
24.04	6.67	2.53	0.54	27.9	7.87	2.30	1.29

4.3. Synchronization Design

In order to apply the audio watermarking technique in real situations, integrity information should be embedded repeatedly in the audio signal within each block unit. Each block must be further segmented to the synchronization codes part and the watermark information part. Synchronization is an effective way to accurately identify the watermark location.

In our algorithm, we can use a certain number of consecutive frames as the synchronization codes parts by embedding a fixed watermark sequence. In the extraction process, by checking whether the extracted consecutive watermark bits are same as the synchronization codes or a certain proportion is correct, we can determine the start of a block.

As we used the correlation between the signal and a delayed version of itself to extract the watermark, finding the exact start sample of each watermarked frame was not necessary because the correlation values $\bar{D}$ around the start point were similar. Figure 15 shows the $\bar{D}$ values calculated for each of the 5000 samples of “Drama” audio after HE-AAC v1 24 kbps compression, where the frame length was 500 samples and the watermark bits [1, 0, 1, 0, 1, 0, 1, 0, 1, 0] were embedded. As we mentioned in Section 3.2, the $\bar{D}$ value was calculated to determine the extracted watermark bit. From the figure, we can see that the $\bar{D}$ values changed slowly and were not likely to have an abrupt change within several samples. Hence, we could efficiently find the start of a block by skipping a number of samples each time when determining if the synchronization codes are found instead of brute-force searching through each sample. The selection for number of skipping samples is affected by frame length, and a larger number of skipping samples could be chosen with a longer frame length.

Figure 15. Correlation $\bar{D}$ values calculated for each of 5000 samples of “Drama” audio after HE-AAC v1 24 kbps, where the watermark bit [1, 0, 1, 0, 1, 0, 1, 0, 1, 0] is embedded.

To ensure extracting complete information, according to the BER and the length of watermark information, an appropriate error correcting code should be applied.

5. Conclusions

In this paper, an autocorrelation modulation based an audio blind watermarking algorithm was proposed. The difference in NCOD relations between the front subframe and back subframe was used to embed and extract the watermark bits of each frame. The SNR before and after HE-AAC compression was used to adaptively adjust the scale factor, which was further used to balance the trade-off between robustness and imperceptibility. In addition, a feedback process was added to the watermark embedding procedure to ensure that the watermark was embedded with enough strength.

By using optimized parameters, the experimental results present the fact that our algorithm is able to be robust against low bit rate (24 kbps for mono, 32 kbps for stereo and 128 kbps for 5.1 channel) HE-AAC compression where the BER is under 3% while ensuring a high level of imperceptibility (the average ODG was over -1) and data payload (96 bps). Synchronization for our algorithm was also discussed.

In the future, we will study a more efficient and effective watermarking method for multichannel audio to achieve a higher data payload. We will further explore a suitable error correction code and synchronization to the method to develop a real-time application.

Author Contributions: Both authors contributed to the research work. Both authors designed the new method and planned the experiments. J.K. led and reviewed the research work. Y.H. performed the experiments and wrote the paper.

Funding: This research was supported by a 2018 Research Grant from Sangmyung University.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Herre, J.; Dietz, M. MPEG-4 high-efficiency AAC coding [Standards in a Nutshell]. *IEEE Signal. Process. Mag.* **2008**, *25*, 137–142. [\[CrossRef\]](#)
- Wolters, M.; Kjorling, K.; Himm, D.; Purnhagen, H. *A Closer Look into MPEG-4 High Efficiency AAC*; Audio Engineering Society: New York, NY, USA, 2003.
- Meltzer, S.; Moser, G. *MPEG-4 HE-AAC v2—Audio Coding for Today's Digital Media World*; EBU Technical Review; EBU: Geneva, Switzerland, 2006; pp. 37–48.
- Shin, D.; Hong, Y.; Kim, J.; Choi, J. Audio Blind Watermarking Robust Against HE-AAC. In Proceedings of the 8th International Conference on Signal Processing Systems, Auckland, New Zealand, 21–24 November 2016; ACM: New York, NY, USA, 2016; pp. 114–118.
- Cox, I.; Miller, M.; Bloom, J.; Fridrich, J.; Kalker, T. *Digital Watermarking and Steganography*, 2nd ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2008; ISBN 978-0-08-055580-5.
- Barni, M. What is the future for watermarking? (Part II). *IEEE Signal Process. Mag.* **2003**, *20*, 53–59. [\[CrossRef\]](#)
- Katzenbeisser, S. *Information Hiding Techniques for Steganography and Digital Watermarking*, 1st ed.; Katzenbeisser, S., Petitcolas, F.A., Eds.; Artech House, Inc.: Norwood, MA, USA, 2000; ISBN 978-1-58053-035-4.
- Cox, I.J.; Kilian, J.; Leighton, F.T.; Shamoon, T. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Process.* **1997**, *6*, 1673–1687. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zhang, P.; Xu, S.Z.; Yang, H.Z. Robust audio watermarking based on extended improved spread spectrum with perceptual masking. *Int. J. Fuzzy Syst.* **2012**, *14*, 289–295.
- Xiang, Y.; Natgunanathan, I.; Rong, Y.; Guo, S. Spread Spectrum-Based High Embedding Capacity Watermarking Method for Audio Signals. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 2228–2237. [\[CrossRef\]](#)
- Yang, S.; Tan, W.; Chen, Y.; Ma, W. Quantization—Based Digital Audio Watermarking in Discrete Fourier Transform Domain. *J. Multimed.* **2010**, *5*. [\[CrossRef\]](#)
- Elshazly, A.R.; Fouad, M.M.; Nasr, M.E. Secure and robust high quality DWT domain audio watermarking algorithm with binary image. In Proceedings of the 2012 Seventh International Conference on Computer Engineering Systems (ICCES), Cairo, Egypt, 27–29 November 2012; pp. 207–212.
- Dhar, P.; Shimamura, T. Blind Audio Watermarking in Transform Domain Based on Singular Value Decomposition and Exponential-Log Operations. *Radioengineering* **2017**, *26*, 552–561. [\[CrossRef\]](#)
- Li, J.; Wang, H.X.; Wu, T.; Sun, X.; Qian, Q. Norm ratio-based audio watermarking scheme in DWT domain. *Multimed. Tools Appl.* **2018**, *77*, 14481–14497. [\[CrossRef\]](#)
- Bassia, P.; Pitas, I.; Nikolaidis, N. Robust audio watermarking in the time domain. *IEEE Trans. Multimed.* **2001**, *3*, 232–241. [\[CrossRef\]](#)
- Wen-Nung, L.; Li-Chun, C. Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification. *IEEE Trans. Multimed.* **2006**, *8*, 46–59. [\[CrossRef\]](#)
- Megías, D.; Serra-Ruiz, J.; Fallahpour, M. Efficient Self-synchronised Blind Audio Watermarking System Based on Time Domain and FFT Amplitude Modification. *Signal Process.* **2010**, *90*, 3078–3092. [\[CrossRef\]](#)

18. Zeng, G.R.; Qiu, Z.D. Audio watermarking in DCT: Embedding strategy and algorithm. In Proceedings of the 2008 9th International Conference on Signal Processing, Beijing, China, 26–29 October 2008; pp. 2193–2196.
19. Chen, S.T.; Huang, H.N.; Chen, C.J.; Wu, G.D. Energy-proportion based scheme for audio watermarking. *IET Signal Process.* **2010**, *4*, 576–587. [[CrossRef](#)]
20. Petrovic, R. Audio signal watermarking based on replica modulation. In Proceedings of the 5th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service, TELSIKS 2001–Proceedings of Papers (Cat. No. 01EX517), Nis, Yugoslavia, 19–21 September 2001; Volume 1, pp. 227–234.
21. Petrovic, R.; Winograd, J.M.; Jemili, K.; Metois, E. Data hiding within audio signals. In Proceedings of the 4th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services. TELSIKS'99 (Cat. No. 99EX365), Nis, Yugoslavia, 15 October 1999; Volume 1, pp. 88–95.
22. Muhaimin, H.; Danudirdjo, D.; Suksmono, A.B.; Shin, D. An efficient audio watermark by autocorrelation methods. In Proceedings of the 2015 International Conference on Electrical Engineering and Informatics (ICEEI), Denpasar, Indonesia, 10–11 August 2015; pp. 606–611.
23. Lei, B.Y.; Soon, I.Y.; Li, Z. Blind and robust audio watermarking scheme based on SVD–DCT. *Signal Process.* **2011**, *91*, 1973–1984. [[CrossRef](#)]
24. Khalil, M.; Adib, A. Audio watermarking with high embedding capacity based on multiple access techniques. *Digit. Signal Process.* **2014**, *34*, 116–125. [[CrossRef](#)]
25. ITU-R BS.1387-1. *Method for Objective Measurements of Perceived Audio Quality*; ITU: Geneva, Switzerland, 2001.
26. Kabal, P. *An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality*, TSP Lab. Technical Report; Department of Electrical and Computer Engineering, McGill University: Montreal, QC, Canada, 2002.
27. Fdkaac.exe. Available online: <https://www.dbpoweramp.com/codec-central-m4a.htm> (accessed on 19 May 2019).
28. Ffmpeg.exe. Available online: <https://ffmpeg.zeranoe.com/builds> (accessed on 19 May 2019).

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).