Research Article

Audio Watermarking Algorithm with a Synchronization Mechanism Based on Spectrum Distribution

Qiuling Wu,¹ Runyu Ding,¹ and Jiangchun Wei²

¹School of Cyber Security, Jinling Institute of Technology, Nanjing 211169, China
²Jiangsu Liyuan Power Engineering Limited Company, Nanjing 211169, China

Correspondence should be addressed to Qiuling Wu; wuqiuling@jit.edu.cn

Received 23 May 2022; Revised 7 August 2022; Accepted 20 August 2022; Published 5 September 2022

Abstract

In order to solve the problem that it is difficult to extract the watermark when the carried audio is subjected to malicious attacks, audio watermarking algorithm based on spectrum distribution is proposed. An eigenvalue is designed to represent the spectrum distribution of the specified frequency-band, and then the binary watermark is embedded by adjusting the difference of the eigenvalues between two adjacent frequency-bands. The polarity of embedding depth is determined by comparing the values of the eigenvalues, which will minimize the modification about the audio so as to improve transparency. The binary watermark can be extracted blindly by judging the difference of the eigenvalues. The improved synchronization mechanism takes the small frame with the largest energy in the voiced frame as the synchronization mark to search the embedding and extracting location of the watermark, which makes the algorithm have strong robustness. Experimental results show that the proposed algorithm has large payload capacity, good transparency, low complexity, and strong robustness against most attacks.

1. Introduction

1.1. Related Works. The rapid development of Internet and computer technology has provided great convenience for people to spread various multimedia resources on the network, but the copyright protection of these multimedia resources has become more and more concerned with people. Digital watermarking technology plays an important role in solving copyright protection problems [1–3]. It uses a specific embedding algorithm to conceal a watermark that can prove the author’s copyright about multimedia resources. When a copyright dispute occurs, the author extracts the watermark from multimedia resources by using the extracting algorithm corresponding to the embedding algorithm to prove his ownership. Audio digital watermarking technology refers to a kind of security protection technology that embeds the watermark into the audio secretly without much impacting on the audio quality [4], so as to achieve the purposes of copyright tracking, integrity protection, content authentication, and recovery advertising timing.

According to different application purposes, audio digital watermarking can be divided into robust watermarking and fragile watermarking. Robust audio watermarking technology has strong robustness. When the carried audio is subjected to malicious attacks, it can still extract the watermark with a very small bit error rate (BER), so it is mainly used for audio copyright protection [5, 6]. Fragile audio watermarking technology means that the watermark is very sensitive to malicious attacks. It can accurately locate and even recover the tampered region of the audio, so it is mainly used for audio integrity protection [7, 8]. Most robust watermarking algorithms are developed in frequency-domain, and the signal processing algorithms mainly include discrete Fourier transform (DFT) [9], discrete cosines transform (DCT) [10, 11], and discrete wavelets transform (DWT) [12–15]. Megias et al. [16] presented a blind audio watermarking algorithm in the DFT domain for overcoming the synchronization attack. The algorithm embedded the watermark in the frequency domain and marked the embedding location in the time domain. However, the synchronization mark in the time domain was
vulnerable to attacks that may lead to the failure in the extracting process because the watermark's location could not be searched. Tewari et al. [17] proposed an audio watermarking algorithm in the DCT domain. The algorithm selected the appropriate audio frame which would be processed by DCT according to the minimum energy threshold, and it embedded the watermark by modifying and quantifying the average energy of the DCT coefficient. This algorithm had strong robustness to MP3 attacks, but its robustness to other attacks needed to be improved. By introducing the spectrum shaping technology in autoregressive model into vector modulation, Hu and Hsu [18] proposed a blind audio watermarking algorithm in the DWT domain. The embedding depth of the algorithm was consistent with the auditory masking threshold and had high embedding capacity, but its robustness was poor. In order to extract the watermark blindly, many scholars use quantization index modulation (QIM) to design audio watermarking algorithm. Hwang et al. [2] presented an audio watermarking algorithm based on QIM and singular value decomposition (SVD). The algorithm directly applied SVD to stereo signal and embedded the watermark according to the ratio of the singular value. It had good robustness against signal processing operations, such as amplitude scaling, compression, and resampling.

In order to improve the robustness of the algorithm, many scholars design water-marking algorithms in a hybrid domain. Merrad and Saadi [19] proposed an audio robust watermarking technology in DWT and DCT hybrid domain according to the characteristics of strong correlation between two continuous samples. The algorithm had good robustness against random cropping, echo addition, amplitude scaling, and so on. Q. L. Wu and M. Wu [20] proposed an audio watermarking algorithm in the DWT-DCT domain. This algorithm embedded binary watermark into audio by modifying the average amplitude of the hybrid transformed coefficients according to the redundancy of the human auditory system. From the experimental results, it had good robustness against conventional signal processing attacks. However, it could not resist synchronization attacks, such as jittering and time scale modification (TSM), because it lacked synchronization mechanism. In a word, with the continuous development of audio watermarking research, scholars have developed more and more watermarking algorithms with different performance. However, most audio watermarking algorithms are robust against conventional signal processing operations, but their resistance to synchronization attacks needs to be improved, mainly because most algorithms lack effective synchronization mechanisms. Synchronization attack will seriously damage the structure of audio data, which will make it difficult to determine the exacting location of the watermark in the audio, resulting in the extraction failure. Therefore, how to overcome synchronization attacks has become a very challenging problem in watermarking research. The most common attack types are random clipping, jittering, TSM, and pitch shifting modification (PSM). TSM adjusts the playing time of the audio by changing its playing speed, that is, the overall playing time is compressed or expanded, while the sample rate is almost unchanged. PSM usually modifies the tone of the audio without changing its playback speed. Jittering means deleting or adding one sample every several samples in the audio. Random clipping refers to randomly cutting out several samples in the different parts of the audio. Jiang et al. [21] proposed an audio watermarking algorithm that indirectly realizes synchronization by using the audio frame sequence number based on the global characteristics of audio frames. This scheme is robust against most of signal processing operations and partial synchronization attacks. Liu et al. [22] proposed an audio watermarking algorithm to resist the synchronization attack. The algorithm constructed a logarithmic mean characteristic by using the frequency-domain coefficients and then uses the residuals of the two sets of characteristics to design the watermarking algorithm. The robustness of this algorithm against signal processing operations and partial synchronization attacks was better than most existing watermarking algorithms. Hu et al. [23] proposed an audio watermarking algorithm with synchronization mechanism based on lifting wavelet transform. The algorithm sorted and reconstructed the approximate coefficients to design the embedding and extracting rules according to the expected bit rate of the watermark. It had strong robustness to synchronization attacks, but its transparency was poor. Wu et al. [24] proposed an audio watermarking algorithm with an implicit synchronization mechanism based on SVD and the genetic algorithm (GA). The synchronization mechanism of this algorithm took the sampling point with the largest amplitude in the voiced frame as the synchronization mark to track the location of the watermark in the audio and achieved good results in resisting synchronization attacks. However, the synchronization mechanism used a single sampling point as the synchronization mark, which would lead to inaccuracy in the region where the watermark was located because the sampling point might no longer have the largest amplitude after the audio was attacked. In the research of the synchronization mechanism, the existing synchronization mechanisms mainly include exhaustive search, explicit synchronization [25], implicit synchronization, autocorrelation, and constant watermark. Each synchronization mechanism has its own advantages and disadvantages. It needs to be coordinated with the specific watermark embedding and extracting algorithm in order to give full play to the best performance of the algorithm.

1.2. Contributions. In the above introduction about the audio watermarking algorithm, we have known that the robustness of the algorithm is an important performance on the premise of ensuring good audio quality. After analyzing the characteristics of the audio in the time domain and the frequency domain, this paper proposes a robust watermarking algorithm based on spectrum distribution and improves the synchronization mechanism in reference [24]. The main contributions are as follows:

(1) Propose a blind audio watermarking algorithm based on spectrum distribution. An eigenvalue that reflect the spectrum distribution of the audio in a narrow
frequency-band is constructed, and then the difference of the eigenvalues between two adjacent frequency-bands is modified by adjusting the DCT coefficients. The embedding algorithm and the extracting algorithm are developed according to the corresponding relationship between the binary watermark and the difference of the eigenvalue.

The proposed algorithm has good transparency, strong robustness, and blind extracting ability. In order to obtain good transparency, the polarity of the embedding depth is determined by comparing the values of the eigenvalues in two frequency-bands. The purpose of this is to minimize the modification of the DCT coefficient so as to ensure audio quality. Since the spectrum distribution of the carried audio does not change much under attack, embedding the watermark on the eigenvalue related to the spectral distribution makes this algorithm more robust. In addition, this algorithm can realize blind extraction, which will facilitate the application of this algorithm in practice.

(2) Improve the synchronization mechanism in reference [24]. Since most of the audio content is concentrated on the voiced frame, the synchronization mechanism in reference [24] took the single sampling point with the largest amplitude in the voiced frame as the synchronization mark to design a synchronization mechanism. This method sometimes was inaccurate in determining the region where the watermark was located, mainly because the synchronization mark might not be the original sampling point with the largest amplitude after the audio was attacked. On this basis, the synchronization mechanism needs to be improved. The improved synchronization mechanism divides the voiced frame into many small frames and then uses the small frame with the maximum energy as the synchronization mark and selects a certain number of continuous small frames around it to form the embedding region.

The remainder of this paper is organized as follows: In Section 1, we will review related work of audio watermarking algorithms in recent years and then introduce our contributions in this paper. Section 2 describes the improved synchronization mechanism and shows the implementation steps in detail. Section 3 elaborates the principle of the proposed audio watermarking algorithm, and this section will be divided into two parts, including the procedure for embedding the watermark and for extracting the watermark. In Section 4, the implementation steps of the embedding algorithm and the extracting algorithm are set out in detail. Section 5 assesses the performance of this proposed algorithm and compares their performance with three related algorithms. Finally, Section 6 draws up the conclusion and gives the future research plan.

2. Synchronization Mechanism

Synchronization attack will seriously damage the structure of the audio data, which will lead to the extracting algorithm unable to accurately find the location of watermark in audio, so it is a very challenging type of attack [26, 27]. Therefore, it is particularly important to design a synchronization mechanism that can accurately search the location of the watermark in the audio. On the basis of the synchronization mechanism proposed in reference [24], it is improved to make the synchronization mechanism have a better performance. This method takes the small frame with the largest energy in a voiced frame as the synchronization mark and takes the audio data in a fixed region around this mark as the carrier for carrying the watermark. The specific steps are described as follows:

Step 1: Convert the watermark $W = \{w(q), 1 \leq q \leq L_w\}$ into a matrix with $L_1$ rows and $L_2$ columns. $L_w$ is the length of the watermark, and $L_w = (L_1 \times L_2)$ . $w(q)$ is the bit value, and $w(q) \in \{0, 1\}$.

Step 2: Divide the audio into $L_1$ fragments and select the voiced frames with the largest energy from each audio fragment to carry the watermark, and the length of each voiced frame is $N_1$.

Step 3: divide each voiced frame into small frames with $N_2$ sample-points and calculate the energy of each small frame.

Step 4: take the small frame with the largest energy as the synchronization mark and select $(a_1 + a_2 + 1)$ small frames around it as the region to be used to carry the watermark.

According to the specific position of the synchronization mark in the voiced frame, it can be divided into the following three cases.

(1) If the number of small frames before the synchronization mark is less than $a_1$, it indicates that the synchronization mark is closer to the head of the voiced frame, and the watermark location should include $(a_1 + a_2 + 1)$ consecutive small frames starting from the first small frame.

(2) If the number of small frames after the synchronization mark is less than $a_2$, it indicates that the synchronization mark is closer to the end of the voiced frame, and the watermark location should include $(a_1 + a_2 + 1)$ consecutive small frames at the end.

(3) In addition to the above two cases, the location where the watermark is located takes the synchronization mark as the benchmark and selects $a_1$ small frames forward and $a_2$ small frames backward.

After the above steps, the selected audio fragment with $(a_1 + a_2 + 1)$ consecutive small frames is used to carry the watermark.
3. Principle of the Watermarking Algorithm

In this section, audio watermarking algorithm with a synchronization mechanism is developed. The proposed algorithm uses the improved synchronization mechanism to identify the embedding and extracting locations of the watermark. The embedding and extracting rules are developed based on spectrum distribution. It not only resists conventional signal processing attacks but also has excellent performance in resisting synchronization attacks.

3.1. Principle of Embedding Watermarks. DCT has strong “energy concentration” characteristics and good decorrelation, so it has been widely used in the field of image and audio signal processing. Suppose that \( x(n) \) is the original audio with \( N \) sample-points, and it can be expressed as the following formula:

\[
x(n) = \{a_n, 1 \leq n \leq N\},
\]

where \( a_n \) is the amplitude of the sample-point. Dividing \( x(n) \) into \( L \) audio fragments and obtain the voiced frame with the largest energy with \( N_1 \) sample-points in each audio fragment. Use the improved synchronization mechanism in Section 2 to obtain the audio data \( x_1(n_1) = \{a_{n_1}, 1 \leq n_1 \leq N_2 \times (a_1 + a_2 + 1)\} \) for carrying the watermark, where \( n_1 \) is the position number of the audio data in the embedding region. Apply DCT on \( x_1(n_1) \), as shown in the formulas (2) and (3).

\[
X_c(0) = \frac{1}{\sqrt{N_2 \times (a_1 + a_2 + 1)}} \sum_{n_i=0}^{N_2 \times (a_1 + a_2 + 1)-1} x_1(n_1),
\]

\[
X_c(k) = \frac{2}{\sqrt{N_2 \times (a_1 + a_2 + 1)}} \sum_{n_i=0}^{N_2 \times (a_1 + a_2 + 1)-1} x_1(n_1) \cos \left( \frac{2\pi}{2N_2 \times (a_1 + a_2 + 1)} \right)(k \neq 0),
\]

where \( X_c(0) \) is the component that its frequency is 0 Hz, and \( X_c(k) \) is the \( k \)th harmonic component. The frequency \( f_i \) of each harmonic component can be calculated by the following formula:

\[
f_k = \frac{k f_s}{2N_2 \times (a_1 + a_2 + 1)} (k \neq 0),
\]

where \( f_s \) is the sampling frequency. The spectrum of the audio can be used to describe the proportion of each harmonic component in audio, which is related to the frequency and amplitude of harmonics. Divide \( X_c(k) \) into \( L \) fragments to obtain the frequency-bands \( X_i(i) \) \( (1 \leq i \leq L, 1 \leq i \leq N_3) \). Each frequency-band contains \( N_3 \) spectrum lines, and \( L = N_3 \times (a_1 + a_2 + 1)/N_3 \). \( (L > 2L_2) \). The spectrum distribution function (SD) shown in the formula (5) is designed to represent the spectrum distribution of the frequency-band \( X_i(i) \).

\[
SD_i = \sum_{i=1}^{N_2} |X_i(i)| \left( f_i - f_j \right),
\]

where \( f_i \) is the initial frequency of the \( i \)th frequency-band, \( f_j \) is the frequency of the \( j \)th harmonic component, and \( i = (k - LN_3) \). After the audio is attacked, it is assumed that the variation of the DCT coefficient is \( \Delta_i \). Then, the new spectrum distribution \( SD'_i \) can be shown in the following formula:

\[
SD'_i = SD_i + \sum_{i=1}^{N_2} \Delta_i f_s \frac{L_2}{2N_2 \times (a_1 + a_2 + 1)}.
\]

Since \( \Delta_i \) is very small, and \( N_2 \times (a_1 + a_2 + 1) \) is much larger than \( \sum_{i=1}^{N_2} \Delta_i f_s \), so \( SD_i \approx SD'_i \). That is, when the audio is attacked, SD only changes very little.

The following experiment can be used to verify this deduction. The experimental parameters are as follows: \( N_2 = 20, a_1 = 555, a_2 = 1500, f_s = 44.1 \text{kHz} \), and \( N_3 = 200 \). Four attack types are applied to the tested audio, respectively, including amplitude scaling, noise corruption with 20 dB, low-pass filtering with 4 kHz, and TSM with +5%. The four curves under attacks in Figure 1 are basically consistent with the original curve, which indicates that SD can be used to express the stability of the audio spectrum structure after being attacked. The experimental results are also consistent with the deduction from the formula (6). It can be seen that the spectrum structure has good stability after the audio is subjected to attacks, the watermark algorithm can be developed according to this feature.
According to the formula (5), the spectrum distributions of $X_l(t)$ and $X_{l+1}(t)$ can be calculated as $SD_l$ and $SD_{l+1}$, then the average value can be expressed in the following formula:

$$SD_M = \frac{(SD_l + SD_{l+1})}{2} \tag{7}$$

The modified coefficients $X'_l(i)$ and $X'_{l+1}(i)$ can be expressed as the formulas (8) and (9).

$$X'_l(i) = X_l(i) \times \frac{(1 + \lambda_1)SD_M}{SD_l} \tag{8}$$

$$X'_{l+1}(i) = X_{l+1}(i) \times \frac{(1 - \lambda_1)SD_M}{SD_{l+1}} \tag{9}$$

According to formula (5), the spectrum distributions of $X'_l(i)$ can be calculated as $SD'_l$, as shown in the following formula:

$$SD'_l = \sum_{i=1}^{N_l} |X'_l(i)| (f_i - f_l)$$

$$= \sum_{i=1}^{N_l} X_l(i) \times \frac{(1 + \lambda_1)SD_M}{SD_l} (f_i - f_l)$$

$$= \frac{(1 + \lambda_1)SD_M}{SD_l} \sum_{i=1}^{N_l} |X_l(i)| (f_i - f_l) \tag{10}$$

$$= \frac{(1 + \lambda_1)SD_M}{SD_l} SD_l$$

$$= (1 + \lambda_1)SD_M.$$  

Similarly, the spectrum distribution of $X'_{l+1}(i)$ can be expressed as $SD'_{l+1} = (1 - \lambda_1)SD_M$.

The difference of spectrum distribution in two adjacent frequency-bands can be described in the following formula:

$$\nabla SD' = |SD'_l - SD'_{l+1}|$$

$$= 2\lambda_1 SD_M, \tag{11}$$

where $\lambda_1$ is the embedding depth, and $\lambda_1 \in (-1, 1)$. $\lambda_1$ can be set according to the following formula:

$$\begin{cases} 
\lambda_1 = 0, & \text{if } w(q) = 0, \\
\lambda_1 > 0, & \text{if } w(q) = 1 \text{ and } SD_l > SD_{l+1}, \tag{12} \\
\lambda_1 < 0, & \text{if } w(q) = 1 \text{ and } SD_l < SD_{l+1}.
\end{cases}$$

In order to prevent the serious degradation of audio quality caused by too much modification of the DCT coefficient, a threshold $\lambda_2$ is set to judge whether the watermark can be embedded in the two adjacent frequency-bands. If $|SD_l - SD_{l+1}| \lambda_2 SD_M$, it indicates that two frequency-bands need be set as invalid and no watermark bit can be embedded in them. Otherwise, modify the DCT coefficients according to the following embedding rules.

1. If $w(q) = 1$, set $\lambda_1 \neq 0$, then $SD'_l = (1 + \lambda_1)SD_M$, $SD'_{l+1} = (1 - \lambda_1)SD_M$, and $\nabla SD' = 2\lambda_1 SD_M \neq 0$. In order to improve the transparency of the algorithm, the polarity of $\lambda_1$ must be adjusted according to $SD_l$ and $SD_{l+1}$, as shown in formula (12).
2. If $w(q) = 0$, set $\lambda_1 = 0$, then $SD'_l = SD'_{l+1}$, and $\nabla SD' = 0$.

3.2. Principle of Extracting Watermarks. In the embedding process, the binary watermark can be embedded into two adjacent frequency-bands by adjusting the DCT coefficients.
Therefore, the extracting rules is to extract the binary watermark $w'(q)$ by calculating the difference between the spectral distributions of two adjacent bands, which can be expressed by the following formula:

$$w'(q) = \begin{cases} 0, & \forall SD' \langle \lambda_1 SD_M, \\ 1, & \lambda_1 SD_M \leq \forall SD' \langle 3 \lambda_1 SD_M, 1 \leq q \leq L_w, \\ \text{invalid, others}, & \end{cases}$$  

(13)

4. Implementation of the Proposed Algorithm

4.1. Procedure for the Embedding Watermark. Figure 2 shows the embedding diagram, and the detailed embedding steps are described as follows:

Step 1: Convert the watermark $w(q)$ into a binary matrix of $(L_2 \times L_2)$

Step 2: Divide the original audio into $L_1$ fragments. In each fragment, the voiced frame with the largest energy is selected to carry the watermark, and the length of the voiced frame is $N_1$.

Step 3: Divide each voiced frame into small frames with $N_2$ sample points and select the small frame with the largest energy as the synchronization mark.

Step 4: Select $(a_1 + a_2 + 1)$ consecutive small frames around the synchronization mark as the embedding region of the watermark. See Step 4 in Section 2 for details.

Step 5: Apply DCT on $x'_c(n_1)$ to obtain DCT coefficients $X'_c(k)$.

Step 6: Starting from the $b_0$ spectrum line, select $L$ frequency-bands from $X'_c(k)$, each frequency-band has $N_3$ spectrum lines.

Step 7: Calculate $SD_{b_1}, SD_{b_1+1}$ and $SD_M$ of two adjacent frequency-bands.

Step 8: If $|SD_{b_1} - SD_{b_1+1}| \lambda_2 SD_M$, set these two frequency-bands as invalid and go to step 7, otherwise, go to step 9.

Step 9: Embed 1 bit watermark by modifying the DCT coefficients of these two frequency-bands according to the formulas (8), (9), and (12).

Step 10: Repeat step 7 to step 9 until $L_2$ bits watermark are embedded into the voiced frame.

Step 11: Apply inverse discrete cosine transform (IDCT) on $X'_c(k)$ to obtain $x'_c(n_1)$.

Step 12: Repeat step 3 to step 11, until all $L_1$ lines of binary watermark are embedded into the original audio.

Step 13: Reconstruct all voiced frames carrying the binary watermark to obtain the carried audio $x'(n)$.

In order to improve the robustness of the algorithm, several voiced frames can be selected in each audio fragment to repeatedly carry the same line of binary watermark. For example, select three voiced frames with the largest energy in the same audio fragment to carry the watermark.

4.2. Procedure for the Extracting Watermark. Figure 3 shows the extracting diagram, and the detailed extracting steps are described as follows:

Step 1: Divide the carried audio $x'(n)$ into $L_1$ fragments. In each fragment, the voiced frame with the largest energy is selected to extract the watermark from it, and the length of the voiced frame is $N_1$.

Step 2: Divide each voiced frame into small frames with $N_2$ sample points and select the small frame with the largest energy as the synchronization mark.

Step 3: Select $(a_1 + a_2 + 1)$ consecutive small frames around the synchronization mark as the extracting region of the watermark. See Step 4 in Section 2 for details.

Step 4: Apply DCT on $x'_c(n_1)$ to obtain DCT coefficients $X'_c(k)$.

Step 5: Starting from the $b_0$ spectrum line, select $L$ frequency-bands from $X'_c(k)$, and each band has $N_3$ spectrum lines.

Step 6: Calculate $SD'_{b_1}, SD'_{b_1+1}$ and $SD_M$ of two adjacent frequency-bands.

Step 7: If $|SD'_{b_1} - SD'_{b_1+1}| \lambda_2 SD_M$, go to step 6, otherwise, go to step 8.

Step 8: Extract 1 bit watermark from these two frequency-bands according to the formula (13).

Step 9: Repeat step 6 to step 8 until $L_2$ bits watermark are extracted from the voiced frame.

Step 10: Repeat step 2 to step 9 until all binary watermark are extracted from the carried audio $x'(n)$.

5. Performance Evaluation

In this section, the performance of the proposed algorithm will be evaluated from four aspects, including payload capacity, transparency, robustness, and complexity. Evaluating transparency is to calculate the decline of audio quality before and after being embedded. It can be measured from both subjective and objective aspects. The subjective index is the mean opinion score (MOS) and the objective index is the signal-to-noise ratio (SNR) which is shown in formula (14), and the object difference grade (ODG) which is the output value of the perceptual evaluation of audio quality (PEAQ). Robustness refers to the characteristic that the algorithm can still extract the
watermark more accurately after the carried audio was attacked by numerous attacks. Robustness can be usually evaluated by BER which refers to the proportion of error bits between the extracted watermark and the original watermark, and BER can be expressed in formula (15). The smaller the BER, the stronger the robustness of the algorithm against attacks. According to the standard of international federation of the phonographic industry (IFPI), SNR should be greater than 20 dB so as to make the audio have good quality, and BER should be less than 20%.

\[
\text{SNR} (x(n), x'(n)) = 10 \log \left( \frac{\sum_{i=1}^{E} x^2(n)}{\sum_{n=1}^{N} (x'(n) - x(n))^2} \right), \quad (14)
\]

\[
\text{BER} = \frac{\sum_{q=1}^{L_w} w(q) \oplus w'(q)}{L_w} \times 100\%. \quad (15)
\]
Normalized correlation (NC) also can be used to evaluate the robustness of the algorithm. It reflects the robustness by evaluating the similarity between the extracted watermark and the original watermark, as shown in formula (16). If the value of NC is close to 1, it indicates that the extracted watermark is very similar to the original watermark, that is, the robustness of the algorithm is strong.

\[
NC = \frac{\sum_{q=1}^{L_1} w(q) \times w'(q)}{\sqrt{\sum_{q=1}^{L_1} w(q)^2 \sum_{q=1}^{L_1} w'(q)^2}}
\]  

(16)

The experimental parameters are as follows: (1) The watermarks are two binary images shown in Figure 4(a) with the size of \((64 \times 64)\) and Figure 4(b) with the size of \((128 \times 32)\); (2) The tested audio are twenty 64-second audio signals which were formatted by WAV and sampled at 44100 Hz with 16-bit resolution, all of which come from TIMIT standard database; (3) Algorithm parameters: \(L_1 = 64\), \(L_2 = 64\), \(N_2 = 8\), \(N_3 = 4\), \(a_1 = 168\), \(a_2 = 343\), \(b_0 = 350\), \(|\lambda_1| = 0.3\), and \(\lambda_2 = 1.5\).

The experimental equipment and application software are described as follows: (1) Computer system: 64-bit
5.1. Capacity and Transparency. Formula (17) can be used to calculate the payload capacity of the proposed algorithm.

\[
\text{Cap} = \frac{L_w}{t} = \frac{(L_1 \times L_0)}{t} \quad (17)
\]

Microsoft Windows 10; (2) Software for processing audio: Cool Edit Pro; (3) Programming Language: Matlab 2016R.

Table 1: Average experimental results (no attack).

<table>
<thead>
<tr>
<th>Items</th>
<th>SNR</th>
<th>MOS</th>
<th>ODG</th>
<th>BER (%)</th>
<th>NC</th>
<th>Cap (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>26.86</td>
<td>4.7</td>
<td>−0.45</td>
<td>0.00</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>Reference [13]</td>
<td>21.20</td>
<td>4.1</td>
<td>−1.22</td>
<td>0.00</td>
<td>1</td>
<td>86.20</td>
</tr>
<tr>
<td>Reference [23]</td>
<td>21.08</td>
<td>4.6</td>
<td>−0.39</td>
<td>0.00</td>
<td>1</td>
<td>43.07</td>
</tr>
<tr>
<td>Reference [24]</td>
<td>25.96</td>
<td>4.5</td>
<td>−0.99</td>
<td>0.00</td>
<td>1</td>
<td>64</td>
</tr>
</tbody>
</table>

Figure 4: Two binary watermarks: (a) the first watermark is an image with dimensions of 64 × 64; (b) the second watermark is an image with dimensions of 128 × 32.

Figure 5: Waveform comparison. (a) The original audio. (b) The carried audio.

Figure 6: Spectrogram comparison. (a) The original audio. (b) The carried audio.
where \( t \) is the duration of the audio used to carry the watermark. In our experiment, \( t \) is 64 seconds, and the binary watermark is 4096 bits, so the payload capacity is \( 4096 \text{bit} / 64 \text{s} = 64 \text{bps} \). Table 1 shows the experimental results, including transparency, which is evaluated by SNR (dB), MOS, ODG of the audio, robustness which is evaluated by BER (%) and NC of the extracted watermark, and the payload capacity which is expressed as \( \text{Cap} \) (bps).

According to the experimental results in Table 1, the proposed algorithm has good transparency, because under the premise of the payload capacity of 64 bps, the average SNR is as high as 26.86 dB, ODG is \(-0.45\), and MOS is 4.7, which is higher than the standard of IFPI. Good transparency is mainly because the polarity of embedding depth is determined by comparing the SD of the two adjacent frequency-bands as shown in formula (12), which will minimize the modification of DCT coefficients. The waveforms of the original audio and the carried audio are shown in Figures 5(a) and 5(b), respectively (only show an audio clip about 3 seconds so as to display the details of the audio). The corresponding spectrograms are shown in Figures 6(a) and 6(b). It can be seen that the waveforms and spectrograms of the audio before and after embedding the watermark both have no obvious difference, which indicates that the proposed algorithm has good transparency.

In Table 1, BER is equal to 0 and NC is equal to 1, which indicates that when there is no attack applied on the carried audio, the watermark can be extracted accurately. Therefore, the watermark image extracted in Figure 7(w) is the same as the original image in Figure 4(a), and the watermark image extracted in Figure 8(w) is the same as the original image in Figure 4(b).

The payload capacity of the proposed algorithm is lower than that in reference [13], but has better transparency. The payload capacity and transparency of the proposed algorithm are both better than those in reference [23]. Under the same payload capacity, the transparency of the proposed algorithm is higher than that in reference [24].

### 5.2. Robustness

This section will evaluate the robustness of the proposed algorithm. Carry out different attacks on the carried audio and then use the extracting algorithm to extract the watermark. Finally, compare the extracted watermark with the original watermark and quantitatively evaluate the robustness of the algorithm by BER and NC. Attack types include a variety of conventional signal processing operations and synchronous attacks, as shown in Table 2.

After the above attacks on the carried audio, the average values of BER (%) and NC calculated under the same type of attacks are listed in Table 3. The extracted images are shown in Figures 7 and 8.

According to the experimental results shown in Table 3, Figures 7 and 8, the proposed algorithm shows excellent robustness against some attacks, including noise corruption with 35 dB, MP3 compression with 128 kbps, re-quantization, resampling, echo addition with the delay of 100 ms and 50 ms, low-pass filter with 8 kHz, and amplitude scaling. The extracted watermarks are very clear. All BER values are equal to 0, and all NC values are equal to 1.

When it resists noise corruption with 20 dB, low-pass filtering with 4 kHz, MP3 compression with 64 kbps, jittering, and random cropping, it has strong robustness against those attacks. The extracted watermarks are very high similarity with the original watermarks, BER are below 2.98%, and NC values are above 0.99.

When resisting TSM, BER values are relatively high, but they meet the standard of IFPI, and the main extracted...
information of the watermarks can be distinguished from the extracted images. When resisting PSM, BER all are high and the extracted watermarks are very fuzzy. It is difficult to obtain the main information of watermark from the extracted image. Those experimental results show that the proposed algorithm has weak robustness against PSM. The

![Figure 8: The extracted images about the second watermark.](image)

<table>
<thead>
<tr>
<th>Item</th>
<th>Attack type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Noise corruption</td>
<td>Add Gaussian noise with 35 dB into the audio.</td>
</tr>
<tr>
<td>B</td>
<td>Noise corruption</td>
<td>Add Gaussian noise with 20 dB into the audio.</td>
</tr>
<tr>
<td>C</td>
<td>MP3 compression</td>
<td>Apply MP3 compression with 128 kbps.</td>
</tr>
<tr>
<td>D</td>
<td>MP3 compression</td>
<td>Apply MP3 compression with 64 kbps.</td>
</tr>
<tr>
<td>E</td>
<td>Re-quantization</td>
<td>Re-quantize the amplitude with 16-8-16 bit per sample.</td>
</tr>
<tr>
<td>F</td>
<td>Resampling</td>
<td>Change the sampling rate with 44.1-22.05-44.1 kHz.</td>
</tr>
<tr>
<td>G</td>
<td>Echo addition</td>
<td>Add an echo with a delay of 100 ms.</td>
</tr>
<tr>
<td>H</td>
<td>Echo addition</td>
<td>Add an echo with a delay of 50 ms.</td>
</tr>
<tr>
<td>I</td>
<td>Low-pass filtering</td>
<td>Applying low-pass filter with 8 kHz.</td>
</tr>
<tr>
<td>J</td>
<td>Low-pass filtering</td>
<td>Applying low-pass filter with 4 kHz.</td>
</tr>
<tr>
<td>K</td>
<td>Amplitude scaling</td>
<td>Scaling the amplitude by 1.2.</td>
</tr>
<tr>
<td>L</td>
<td>Amplitude scaling</td>
<td>Scaling the amplitude by 0.8.</td>
</tr>
<tr>
<td>M</td>
<td>TSM</td>
<td>Apply TSM with −5% on the audio.</td>
</tr>
<tr>
<td>N</td>
<td>TSM</td>
<td>Apply TSM with +5% on the audio.</td>
</tr>
<tr>
<td>O</td>
<td>TSM</td>
<td>Apply TSM with −10% on the audio.</td>
</tr>
<tr>
<td>P</td>
<td>TSM</td>
<td>Apply TSM with +10% on the audio.</td>
</tr>
<tr>
<td>Q</td>
<td>PSM</td>
<td>Apply PSM with −1% on the audio.</td>
</tr>
<tr>
<td>R</td>
<td>PSM</td>
<td>Apply PSM with +1% on the audio.</td>
</tr>
<tr>
<td>S</td>
<td>Jittering</td>
<td>Delete one sample every 500 samples in the audio.</td>
</tr>
<tr>
<td>T</td>
<td>Jittering</td>
<td>Delete one sample every 1000 samples in the audio.</td>
</tr>
<tr>
<td>U</td>
<td>Random cropping</td>
<td>Randomly cut out 100 samples.</td>
</tr>
<tr>
<td>V</td>
<td>Random cropping</td>
<td>Randomly cut out 200 samples.</td>
</tr>
</tbody>
</table>
5.3. Complexity. The complexity of the algorithm usually can be measured by calculating the running time. Too much complexity will limit the scope of application of the algorithm. In reference [24], the average running time for the embedding watermark is 1055.99 seconds, and the average running time for extracting watermark is 0.8544 seconds. In our study, the embedding time and extraction time are greatly shortened. The average running time for the embedding watermark is 1.2612 seconds, and the average running time of the extracting watermark is 0.6012 seconds, so this proposed algorithm has low complexity.

The experimental results show that the algorithm has a good comprehensive performance. Compared with reference [13], the payload capacity of this algorithm is slightly smaller, but it has higher transparency and better robustness against most attacks. Compared with reference [23], this algorithm has a larger payload capacity and better transparency, and its robustness is stronger except for noise corruption with 20 dB, TSM, PSM, and jittering. Under the same payload capacity, the proposed algorithm has lower complexity, better transparency, and stronger robustness against most attacks, especially low-pass filtering with 4 kHz, except for noise corruption with 20 dB, TSM, and PSM than the algorithm in reference [24]. This is mainly because the proposed algorithm designs the embedding algorithm and the extracting algorithm based on the spectrum distribution which has good stability, and the synchronization mechanism is an improvement on the synchronization mechanism proposed in reference [24].

6. Conclusion

The spectrum distribution of low-frequency components of audio will not change much after being attacked. Based on this feature, a robust and blind audio watermarking algorithm based on spectrum distribution is proposed. The proposed algorithm designs an eigenvalue to represent the spectrum distribution of the frequency-band and then develops the embedding algorithm by adjusting the difference of the eigenvalues between two adjacent frequency-bands. The DCT coefficients are modified according to the embedding rules, and the polarity of embedding depth is determined by comparing the eigenvalues of the two frequency-bands, which will minimize the modification of DCT coefficients, so that the algorithm has good transparency. When extracting the watermark, the binary watermark can be extracted blindly only by judging the difference of the eigenvalues. Therefore, the original audio is not needed, which is very convenient for practical application. In this study, the improved synchronization mechanism takes the small frame with the largest amplitude in the voiced frame as the synchronization mark to obtain the embedding and extracting location of the watermark in the audio, so as to improve the robustness of the algorithm. In addition, other measures are taken to improve the robustness of the algorithm, such as embedding the watermark into three voiced frames repeatedly. From the experimental results and the comparison with other similar algorithms, SNR of the proposed algorithm reaches 26.86 dB when the payload capacity is 64 bps, so it has a large payload capacity and good transparency. When the carried audio is attacked by noise corruption, MP3 compression, low-pass filtering, amplitude scaling, TSM, jittering, and random cropping, the extracted watermarks are very similar with the original watermarks, so this algorithm has strong robustness.

Although the proposed algorithm has advantages in transparency, payload capacity, complexity, and robustness against most attacks, it also has some shortcomings, such as the robustness against TSM and PSM that needs to be improved. In future research, we will overcome this and further improve the robustness for resisting in more attack types.

Data Availability

The data are available upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.
Acknowledgments

This research was funded by the National Natural Science Foundation of China, China (Grant no. 11601202), the High-Level Talent Scientific Research Foundation of the Jining Institute of Technology, China (Grant no. jjt-b-201918), Jiangsu University Student Innovation Training Program (Grant no. 202213573082Y), Collaborative Education Project of the Ministry of Education (Grant no. 202102089002), and Deputy general manager of the Science and Technology of Jiangsu Province (Grant no. FZ20220114).

References


