Dual Channel Audio Watermarking Algorithm Based on Embedded Strength Optimization

Yifan Luo, Dezhong Peng

Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China
30424343@qq.com

Abstract. This work proposes a new dual-channel audio watermarking algorithm to improve the imperceptibility of watermark while maintaining the robustness. First, we use the correlation method to detect the pitch of the origin audio, by which the audio will be segmented adaptively. Then, we implement the discrete wavelet transform (DWT) on the audio signal to get the approximate decomposition coefficients. On this basis, the coefficients are divided into odd-order and even-order channels by the coefficient serial number. According to the high similarity between the two channels, the coefficients of the even-order channel are embedded into the odd-order channel in the form of an echo. To improve the imperceptibility of the watermark, we apply the cross-correlation method to optimize the embedding strength to minimize the signal-to-noise ratio (SNR). By this way, the robustness of the watermark will be improved without significantly affecting the original audio quality. In the watermark extraction stage, the coefficients cross-correlation between the embedding channel and the embedded channel is used to detect the signal peaks. According to the peak positions, the watermark can be extracted, so the blind detection is realized. We test the proposed method by simulation experiments, and the experimental data show that the robustness of the proposed method is higher than the method proposed in [1].

1. Introduction
In recent years, an increasing number of audio watermarking methods have been proposed for music copyright protection. There are two main categories, the time domain embedding [2-6] and frequency domain embedding [7-11]. For the time domain embedding, the watermark is vulnerable to various attacks such as synchronization, signal amplitude change, and frame average attacks. On the other hand, embedding in the frequency domain can improve the robustness of anti-attacks. However, the frequency domain embedding cannot take advantage of the time domain characteristics of the audio signal, and there is also a problem of computational complexity. To solve the problems of the two categories of methods, the methods that embedding in the wavelet domain were designed to make use of both the time and frequency domain features, and became a typical embedding strategy. [12] proposed a self-synchronizing audio watermarking method, which used the discrete wavelet transform (DWT) feature as the marker of synchronization information, and embedded both the synchronization code and the watermark into the DWT low-frequency sub-band. [13] proposed a DWT-based audio watermarking method to resist frame attacks. Xiang et al. [14] proposed a DWT-based audio watermarking method, the original audio was divided into groups, and then the watermark was embedded by using the relationship between the energies of low-frequency sub-band coefficients in different groups, and the embedding strengths were varied adaptively for different groups.
Watermark embedding in the time-frequency domain is more robust than embedding in the time domain or frequency domain. To further improve the imperceptibility of the watermark, [1] applied the Karush-Kuhn-Tucker theory (KKT) method to minimize the difference between the original DWT decomposition coefficients and the coefficients after embedding.

In DWT, the wavelet is gotten by the wavelet function. The function is controlled by the scale parameter and the translation parameter. The discrete scale normalization function and the wavelet basis function are defined as:

\[ \varphi_{j,k}(t) = 2^{j/2} h_j \varphi(2^j t - k) \]

\[ \psi_{j,k}(t) = 2^{j/2} g_j \psi(2^j t - k) \]

where, \( j \) and \( k \) are scale parameters and translation parameters, respectively. \( g_j \) and \( h_j \) are low pass filtering and high pass filtering, respectively. The scaling function and the wavelet function are orthogonal to each other.

In the method proposed in [1], the synchronization code and the watermark were simultaneously embedded into the DWT low-frequency sub-band coefficients.

| ... | Synchronization code | Watermark | Synchronization code | Watermark | ... |
|-----|----------------------|-----------|----------------------|-----------|-----|

The original audio was segmented to parts, and the synchronization codes were embedded to mark the embedding position. The audio signal was further decomposed by DWT to get the decomposition coefficients. Then, the coefficients at the lowest frequency in each segment were divided into several consecutive groups, and each group was further divided into three parts that did not overlap each other. The length of each part was \( N \) coefficients, and each group had \( 3 \times N \) coefficients, which could be expressed as:

\[ E_{i1} = \sum_{j=0}^{N-1} |C_j|, \quad E_{i2} = \sum_{j=N}^{2N-1} |C_j|, \quad E_{i3} = \sum_{j=2N}^{3N-1} |C_j| \]  

(1)

where \( i \) denoted the \( j \)-th group, \( \{C_j\} \) was the low-frequency coefficient, and the coefficients of each three parts were denoted as \( C_{3N} \).

One-bit watermark or synchronization code was embedded in all coefficients of the current group by Eq. (2).

\[
\begin{cases} 
    k_1E_{i1} + k_2E_{i2} + k_3E_{i3} \geq Thd1, \text{ if } w = 1 \\
    k_1E_{i1} + k_2E_{i2} + k_3E_{i3} \leq Thd0, \text{ if } w = 0
\end{cases}
\]  

(2)

where \( k_1, k_2, \) and \( k_3 \) were three weight coefficients, and \( k_1 = k_2 = k_3 = 1 \). \( Thd1 \) and \( Thd0 \) were embedding judgment thresholds, which were determined by Eq. (3) and (4).

\[ Thd1 = MA + \frac{3}{4} (E_{\text{max}} + E_{\text{min}}) \]  

(3)

\[ Thd0 = MA + \frac{1}{4} (E_{\text{max}} + E_{\text{min}}) \]  

(4)

where \( MA \) was the mean of all coefficients in the entire audio signal, \( E_{\text{max}} \) and \( E_{\text{min}} \) were the maximum and minimum of the coefficients, respectively. Then, embedding strength was optimized to improve the imperceptibility of the watermark. However, this method still had room for improvement in two aspects.

(1) The synchronization codes needed to be embedded to record the embedding positions. When a synchronization attacks occur, such as frame cropping attacks, especially when the amount of cropped data exceeds the segment length, not only the synchronization codes will be lost, but also the segmentation of the original audio will be confusing. Therefore, it is necessary to design a more reliable anti-synchronization attacks method.
(2) The watermark was embedded in the lowest frequency coefficients. But the coefficients change did not correlate with the coefficients themselves. When the watermark was extracted, the correlation relationship between the coefficients could not be used to assist the extraction. So, how to use the correlation characteristics between DWT coefficients to optimize the embedding strength and improve the extraction accuracy are essential tasks.

Thus, based on the method in [1], we propose a novel audio watermarking strategy to improve the robustness and imperceptibility by the following two aspects.

(1) The pitch period is used as the audio segmentation standard. The pitch period of the audio signal is detected as the flag for the audio segment by using the autocorrelation method. For one pitch, one-bit watermark data will be embedded into the DWT low-frequency coefficients. While facing synchronization attacks, the pitch period will be less affected, and the correctness of watermark extraction will be improved.

(2) The correlation relationship between the DWT decomposition coefficients is used to assist the watermark extraction. We divide the coefficients into two-channel data: odd-order coefficients channel and even-order coefficients channel. Using the high similarity between the two channels, the coefficients of the even-order channel are cross-embedded into the odd-order channel in the form of echoes. According to the watermark data, different embedding time-delay is selected as the watermark position. In the watermark extraction stage, based on the similarity between the two channels, the cross-correlation peaks of the two channels will be gotten, and the watermark can be extracted according to the positions of the peaks.

2. Pitch period detection
In the past methods, the original audio was segmented by the fixed length. However, the fixed length segmentation will be affected by the synchronization attacks, which will cause the segmentation error in the extraction stage. To solve this problem, we design an adaptive segmentation method. The audio pitch detected by the short-time autocorrelation method will be used as the segmentation flag. The whole detection process is described as the following steps.

(1) Pitch endpoint detection
First, we use the energy-entropy ratio to detect the endpoint [14]. A fast Fourier transform (FFT) is implemented on the j-th audio frame $x_j(n)$ to get frequency components of each spectral line. We express the frequency component $f_k$ of k-th spectral by the energy spectrum, which is set as $Y_j(k)$. The FFT length is set as N frequency components, then the components are normalized. The spectral probability density can be defined in Eq. (5).

$$P_j(k) = \frac{Y_j(k)}{\sum_{l=0}^{N/2} Y_j(l)}$$

Then, the short-time spectral entropy of the audio signal for each frame can be calculated by Eq. (6)

$$H_j = -\sum_{k=0}^{N/2} p_j(k) \log P_j(k)$$

The energy of each frame is:

$$AMP_j = \sum_{m=1}^{N} x_j^2(m)$$

So, the improved energy calculation relation is:

$$LE_j = \log_{10}(1 + \frac{AMP_j}{a})$$

where, $a$ is constant. Thus, the ratio of energy entropy can be calculated by Eq. (9).
\[ EEF_j = \sqrt{1 + [EL_j / H_j]} \quad (9) \]

We set the scale factor as \( \beta \) and the threshold as \( Th \) to detect the endpoint. When \( \beta \cdot EEF_j > T_0 \), the frame is determined as the start of the endpoint. On the contrary, when \( \beta \cdot EEF_j \leq T_0 \), the frame is determined as the end of the endpoint. The signal between every two continuous endpoints is defined as a fragment.

(2) Short-time autocorrelation function pitch detection

After completing the endpoint detection, we use the short-term autocorrelation method to detect the pitch in every fragment. We calculate the autocorrelation between two adjacent frames:

\[ R_j(m) = \sum_{s=1}^{L-m} x_{j-1}(s)x_j(s + m) \quad (10) \]

where \( L \) is the frame length. For each frame, the maximum of the autocorrelation function appears at \( R_j(0) \). To avoid the influence of energy on endpoint detection, we normalize the autocorrelation value:

\[ R(m) = \frac{R_j(m)}{R_j(0)} \quad (11) \]

We set two thresholds \( T_1 \) and \( T_2 \). When the correlation function value is higher than \( T_1 \), the frame is determined as the start of the pitch, and when the value is less than \( T_2 \), it is determined as the end. We use the pitch period as the segmentation standard, and the watermark will be embedded in each pitch.

3. Dual channel division

In each segment, for the low-frequency approximation coefficients of the j-th layer decomposition, the first \( N \) coefficients are selected as the new segment, where \( N \) needs to satisfy \( N \mod 2 = 0 \). Each channel has a length of \( n \) coefficients, and the division rules are:

- Odd-order channel: a new sequence of coefficients with odd sequence numbers.
  \[ x_{odd}(n) = \{x(1), x(3), x(5), \ldots, x(N/2 - 1)\} \quad (12) \]

- Even-order channel: a new sequence of coefficients with even sequence numbers.
  \[ x_{even}(n) = \{x(2), x(4), x(6), \ldots, x(N/2)\} \quad (13) \]

Taking the two-layer DWT decomposition as an example, Figure 1 shows the division process. The elements corresponding to the odd-order channel and the even-order channel are adjacent DWT coefficients, and the difference between the elements is small. Therefore, the two channel elements have high similarity. Channel elements can be cross-superimposed between channels to complete watermark embedding. At the same time, using the similarity between channels, the watermark can be extracted by cross-correlation method. The specific embedding and extraction methods are described in sections 4 and 5.

![Fig.1 Dual channels division](image-url)
4. Watermarking embedding

Both the odd-order and even-order channels are equally divided into three parts; each part has $n/3$ coefficients. In each segment, let the time offset corresponding to the N-th coefficient be $T$, as shown in Figure 2, the time offset corresponding to each part is $T/3$, $2T/3$ and $T$.

The coefficients of the first part of the even-order channel are additively embedded into the second or third part of the odd-order channel by echo-embedding form. The embedding rules are:

1. If the embedded watermark is '0': the coefficients of the first part in the even channel are embedded in the second part of the odd-order channel.

2. If the embedded watermark is '1': the coefficients of the first part in the even channel are embedded in the third part of the odd-order channel.

The corresponding embedding formulas are:

1. Embedding watermark '0':
\[
x'_{\text{odd}}(i + \frac{n}{3}) = \alpha x_{\text{even}}(i) + x_{\text{odd}}(i + \frac{n}{3})
\]

2. Embedding watermark '1':
\[
x'_{\text{odd}}(i + \frac{2n}{3}) = \alpha x_{\text{even}}(i) + x_{\text{odd}}(i + \frac{2n}{3})
\]

where $i \in [1, n]$, $\alpha$ is the embedding strength. Fig. 3 shows the embedding method.

After embedding, the two channels are synthesized, and the inverse DWT (IDWT) is implemented on the synthesized coefficients to restore the watermarked audio. The embedding steps can be summarized as follows and shown in Figure 4.

1. Detect the pitch of the original audio, and segment the audio signal by every pitch period.

2. Implement DWT on each segment to get the first $N$ decomposition coefficients of the low-frequency.

3. Divide the selected coefficients into the odd-order channel and even-order channel.

4. Divide each channel into three equal length parts.

5. The first part of the even-order channel is additively embedded in the second or third part of the odd-order channel according to the embedded watermark.

6. Synthesize the coefficients of the two channels, and IDWT is implemented on the synthetic coefficients to restore the final watermarked audio.

5. Watermark extraction

The first half process of the watermark extraction is the same as the embedding process. The pitch period detection is implemented on the watermarked audio, and the detection result is used to segment the audio signal. The j-layer DWT decomposition is implemented on each segment, in which the first $N$ coefficients of the low-frequency are selected as the watermark extraction region. The coefficients in the region are divided into odd and even order channels. Each channel has $n$ coefficients and will be further divided into three equal length parts with $n/3$ coefficients. We can get the channel coefficient.
difference sequence by subtracting the coefficients of the even-order channel from the odd-order channel.

\[ x_{	ext{sub}}(i) = x_{	ext{odd}}(i) - x_{	ext{even}}(i) \]  

where \( i \in [1, N] \). Because of the similarity between the channels, the difference sequence and the even-order channel have high similarity. So, we can use the correlation between the first part coefficients of the even-order channel and the difference sequence to judge the positions of the watermark according to the correlation peaks.

\[ x_{	ext{sub}}(i) = x_{	ext{odd}}(i) - x_{	ext{even}}(i) \]  

Let \( t_{\text{max}} \) be the time offset when \( R(t) \) getting the maximum value; the watermark will be extracted according to the following criteria.

\[
\begin{align*}
\text{The extracted watermark is '0' if } & |t_{\text{max}} - \frac{T}{3}| < |t_{\text{max}} - \frac{2T}{3}|, \\
\text{The extracted watermark is '1' if } & |t_{\text{max}} - \frac{T}{3}| > |t_{\text{max}} - \frac{2T}{3}|
\end{align*}
\]  

The watermark extraction process is shown in Figure 5.

6. Watermark embedding strength optimization

We need to optimize the embedding strength so that the watermark can achieve an imperceptible optimal state while maintaining the robustness. In this section, we take embedding watermark '0' as the example to illustrate the process of optimization. The Karush-Kuhn-Tucker (KKT) theorem [15] is applied as the optimization method.

The measure index of the imperceptibility is the signal-to-noise ratio (SNR).

\[ \text{SNR} = -10 \log_{10} \left( \frac{\|X_{\text{odd}} - \tilde{X}_{\text{odd}}\|_2^2}{\|X_{\text{odd}}\|_2^2} \right) \]

where \( X_{\text{odd}} = [x_{\text{odd}}(\frac{n}{3}+1), x_{\text{odd}}(\frac{n}{3}+2), \ldots, x_{\text{odd}}(\frac{2n}{3}-1)]^T \) is the original coefficients vector, \( \tilde{X}_{\text{odd}} \) is the coefficients vector after embedding.
According to the embedding method, the cross-correlation value of the two channels in Eq. (11) must satisfy Eq. (19).

\[ g(X_{\text{even}}) = R(t) - R(2t) \geq 0 \]  

(19)

where \( X_{\text{even}} \) is the even-order coefficients vector, \( X_{\text{even}} = [x_{\text{even}}(\frac{n}{3}+1), x_{\text{even}}(\frac{n}{3}+2), ..., x_{\text{even}}(\frac{2n}{3}-1)]^T \).

So, optimization goals can be simplified to Eq. (20).

\[
\text{minimize } f(X_{\text{odd}})
\]

subject to \( g(X_{\text{even}}) \geq 0 \)

(20)

where, \( f(X_{\text{odd}}) = SNR \). Introducing the scale vector \( \mu \) as the KKT multiplier vector, we can convert Eq. (15) and (16) to the unconstrained optimization.

\[
J(X_{\text{odd}}, \mu) = \frac{(X_{\text{odd}} - X_{\text{odd}})^T(X_{\text{odd}} - X_{\text{odd}})}{X_{\text{odd}}^T X_{\text{odd}}} + \mu[R(t) - R(2t)]
\]

(21)

Thus, the necessary condition that \( J(X_{\text{odd}}, \mu) \) has a minimum value is:

\[
\frac{\partial J}{\partial \mu} = R(t) - R(2t) = 0
\]

(24)

According to the embedding rule \( X_{\text{odd}} - X_{\text{odd}} = aX_{\text{even}}^1 \), \( (X_{\text{even}}^1 = [x_{\text{even}}(1), x_{\text{even}}(2), ..., x_{\text{even}}(\frac{n}{3})]^T) \), we can get the final expression of \( J(X_{\text{odd}}, \mu) \):

\[
J(X_{\text{odd}}, \mu) = (aX_{\text{even}}^1)^T(aX_{\text{even}}^1) + \mu[R(t) - R(2t)]X_{\text{odd}}^T X_{\text{odd}}
\]

(25)

Thus, the necessary conditions for \( J(X_{\text{odd}}, \mu) \) to have a minimum are Eq. (26) and (24).

\[
\frac{\partial J}{\partial a} = 2a(X_{\text{even}}^1)^TX_{\text{even}}^1 + \mu X_{\text{odd}}^T X_{\text{odd}} = 0
\]

(26)
According to Eq. (16), $X_{sub} \approx \alpha X_{even}^3$. For embedding watermark '0', we can get Eq. (28) by substituting $X_{sub}$ into (27).

$$R(t) = \alpha \frac{n}{3} \sum_{i=1}^{i=\frac{n}{3}} [X_{even}^i(i)]^2$$  \hspace{1cm} (28)$$

Then, we can get the expression of $\alpha$ by substituting (28) into (24).

$$\alpha \approx \frac{3R(2t)}{\sum_{i=1}^{i=\frac{n}{3}} [X_{even}^i(i)]^2}$$  \hspace{1cm} (29)$$

The optimized value of $\mu$ will be gotten by substituting (29) into (26).

$$\mu^* = -\frac{6R(2t)(X_{even}^i)^T}{nX_{even}^i X_{odd}^i X_{odd}}$$  \hspace{1cm} (30)$$

Then, the value of $\alpha$ that satisfies the minimized SNR will be gotten.

$$\alpha \approx \frac{3R(2t)(X_{even}^i)^T X_{odd}^i X_{odd}}{nX_{even}^i X_{odd} X_{odd} (X_{even}^i)^T X_{even}^i}$$  \hspace{1cm} (31)$$

Set $\beta_0 = \frac{3(X_{even}^i)^T X_{odd}^i X_{odd}}{X_{even}^i X_{odd} X_{odd} (X_{even}^i)^T X_{even}^i}$, the final expression of $\alpha$ is:

$$\alpha \approx \frac{\beta_0 R(2t)}{n}$$  \hspace{1cm} (32)$$

For the case of embedding '0', the watermark is embedded in the $2t$-delay time, and the parity sequence is weakly correlated. Then, $R(2t) \rightarrow 0$. Thus, we can ensure that $\alpha$ is a small value. The imperceptibility of the watermark will be improved.

7. Experimental result

In this section, the experiment results of the proposed method are compared with the method in [1] to test the effectiveness. We select two kinds of music as test audio samples; one is popular music; the other is piano music. The audio samples are 44.1 kHz, the sample quality is 16 bits/sample, and the duration is 30 seconds. In the experiment, the 7-layer DWT decomposition was implemented on the samples. The experimental evaluation is carried out in two aspects: watermark quality and robustness. The quality is tested by objective evaluation and subjective evaluation. The objective evaluation index is SNR, and the subjective evaluation index is Subjective Difference Grade (SDG). The robustness evaluation is to tested by the synchronization attacks, resampling attacks, MP3 compression attacks, low-pass filtering attacks, energy amplitude change attacks, echo attacks and Gaussian noise attacks, and the evaluation index is Bit Error Rate (BER).

7.1. Watermark quality

Table 1 lists the SNR and SDG values after watermark embedding. For the subjective evaluation, we invite 20 listeners to participate in the experiment. The audiences are ten males and ten females, aged between 18 and 30 years old. The test audio clips are divided into two groups: Group A is the original audio clip and Group B is the corresponding watermarked audio clip. All audiences listen to the audio in Group A and B and judge the differences. Compared with the method in [1], the proposed method has higher SNR values, and the SDG values are close to zero.

7.2. Watermark robustness

(1) Synchronization attacks
We implement frame deletion, frame switching and TSM attacks on watermarked audio. Table 2 lists the BER values. The experimental results show that the robustness of the proposed method is higher than the method in [1].

Table 1. SNR and SDG values

| Method          | SNR (dB) | SDG |
|-----------------|----------|-----|
|                 |          |     |
| proposed method | 26.5     | 0.0 |
|                 | 23.3     | -0.1|
| Method in [1]   | 24.6     | 0.0 |
|                 | 22.7     | -0.1|

Table 2. Synchronization attacks

| Attacks                      | The proposed method (BER) | The method in [1] (BER) |
|------------------------------|----------------------------|------------------------|
|                              | popular music | piano | popular music | piano |
| Delete 1000 bits consecutively | 0 | 0.01  | 0 | 0.05 |
| Delete 2000 bits consecutively | 0.51 | 0.79 | 0.49 | 0.82 |
| Delete 5% of the data randomly | 0.96 | 1.08 | 1.23 | 1.56 |
| Delete 10% of the data randomly | 2.32 | 3.02 | 2.91 | 4.02 |
| Swap 3% of the data randomly | 1.03 | 1.11 | 1.12 | 1.33 |
| Swap 5% of the data randomly | 2.87 | 2.90 | 2.96 | 3.12 |
| Change TSM 2%              | 35.6     | 37.2 | 39.7 | 41.2 |
| Change TSM 5%              | 38.7     | 39.4 | 41.8 | 42.6 |
| Change TSM -2%             | 34.5     | 36.3 | 38.5 | 40.8 |
| Change TSM -5%             | 40.9     | 41.1 | 41.4 | 43.2 |

(2) Resampling attacks

We implement down-sampling on the watermarked audio, then use interpolation to restore the sampling frequency to 44.1 kHz. Table 2 lists the BER values. The experimental results show that the robustness of the proposed method against resampling attacks is the same as the method in [1].

Table 3. Resampling attacks

| Attacks          | The proposed method (BER) | The method in [1] (BER) |
|------------------|---------------------------|------------------------|
|                  | popular music | piano | popular music | piano |
| Resampling 22.05 kHz | 0 | 0.02  | 0 | 0.06 |
| Resampling 11.025 kHz | 0 | 0.16  | 0 | 0.27 |
| Resampling 8 kHz   | 0.13 | 0.28 | 0 | 0.33 |
| Resampling 6 kHz   | 0.22 | 0.32 | 0 | 0.40 |

(3) MP3 compression attacks

The compression rates are 128 kbps, 112 kbps, 96 kbps, are 80 kbps. Table 3 lists the BER values. The experimental results show that the robustness of the proposed method against MP3 compression attacks is higher than the method in [1].

Table 4. MP3 compression attacks

| Attacks  | The proposed method (BER) | The method in [1] (BER) |
|----------|---------------------------|------------------------|
|          | popular music | piano | popular music | piano |
| 128 kbps | 1.78         | 2.63  | 2.69         | 4.21  |
| 112 kbps | 6.87         | 8.15  | 8.75         | 10.33 |
| 96 kbps  | 15.46        | 17.22 | 19.32        | 22.46 |
| 80 kbps  | 22.71        | 25.34 | 26.45        | 29.12 |

(4) Low pass filtering attacks
Table 5 lists the BER values after 3 kHz and 5 kHz low-pass filtering. The experimental results show that the robustness of the proposed method against low pass filtering attacks is higher than the method in [1].

(5) Signal energy amplitude attacks
The amplitude of the watermarked audio signal is scaled by 0.8, 0.9, 1.1, and 1.2 times, respectively. The experimental results show that the robustness of the proposed method against signal energy amplitude attacks is the same as the method in [1].

| Attacks                      | The proposed method (BER) | The method in [1] (BER) |
|------------------------------|---------------------------|------------------------|
|                              | popular music            | piano                  |
| 3 kHz low pass filtering     | 5.86                      | 5.93                   |
| 5 kHz low pass filtering     | 1.55                      | 1.66                   |

(6) Echo attacks
The 40% echo signal is added to the watermarked audio signal by delays of 100ms, 200ms, 400ms, and 800ms. The experimental results show that the robustness of the proposed method against echo attacks is higher than the method in [1].

| Attacks                      | The proposed method (BER) | The method in [1] (BER) |
|------------------------------|---------------------------|------------------------|
|                              | popular music            | piano                  |
| 100ms, 40%                   | 0.13                      | 0.26                   |
| 200ms, 40%                   | 5.25                      | 7.53                   |
| 400ms, 40%                   | 4.27                      | 5.66                   |
| 800ms, 40%                   | 4.11                      | 5.78                   |

(7) Gaussian noise attacks
We add 10 dB, 20 dB, 30 dB of Gaussian white noise to the watermarked audio signal. The experimental results show that the robustness of the proposed method against Gaussian noise attacks is higher than the method in [1].

| Attacks | The proposed method (BER) | The method in [1] (BER) |
|---------|---------------------------|------------------------|
|         | popular music             | piano                  |
| 15 dB   | 0.13                      | 0.26                   |
| 20 dB   | 1.25                      | 1.07                   |
| 30 dB   | 15.72                     | 14.98                  |
| 40 dB   | 16.23                     | 16.49                  |

8. Conclusion
This work proposes a new dual-channel audio watermarking strategy, which uses the cross-correlation between channels to optimize the embedding strength and improved the imperceptibility of the watermark. We use the correlation method to detect the pitch and segment the audio with the pitch. The DWT is implemented on the audio signal to get the approximate coefficients. On this basis, the
coefficients are divided into odd-order and even-order channels. According to the high similarity between the two channels, the coefficients of one channel are embedded into the other channel in the form of echo. To improve the imperceptibility, the lowest SNR value after watermark embedding is used as the optimization target. By the optimization, the robustness is improved without affecting the original audio quality. In the extraction stage, the cross-correlation detection method is used to detect the signal peaks. According to the positions of the peaks, the final embedded watermark is extracted. Finally, we evaluate the proposed method through simulation experiments. The experimental data show that the proposed method has better robustness than the method in [1].

References
[1] Huang, Huang-Nan, et al. Optimization-based embedding for wavelet-domain audio watermarking[J]. Journal of Signal Processing Systems, 2015, 80(2): 197-208.
[2] Xiang, Yong, et al. A dual-channel time-spread echo method for audio watermarking[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(2): 383-392.
[3] Wang, Xiang-Yang, Pan-Pan Niu, and Hong-Ying Yang. A robust digital audio watermarking based on statistics characteristics[J]. Pattern recognition, 2009, 42(11): 3057-3064.
[4] Xiang, Yong, et al. Patchwork-based audio watermarking method robust to de-synchronization attacks[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2014, 22(9) : 1413-1423.
[5] Shin, Seungwon, et al. A robust audio watermarking algorithm using pitch scaling[C]. Digital Signal Processing, 2002 14th International Conference on Digital Signal Processing Proceedings. 2002, 2: 701-704.
[6] Coumou, David, and Gaurav Sharma. Watermark synchronization for feature-based embedding: Application to speech[C]. 2006 IEEE International Conference on Multimedia and Expo. IEEE, 2006.
[7] Charfeddine, Maha, Maher El’arbi, and Chokri Ben Amar. A new DCT audio watermarking scheme based on preliminary MP3 study[J]. Multimedia tools and applications, 2014, 70(3): 1521-1557.
[8] Lalitha, N. V., G. Suresh, and V. Sailinga. Improved audio watermarking using DWT-SVD[J]. International Journal of Scientific & Engineering Research, 2011, 2(6): 1-6.
[9] Ketcham, M., and Sartid Vongpradhip. Intelligent audio watermarking using genetic algorithm in DWT domain[J]. International Journal Of Intelligent Technology, 2007, 2(2): 135-140.
[10] Wang, Xiang-Yang, and Hong Zhao. A novel synchronization invariant audio watermarking scheme based on DWT and DCT[J]. IEEE Transactions on signal processing, 2006, 54(12): 4835-4840.
[11] Dai, Hua-liang, and Di He. An efficient and robust zero-watermarking scheme for audio based on DWT and DCT[C]. Microelectronics & Electronics, 2009. PrimeAsia 2009. Asia Pacific Conference on Postgraduate Research in. IEEE, 2009.
[12] Wu, Shaoyuan, et al. Efficiently self-synchronized audio watermarking for assured audio data transmission[J]. IEEE Transactions on Broadcasting, 2005, 51(1): 69-76.
[13] Jiasheng, Mei, Li Sukang, and Tan Xiaomei. A digital watermarking algorithm based on DCT and DWT[C]. Proceedings. The 2009 International Symposium on Web Information Systems and Applications (WISA 2009). Academy Publisher, 2009.
[14] Xiang, Shijun, Hyoong Joong Kim, and Jiwu Huang. Audio watermarking robust against time-scale modification and MP3 compression[J]. Signal Processing, 2008, 88(10): 2372-2387.
[15] Tzanetakis, George, and Perry Cook. Musical genre classification of audio signals[J]. IEEE Transactions on speech and audio processing, 2002, 10(5): 293-302.
[16] Wu, Hsien-Chung. The Karush–Kuhn–Tucker optimality conditions in an optimization problem with interval-valued objective function[J]. European Journal of Operational Research, 2007, 176(1): 46-59.