International Journal of Innovative Computing, Information and Control
Volume **16**, Number **6**, December **2020**

ICIC International ©2020 ISSN 1349-4198
pp. 1973–1985

DISCRETE FOURIER TRANSFORM PEAK DETECTION BASED ROBUST AUDIO WATERMARKING AGAINST TIME SCALE MODULATION AND PITCH SHIFTING

Xu-Qing Zhang and Zhe-Ming Lu*

School of Aeronautics and Astronautics
Zhejiang University
No. 38, Zheda Road, Hangzhou 310027, P. R. China
*Corresponding author: zheminglu@zju.edu.cn

Received July 2020; revised November 2020

Abstract. *Watermarking is a good way to protect the safety of audio information. In the fields of watermark research, the ability of defending synchronous attacks has always been the key and difficult point. This paper provides an audio watermark algorithm based on Discrete Fourier Transform (DFT) peak detection, which can commendably defend synchronous attacks like Time Scale Modulation (TSM) and pitch shifting. Discrete Wavelet Transform (DWT) is performed firstly to get the detailed information. Then we divided it into a certain number of segments and for each segment we perform DFT transform, we extract the mark, and we only need to find if there have been peaks. Experiments demonstrate that the watermarked audio is perceptually close to the original one as the average signal-to-noise ratio is about 31.62 dB.*

Keywords: Audio watermarking, Synchronization attack, DFT transform, Peak detection

1. **Introduction.** With the development of Internet, nowadays, media files including audios have been widely used in our daily life. Audio is a good form of information, which has been focused on all the time. It has many advantages like easy to store, easy to copy and easy to edit. However, in the meanwhile, these qualities also make the information become easy to eavesdrop or tamper, which will cause many safety problems. New models have been made to evaluate the quality of audios [1] and a lot of methods have been proposed to protect the information. One of the most popular ways is digital watermarking. In this way, we can embed extra information into audio files with imperceptible changes. When we need to clarify the copyright, we only need to extract the embedded information even if the media file has been attacked.

Digital watermarking schemes can be divided into robust watermarking and fragile watermarking schemes. Robust watermarking will not change when media files have been attacked while fragile watermarking changes. There are three criterions to evaluate a robust watermarking algorithm: robustness, imperceptibility and capacity, which are negatively correlated. When you improve one criterion, the other two are very likely to recede. According to the standard of Intentional Federation of the Phonographic Industry (IFPI), to get an imperceptible watermarked audio, the Signal to Noise Ratio (SNR) should be larger than 20 dB.

Audio watermarking algorithms are mainly based on time domain, frequency domain and compression domain. Compared to the other two types, frequency domain algorithms can preferably balance the performance and complexity. The most widely used frequency

DOI: 10.24507/ijicic.16.06.1973

transforms are Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT). DFT transform can obtain the frequency characteristics of the audio. When we change the frequency data rather than the time domain data, our ears are less likely to recognize the difference. And the frequency characteristics are more stable than time domain ones when audios being attacked. Many audio schemes based on DFT transform have been proposed. Tilki and Bexx [2] used DFT transform in their algorithm as early as 1996. And Fallahpou and Megias [3] combined FFT with linear regression and embedded watermarks by changing the magnitude of FFT results, whose value depends on the linear regression analysis. DCT transform is based on the DFT transform. It retains the real part of the DFT result while discards the imaginary part. In 1998, Wang [4] proposed a watermark algorithm based on DCT and then proposed MDCT (Modified DCT) algorithm. In 2015, Hu and Hsu [5] proposed an algorithm that embeds watermark in three groups of DCT coefficients according to the masking characteristics of human auditory system. The DWT transform can extract the rough and detailed information respectively in high frequency and low frequency data. It also has been widely used in audio watermarking schemes. Huang et al. [6] in 2015 proposed an audio blind watermarking scheme in DWT domain. It can adaptively modify algorithm parameters according to the Signal to Noise Ratio (SNR) of audios, and this algorithm optimized the embedding formula by minimizing the difference between the original coefficient and the embedding coefficient. Considering the different features of different frequency transforms, in this paper, we choose to combine the DFT transform and DWT transform to make a better scheme.

In the field of audio watermarking, the ability to resist synchronization attacks has always been the key and difficult point. Synchronization attacks, including random cropping, Time Scale Modulation (TSM), pitch shifting, etc., are attacks which will destroy the synchronism between watermark and audio. Many synchronization mechanisms have been proposed and many researches also show some ability to resist synchronous attacks. DFT is the most widely used frequency domain among existing audio watermarking algorithms, especially those against synchronization attacks. Kang et al. [7] proposed a method using Logarithmic Coordinate Mapping (LCM) to synchronize signals, which uses the geometric invariance of logarithmic coordinate mapping to embed watermark in the Fourier coefficient. Fan and Wang [8] used the statistical properties of the DFT coefficients to solve the problem of different audio playback speeds. They divided the audio into multiple frames, segmented each frame into DFT, and embedded the watermark by modulating the average of DFT magnitude using three consecutive segments. One famous synchronous watermarking algorithm in time domain is based on the statistical property of audios. Xiang and Huang [9] proposed the method of redistribution of the number of points in the histogram and subsequently proposed the application of DWT transformation to improve the performance of the algorithm [10]. Recently, more complicated algorithms using multiple mechanism have been proposed to improve the performance. In 2019, Liu et al. [11] used a patchwork framework to embed the watermark, which took advantage of the residual of the constructed feature Frequency-Domain coefficients Logarithmic Mean (FDLM) belonging to two groups. In addition, Hu and Chang [12] used multi-layer DWT to insert synchronous sinusoidal signals into the 11th sub-band. Based on the LWT-SSR method, Hu et al. [13] proposed method locating the embedded watermark by means of histogram and synchronization signal. In 2019, Jiang et al. [14] proposed a de-synchronization method based on the global characteristics of the frame and achieved indirect synchronization by employing a corresponding frame sequence number. Especially, speech is a specific kind of audio, which have many parts of silence. As for this specific type of audio, Liu et al. [15] proposed a synchronous watermarking method in 2020, in which they defined a feature called the Discrete Cosine Transform Coefficients Logarithm Mean (DCT-CLM) and embedded watermark into this feature belonging to non-silence parts.

In this paper, we proposed an algorithm which can better resist synchronous attacks than existing methods. We combine the advantages of DFT and DWT transform. We use the stability of audio signal in DFT domain and the watermark is embedded by superimposing peaks at specific positions. At the same time, in order to increase the concealment to make the human ear cannot distinguish the audio before and after embedding the watermark, DWT transformation is performed on the signal before embedding and the high frequency coefficient is taken for processing.

The remainder of this paper is organized as follows. Section 2 will introduce some basic methods applied in the algorithm in detail and analyze the stability of the characteristics under synchronous attack. In this section, we will also give the specific watermark embedding and extraction steps. Section 3 includes some algorithm testing and experimental analysis. And in Section 4, we will give the conclusion.

2. **Techniques for Synchronous Audio Watermarking.**

2.1. **DFT and DWT definitions.**

Discrete Fourier Transformation (DFT) is a transform focusing on discrete signals, which changes the signal from the time domain to the frequency domain. The transform is reversible. The DFT signal can be transformed back to the original signal without damage using the inverse transform. The formula of DFT transform and the inverse one is respectively shown in Equation (1) and Equation (2). The $x(n)$ represents the original discrete sequence and $X(k)$ represents the transformed DFT domain sequence.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \tag{1}$$

$$x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(k) e^{j\frac{2\pi}{N}nk} \tag{2}$$

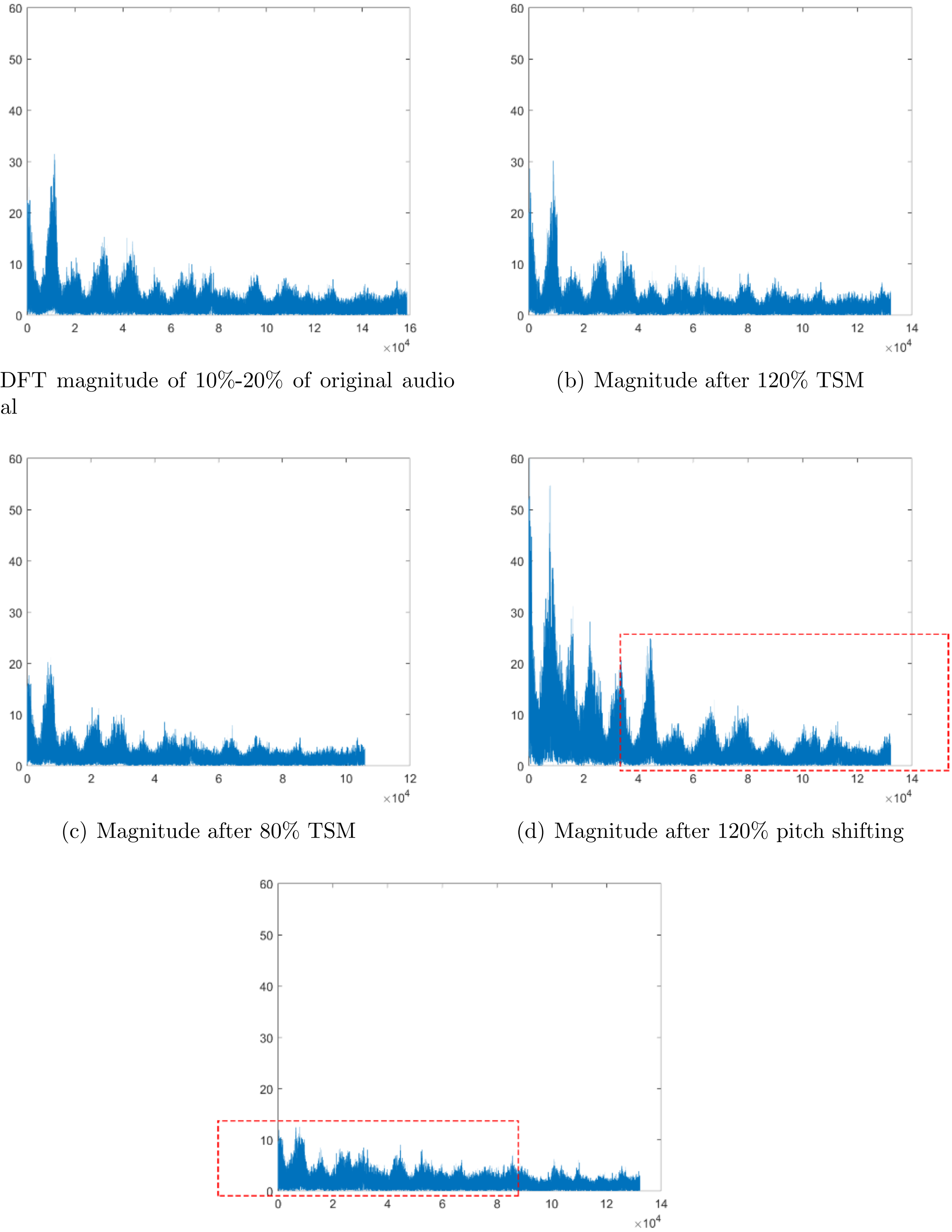
The DFT coefficients of an audio have stability in some respects. Suppose the result of DFT transform coefficients of the audio signal $f(t)$ is $F(\omega)$.

When we apply the TSM attack to the audio, the audio signal will become $f'(t) = f(t/\alpha)$, and α is the transform coefficient. After the attack, DFT coefficients become $F'(\omega) = \alpha F(\alpha\omega)$, which means the DFT coefficient is α times bigger than the original one while the shape of the coefficient remains the same as before.

And when we apply pitch shifting attack to the audio, its tone will rise or fall. When the tone of the audio changes, the DFT coefficient is shifted to the left or right as a whole, and the shape of the coefficient remains unchanged.

Figure 1 verifies this conclusion through experiments. Taking the audio “Symphony no. 40 in G Minor” for example, we take the first 30 seconds of the audio for experiment. To facilitate observation, 10% to 20% of the DFT coefficients is shown in the figure. Figure 1(a) shows the DFT magnitude coefficients of the audio. Figure 1(b) and Figure 1(c) respectively show the DFT magnitude after 120% and 80% resampling TSM attack. And Figure 1(d) and Figure 1(e) show the results after 120% and 80% pitch shifting attack.

Discrete Wavelet Transform (DWT) is a multi-scale transform, which is an improvement of Fourier transform. Traditional Fourier transform only transforms the signal from the time domain to the frequency domain. It is a completely frequency-domain analysis and cannot locate the time domain. DWT transform can decompose the signal on multiple scales. The decomposed signal retains the characteristics of the time domain, and the



(a) DFT magnitude of 10%-20% of original audio signal

(b) Magnitude after 120% TSM

(c) Magnitude after 80% TSM

(d) Magnitude after 120% pitch shifting

(e) Magnitude after 80% pitch shifting

FIGURE 1. The 10%-20% DFT magnitude of the original audio and the audio after attacks: (a) is the original DFT magnitude; (b) and (c) present the magnitude after 120% and 80% TSM; (d) and (e) show the magnitude after 120% and 80% pitch shifting

decomposed low-frequency result highlights the main characteristics of the original signal, while the high-frequency result reflects the details.

DWT transform mainly uses the famous Mallat algorithm. It passes the signal through the low-pass filter and the high-pass filter respectively, then carries on the down sampling, and thus obtains the low frequency and high frequency components. The specific process is shown in Figure 2. S is the original sequence which will be transformed. $H1$ and $L1$ respectively represent the high and low frequency parts of DWT result. We can also do the decomposition to the low-frequency part $L1$ and get the second level DWT coefficients $H2$ and $L2$. We can get higher level of DWT by continually decomposing the low frequency part.

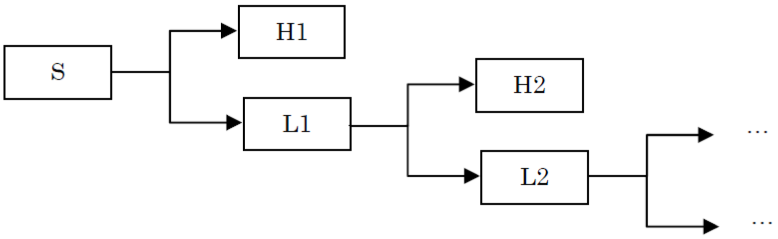


Figure 2. Specific process of Mallat algorithm

In order to enhance the concealment, our algorithm conducts DWT on the signal, takes the high frequency coefficient, carries on the segmentation, takes the DFT respectively to each section, and finally does the watermark embedding process. Because the high frequency coefficients of DWT transform are the detailed reflection of audio signal, they change with the original audio. Therefore, when the original signal suffers synchronous attack, the DFT coefficients of high frequency details have the same changing tendency with the DFT results of original signal. As a result, the above analysis of DFT transform is also applicable to DWT high frequency signals.

2.2. **Peak detection.** The watermarking method adopted in this paper is as follows: firstly, DWT transform is carried out on audio, high frequency coefficients are taken for segmentation, and then DFT transform is carried out on each segment respectively. When the watermark bit 0 is embedded, a peak value is added to the five specific positions of DFT magnitude coefficients. No operation is performed when embedding the watermark bit 1. When we extract the watermark, the watermark value can be determined by whether we can detect a certain number of peaks.

When the audio is attacked, the peak value of the DFT may change as follows. 1) The value of the peak changes. After various attacks, the coefficients of the DFT may become larger or smaller, and the value at the peak point may decrease so much that it changes from peak to non-peak. 2) The location of the peak changes. According to the analysis of the change of DFT coefficients in Section 2.1, the DFT coefficients will shift after the audio tone is changed. This results in the peaks' location changing, and the peaks will be shifted away from the original embedding position. When the modulation is too large, the deviation will be larger and even cause the edge peak to disappear. 3) The peak range is enlarged. When embedding the mark, the algorithm only changes the DFT coefficient at one point. However, after various attacks, the points around the peak point may be convolved with the peak point; thus their value will also be larger. This makes the original steep peaks become smoother and they will be more difficult to detect.

In view of the above changes, an effective peak detection method is proposed. Our method has a high correct rate when recognizing peaks. Because when audios suffer from synchronous attacks, the location of the peak changes, and we firstly need to locate the peaks. We use the way of voting to get the most likely positions. As for the change of peak value, we set a parameter to control the threshold. And we use the thought of local average to reduce the influence of the enlarged peak range. The specific steps are as follows.

2.2.1. *Determine the peak position.* When the audio carries out an overall synchronous attack, the DFT coefficients of every segment will have the same trend. Therefore, the proposed scheme selects the point positions by voting. For each segment, our method selects a range of 20%-50% of DFT coefficients. Then we select 5 points within this range that have a local maximum and count their positions in the vote. The local range is δ, so the points must accord with the formula: $|p_{\max i} - p_{\max j}| > \delta$. Based on the overall voting results, the top 5 positions are selected as the peak positions after the attack, denoted as p_i $(i = 1, 2, 3, 4, 5)$.

2.2.2. *Determine the existence of peaks.* For the DFT coefficients of each segment, firstly we select five points with local maximum values. Then for each point, we will do the judgement and determine if it is within the local range of p_i and greater than the threshold th. If the condition is satisfied, the point is considered to have a peak value at p_i; otherwise, it will be considered as a false peak. That is, the conditions to be met are:

$$\begin{cases} |p - p_i| < \Delta \\ F(p) > th \end{cases} \tag{3}$$

When we want to decide whether a peak is a real peak, we only need to compare it to the large ones around it. As a result, we choose a certain number of maximal points and do the judgement. The number of points we choose is n, which all belong to the range $F(i)$ $(i = p - l_1, p - l_1 + 1, \ldots, p - l_2 - 1, p - l_2, p + l_2, p + l_2 + 1, \ldots, p + l_1 - 1, p + l_1)$. l_1 and l_2 are set to prevent points too far away and too close to the peak from affecting the threshold calculation results. The formula of threshold is as follows:

$$th = \beta \times \frac{\sum_{i=1}^{n} F(posi(i))}{n} \tag{4}$$

$posi(i)$ is the position of the ith local maximum point, n is the number of chosen points, and β is an adjustable parameter that adjusts the range of thresholds.

2.2.3. *Determine the watermark value.* Using the calculation method of Section 2.2.2, if there are 3 or more true peaks in the calculated 5 points, the watermark value is considered as 0; otherwise, the watermark value is considered as 1.

2.3. **Procedures for watermark embedding.** To improve the algorithm's ability of correcting errors, we firstly apply BCH encode to the original watermark. In this paper, we use $(255, 215, 11)$ BCH code, which means we can embed 215 bits information and can correct 5 bits errors at most. Then we use the algorithm based on DFT peak detection to embed the watermark into the original audio signal. The embedding process of watermark is shown in Figure 3. Specific embedding steps are as follows:

1) DWT transformation. We firstly perform DWT transform on the original signal, and then the transformed high frequency coefficients are taken.

2) Segmentation. Assuming the length of embedding watermark is L, then we divide the high frequency coefficients into L segments on average.

3) Watermark embedding. If the watermark bit is 1, the data is not transformed. If the watermark bit is 0, new peaks are added. Specific methods are as follows.

DFT transform is first performed for each section, and its magnitude is calculated. Then we select 10%-50% of the magnitude data to get the maximum value, and take 3 times of the maximum value as the newly added peak pk, namely:

$$pk = 3 \times \max\{F(w_j)\} \tag{5}$$

Select five points as the embedding point of the peak. At these embedding points, the value of the DFT modulus is set to peak pk. The points we choose should not be at too low frequency part because human's ears are more likely to notice the change of low frequency signal. And they also should not be at too high frequency part. The reason is that when the audio suffers from synchronous attacks, DFT coefficients will be translated. If the attack is strong enough, some high frequency part will be translated too much that we will lose them. Besides, if the points are too close, the calculation of one point will be influenced by neighboring ones. Additionally, we need to narrow the peaks' range as much as possible to improve the imperceptibility. As a result, the five points selected are:

$$\begin{cases} p_1 = \left\lfloor \dfrac{10}{32} \times length\_data \right\rfloor, & p_2 = \left\lfloor \dfrac{11}{32} \times length\_data \right\rfloor \\ p_3 = \left\lfloor \dfrac{12}{32} \times length\_data \right\rfloor, & p_4 = \left\lfloor \dfrac{13}{32} \times length\_data \right\rfloor \\ p_5 = \left\lfloor \dfrac{14}{32} \times length\_data \right\rfloor \end{cases} \tag{6}$$

The symbol $\lfloor . \rfloor$ means round down.

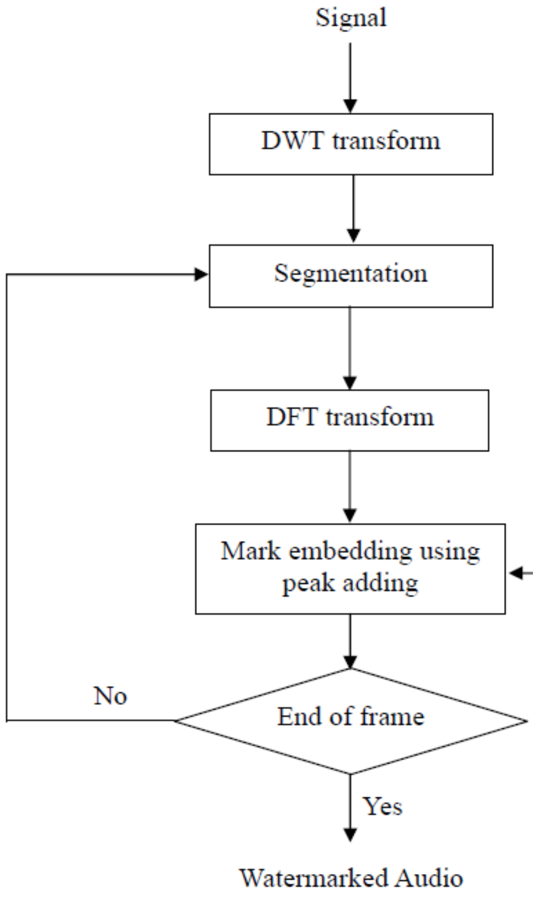


Figure 3. Watermark embedding procedure

4) Inverse transform. The inverse DFT transform is carried out for each segment of DFT data embedded with watermark, and segment integration is carried out after transform. The integrated data and the previous low frequency data of DWT were combined for DWT inverse transform. The transformed data is the final audio data embedded with watermark.

2.4. **Procedures for watermark extracting.** The extracting process of watermark is shown in Figure 4. Specific extracting steps are as follows:

1) DWT transform. We firstly perform DWT transform on the original signal, and then the transformed high frequency coefficient is taken.

2) Segmentation. Assuming the length of embedding watermark is L, then we divide the high frequency coefficients into L segments on average.

3) Watermark extraction. Peaks are detected using the peak detection method described in Section 2.2. If more than 3 peaks are detected, the watermark bit is considered to be 0; otherwise, the watermark bit is 1.

4) BCH decoding. The extracted watermark was decoded by BCH to correct the errors in the extraction process. The watermark obtained after decoding is the final information obtained.

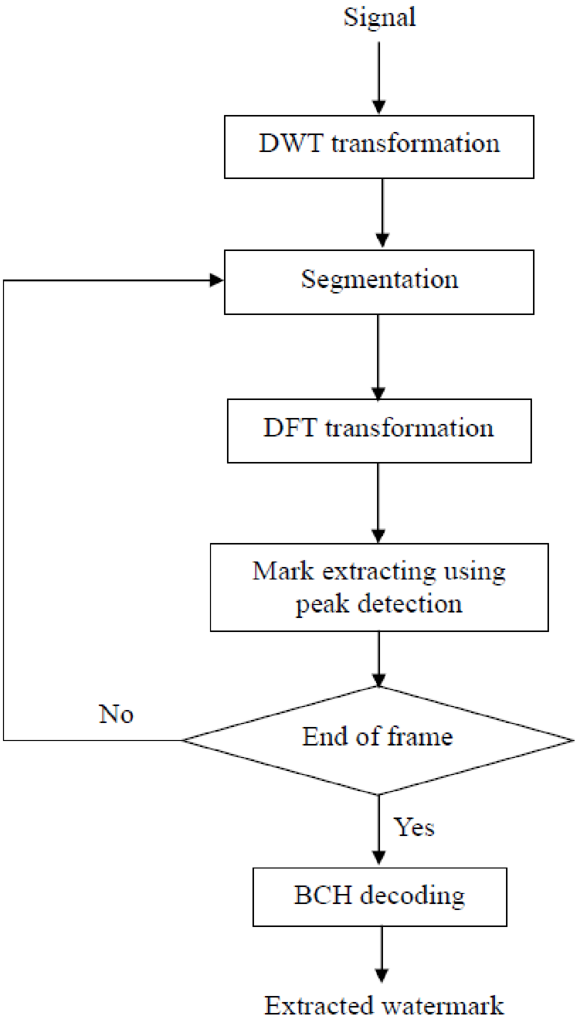


Figure 4. Watermark extracting procedure

3. **Performance Evaluation.** The performance of the algorithm can be judged by capacity, imperceptibility and robustness.

Our experiment uses 40 audios of .wav format with length of 30 s, sampling rate of 44100 Hz and depth of 16 bits for test. The audio covers eight genres, including classical, dance, rock, pop, piano, jazz, Latin and vocal, with five samples each. The extensive sample quantity and sample type guarantee the universality of the algorithm.

During the test, the parameter β of threshold th is set as 1.4 in this paper, which is determined by experiments shown in Figure 5. What we mostly care about is the BER after BCH decode and whether the extracted mark can be back to the embedding one, so we compare the BER after decoding for different β. And we use the audios attacked respectively by 70% resample TSM, pitch-invariant TSM and pitch shifting as the test samples. As we can see, the best performance occurs when the β is set to 1.4. Because the interval between the two peaks is 1/32 of the block length when we embed the mark, we choose half of the original interval to be the local range. As a result, the local range δ is 1/64 of the block length. The test uses $(255, 215, 11)$ BCH code, which can correct up to 5 bits of errors, that is, when the number of errors is less than 5, the information can be completely restored to the original information content.

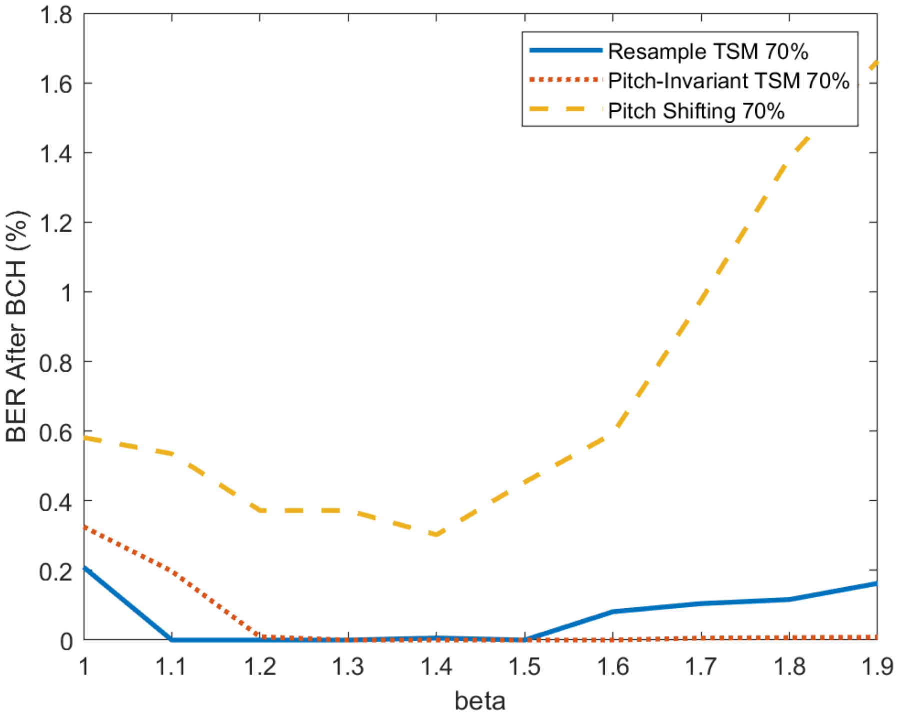
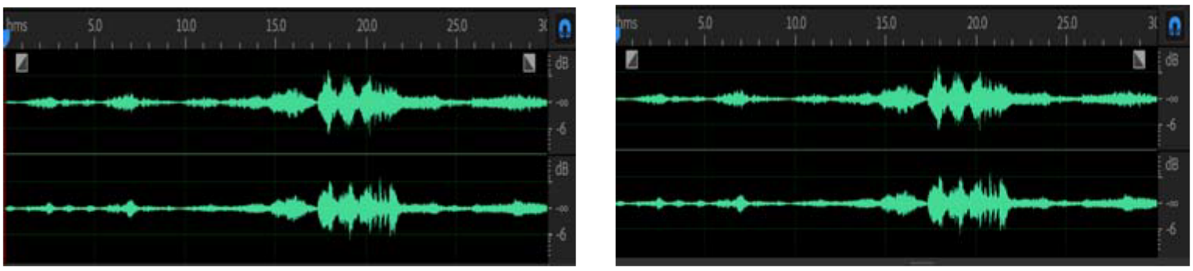


Figure 5. BER after BCH for different β and different synchronous attacks

3.1. **Capacity.** The audio sampling rate is 44100 Hz, indicating that the audio contains 44,100 data per second. For each piece of audio, we embed with 255 bits of data. Due to the addition of BCH code, the actual effective message length is 215 bits. Therefore, the watermark capacity of the algorithm is 7.17 bps, that is, 7.17 bits of data can be embedded in the audio data of 1 s.

3.2. **Imperceptibility.** The paper adopts SNR to evaluate the concealment. To be inaudible, the SNR value should be greater than 20 dB according to the IFPI. In the experiment, the watermark was embedded into 40 audios. Figure 6 shows the waveform



(a) The waveform before embedding

(b) The waveform after embedding

Figure 6. Comparison of audio waveform before and after embedding watermark

Table 1. The SNR results of audios before and after watermark embedding

	Type								Average
	Classical	Dance	Rock	Popular	Latin	Jazz	Piano	Vocal	
SNR (dB)	37.49	30.63	24.40	29.65	26.55	34.91	37.76	31.55	31.62

of one of the audios before and after watermark embedding. As we can see, its waveform is consistent. The SNR between two audios before and after watermark embedding is calculated, and the calculated results are shown in Table 1. The results show that the average SNR value of audio is 31.62 dB, which conforms to the required range, and the SNR value calculated separately for each category is also greater than 20 dB, which meets the SNR requirement. This experiment shows that this algorithm performs well in the aspect of imperceptibility for multiple audio types.

3.3. **Robustness.** Robustness means that the audio watermark can be extracted successfully under various attacks, and BER (Bit Error Rate) is calculated to conduct performance evaluation. IFPT provides that BER less than 20% means the watermark can be extracted successfully.

The attacks in our test are divided into ordinary attacks and synchronous attacks. Ordinary attacks include noise, resampling, requantization, MP3 compression and low-pass filtering, as shown in Table 2. For ordinary attacks, the specific test results are shown in Table 3. It can be seen from the test results that the proposed algorithm can completely resist the noise attack, and can partially resist the MP3 format conversion, requantization and low-pass filtering, but it is completely unable to resist the resampling.

Synchronous attack test mainly includes resampling TSM test, pitch-invariant TSM test and tempo invariant pitch shifting test. The specific test results of synchronization

Table 2. Ordinary attack types and specifications

	Attack type	Specification
A	Noise (1)	Add Gaussian noise to the audio with SNR = 30 dB.
B	Noise (2)	Add Gaussian noise to the audio with SNR = 20 dB.
C	Resample	Resample the watermarked signal to 22.05 kHz and then back to 44.1 kHz.
D	Requantization	Quantize the watermarked signal to 8 bits/sample and then back to 16 bits/sample.
E	MP3	Compress the watermarked audio signal with an MPEG-1 layer 3 coder at a bit rate of 128 kbps.
F	Low-pass filter	Apply a low-pass filter with a cutoff frequency of 8 kHz.

Table 3. The results of ordinary attacks

Attack	BER before BCH (%)	BER after BCH (%)
No attack	0.039	0
A	0.029	0
B	0.039	0
C	50.392	50.047
D	3.245	3.163
E	35.029	34.721
F	7.029	6.791

Table 4. The results of synchronous attacks

Attacks		Max error number	Min error number	Unable to fully recover audio number (unable to extract number/total audio number)	BER before BCH (%)	BER after BCH (%)
No attack		3	0	0/40	0.0392	0.0000
Resample TSM	70%	6	0	1/40	0.1078	0.0058
	80%	3	0	0/40	0.0098	0.0000
	90%	1	0	0/40	0.0039	0.0000
	110%	2	0	0/40	0.0490	0.0000
	120%	2	0	0/40	0.0078	0.0000
	130%	4	0	0/40	0.0980	0.0000
	140%	2	0	0/40	0.0078	0.0000
Tempo invariant pitch shift	70%	11	0	3/40	1.0490	0.3023
	80%	7	0	2/40	0.4314	0.1628
	90%	1	0	0/40	0.0294	0.0000
	110%	2	0	0/40	0.0049	0.0000
	120%	2	0	0/40	0.1176	0.0000
	130%	3	0	0/40	0.1176	0.0000
	140%	4	0	0/40	0.3235	0.0000
Pitch-invariant TSM	70%	2	0	0/40	0.1417	0.0000
	80%	1	0	0/40	0.0010	0.0000
	90%	1	0	0/40	0.0010	0.0000
	110%	2	0	0/40	0.0098	0.0000
	120%	2	0	0/40	0.1373	0.0000
	130%	3	0	0/40	0.1471	0.0000
	140%	4	0	0/40	0.1765	0.0000

attacks are shown in Table 4. As we can see, for 80%∼140% resampling TSM attacks, 70%∼140% pitch-invariant attacks and 90%∼140% tempo invariant pitch shifting attacks, all test audio can completely resist the attacks, and the extracted information can be completely restored to the original information. For 70% resampling TSM attacks and 70% and 80% tempo invariant pitch shifting attacks, a small part of audio cannot achieve complete resistance. However, for most audio, there are very few audio errors that can be fully recovered to the original message.

We compare our method to existing well performed methods in the TSM attack field. The methods we use involve different domains and different embedding ways. Method [8]

Table 5. Comparison of the ability to resisting TSM attacks (BER in percent)

Attack type	Our	Method [8]	Method [7]	Method [10]
Resample TSM −20%	0	6.6667	0	0.3000
Resample TSM −10%	0	6.1111	0	0.2600
Resample TSM +10%	0	5.5556	0	1.2800
Resample TSM +20%	0	6.6667	0	0.5300
Pitch-invariant TSM −20%	0	0	5.4675	Not Mentioned
Pitch-invariant TSM −10%	0	0	0.3900	Not Mentioned
Pitch-invariant TSM +10%	0	0	0.7800	Not Mentioned
Pitch-invariant TSM +20%	0	0	7.0325	Not Mentioned

uses the statistical property of the audio signal, and embeds the watermark by changing the neighboring bins of its histogram. Method [7] is a method in DFT domain. It uses Logarithmic Coordinate Mapping (LCM) to synchronize signals, and embeds watermark in DFT coefficient. Method [10] synchronizes signals by adding a sinusoid signal, and uses the shape configuration of sorted LWT coefficient magnitudes to embed watermark. As we can see, our method performs better than those methods both in resampling TSM and pitch-invariant TSM.

The test results show that this algorithm has strong resistance to synchronous attacks, and can withstand about 30% of resampling TSM, pitch-invariant TSM and tempo invariant pitch shifting. Compared with the existing anti-synchronous attack algorithms, our algorithm shows excellent performance in this aspect. However, its resistance to ordinary attacks needs to be further strengthened.

4. **Conclusion.** In this paper, we proposed an audio watermarking algorithm based on DWT-DFT peak detection. This algorithm utilizes DWT to extract features and uses the characteristics of DFT coefficient stability under synchronous attacks. We combine DWT and DFT's advantages, and adopt the peak detection method to extract watermarks. The algorithm innovatively embeds the peak value in the DFT coefficient and uses the detection of the peak value to determine the extracted watermark value. In addition, BCH error correction code is added to the algorithm. BCH code can effectively correct errors of a certain number of bits at the expense of embedded information capacity.

The experimental results show that our algorithm is effective in anti-synchronous attacks. For about 70%∼130% resampling TSM, pitch-invariant TSM and tempo invariant pitch shifting, the number of errors generated by most audio extraction is less than the upper limit of BCH error correction, so the information can be fully extracted. For a small number of audios, when under strong attacks, the number of detected errors will exceed the error correction range. However, the number of errors is small, and the number of exceeded audios is very small. Therefore, it can be considered that the basic watermark extraction can be realized under synchronous attacks in the range of 70%∼140%.

REFERENCES

[1] A. Kanda, S. Ishimitsu, K. Wakamatsu, M. Nakashima and H. Yamanaka, Objective evaluation of sound quality for audio system in car, *ICIC Express Letters, Part B: Applications*, vol.10, no.4, pp.335-342, 2019.

[2] J. F. Tilki and A. A. Bexx, Encoding a hidden auxiliary channel onto digital audio signal using psychoacoustic masking, *Proc. of the 7th International Conference on Signal Processing Application Technology*, Blacksburg, VA, USA, pp.476-480, 1996.

[3] M. Fallahpou and D. Megias, High capacity robust audio watermarking scheme based on FFT and linear regression, *International Journal of Innovative Computing, Information and Control*, vol.8, no.4, pp.2477-2489, 2012.

[4] Y. Wang, A new watermarking method of digital audio content for copyright protection, *Proc. of the 4th International Conference on Signal Processing*, Beijing, China, pp.1420-1423, 1998.

[5] H. T. Hu and L. Y. Hsu, Robust transparent and high capacity audio watermarking in DCT domain, *Signal Processing*, vol.109, pp.226-235, 2015.

[6] H. N. Huang, S. T. Chen, M. S. Lin, W. M. Kung and C. Y. Hsu, Optimization-based embedding for wavelet-domain audio watermarking, *Journal of Signal Processing Systems*, vol.80, no.2, pp.197-208, 2015.

[7] X. Kang, R. Yang and J. Huang, Geometric invariant audio watermarking based on an LCM feature, *IEEE Trans. Multimedia*, vol.13, no.2, pp.181-190, 2011.

[8] M. Q. Fan and H. X. Wang, Statistical characteristic-based robust audio watermarking for resolving playback speed modification, *Digital Signal Processing*, vol.21, no.1, pp.110-117, 2011.

[9] S. Xiang and J. Huang, Histogram-based audio watermarking against time-scale modification and cropping attacks, *IEEE Trans. Multimedia*, vol.9, no.7, pp.1357-1372, 2007.

[10] S. Xiang, H. J. Kim and J. Huang, Audio watermarking robust against time-scale modification and MP3 compression, *Signal Processing*, vol.88, no.10, pp.2372-2387, 2008.

[11] Z. H. Liu, Y. K. Huang and J. W. Huang, Patchwork-based audio watermarking robust against de-synchronization and recapturing attacks, *IEEE Trans. Information Forensics and Security*, vol.14, no.5, pp.1171-1180, 2019.

[12] H. T. Hu and J. R. Chang, Efficient and robust frame-synchronized blind audio watermarking by featuring multilevel DWT and DCT, *Cluster Computing*, vol.20, no.1, pp.805-816, 2017.

[13] H. T. Hu, J. R. Chang and S. J. Lin, Synchronous blind audio watermarking via shape configuration of sorted LWT coefficient magnitudes, *Signal Processing*, vol.147, no.1, pp.190-202, 2018.

[14] W. Z. Jiang, X. H. Huang and Y. J. Quan, Audio watermarking algorithm against synchronization attacks using global characteristics and adaptive frame division, *Signal Processing*, vol.162, no.1, pp.153-160, 2019.

[15] Z. H. Liu, Y. C. Yang, D. Luo and C. D. Qi, Speech watermarking robust against recapturing and de-synchronization attacks, *Multimedia Tools and Applications*, vol.79, nos.9-10, pp.6009-6024, 2020.