An Audio Blind Watermarking Algorithm by Modifying the DWT Low-frequency Coefficients

Xuesong Chen[1] **Haotian Li**[1] **Guanglong Bu**[1] **Hongbo Bi**[1]

Abstract. An audio watermarking algorithm by modifying the low-frequency coefficients of DWT is discussed in this paper. Audio signal is divided into many sections, whose quantity is equal to the size of the watermark. The maximal value and the second maximal value of DWT low frequency coefficient can be found in each section. Then we can extract the last non-zero number of the two extreme values. The watermark is embedded based on the two numbers' parity. We take the encryption twice in order to improve security of the watermark. Experiments show that after some common attacks such as resample, Gauss noise, re-quantization and low-pass filtering, the algorithm has good imperceptibility and robustness.

Keywords: Audio blind watermarking. DWT. Security

1.1 Introduction

Discrete wavelet transform (*DWT*) has many advantages such as multi-scale and multi-resolution. It is good at processing one-dimensional signal, so it is widely used in the field of digital watermarking. Xu Dacheng proposed a non-blind audio watermarking algorithm basing on DWT.[2] It can't realize blind extraction. Chen Licong proposed a blind audio watermark algorithm based on local extreme points.[1] The algorithm is essentially to find the extreme value point of the wavelet low frequency coefficient. Watermark embedding is implemented by modifying the amplitude value of extreme points. Because the algorithm is complicated and needs large amount of calculation, so it is very difficult to realize. Zhang Zhijie[3] proposed an algorithm by modifying intermediate frequency coefficients in wavelet domain. But comparing with the wavelet low frequency domain and intermediate frequency domain, low frequency domain usually has better robustness.

[1]Xuesong Chen (✉) Haotian Li(✉) Guanglong Bu(✉) Hongbo Bi (✉)
Electrical & Information Engineering College, Northeast Petroleum University. 163318, Daqing, China
e-mail: cxsnepu@163.com lihaotian0010@163.com

1.2 Watermark Image Preprocessing

In order to compare the original watermark and the extracted watermark more intuitively, the watermark image uses the binary image which is named as w_0, and which size is $32*32$. Watermark sequence is encrypted twice for improving watermark's security. **Fig. 1.1** is the watermark encryption procession.

Arnold scrambling is the first step. Then watermark image is down-dimensionality denoted by A. Using the former 1024 numbers of the logistic chaotic sequence, the XOR operation is made. The secondary encryption watermark is implemented. **Fig. 1.2** is Encrypted watermark. And w_0 is original watermark in **Fig. 1.2(a)**, w_1 is the first encrypted watermark by Arnold scrambling in **Fig. 1.2(b)**, and w_2 is the second encrypted watermark by logistic chaotic encryption in **Fig. 1.2(c)**.

1.3 Algorithm Implementation

1.3.1 Watermark Embedding

Because high frequency coefficients of wavelet describe the detail component of the audio signal, and low frequency coefficients describe the rough component of the audio signal, low frequency domain usually has better robustness besides comparing with low frequency domain and intermediate frequency domain. So we choose the low-frequency component of wavelet to embed the watermark.[4] The information of encrypted watermark is embedded into the low frequency coefficients of wavelet in DWT domain. Finally, the watermarked audio signal can be obtained by wavelet reconstructing, which is shown as **Fig. 1.3**.

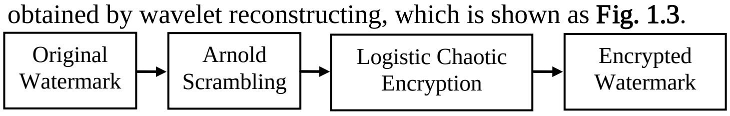
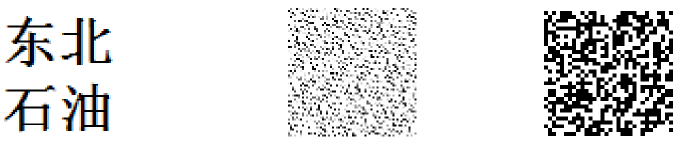


Fig. 1.2 Watermark encryption procession



(a) Original watermark w_0 (b) Arnold scrambling w_1 (c) Logistic Chaotic Encryption w_2

Fig. 1.2 Encrypted watermark

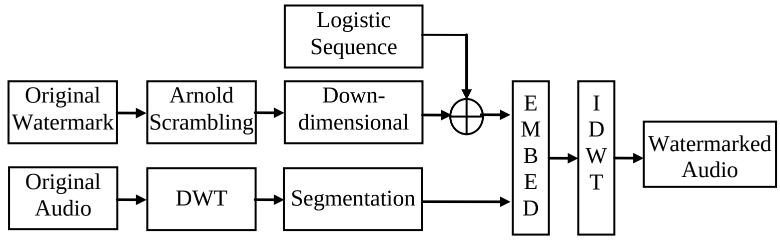


Fig. 1.3 Watermark Embedding Procession

1. Audio signal is divided into many sections. The quantity is equal to the size of the watermark ($M_1 \times M_2$). Each data segment has N sampling points. **Fig. 1.4** is original audio segmentation schematic

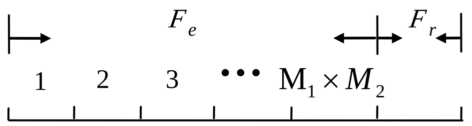


Fig. 1.4 Original audio segmentation schematic

Audio signal is divided into two sections.

$$\mathrm{F} = F_e + F_r \tag{1.1}$$

The relevant part of embedding watermark can be seen as F_e, whose size is $N \times M_1 \times M_2$. This algorithm chooses the value of N carefully. If the value of N is too small, the proportion of F_e is small over the entire audio and F_e is clipped easily. Not all information is embedded if the value of N is not too large. And F_r is the independent part of embedding watermark. This part remains unchanged in the procession of watermark embedding.

2. L level discrete wavelet transforms may be done to the each data segment of F_e. We find the absolute value of the maximum value and the second maximum value from each L-level low-frequency component, which is denoted by *MAX1* and *MAX2* respectively. The last nonzero number of *MAX1* and *MAX2* is extracted and denoted by $MAX1_l$ and $MAX2_l$ respectively.
3. When the embedded watermark is "1", if $MAX1_l$ mod 2 is equal to $MAX2_l$ mod 2, you do not need to modify the value of $MAX1_l$; if $MAX1_l$ mod 2 is not equal to $MAX2_l$ mod 2, you need to modify the value of $MAX1_l$. If the value of $MAX1_l$ is equal to 9, the value of $MAX1_l'$ is changed to $MAX1_l$-1, the value of $MAX1_l'$ is equal to $MAX1_l$+1 in other conditions. When the embedded watermark is "0", if $MAX1_l$ mod 2 is not equal to $MAX2_l$ mod 2, you don't need to modify the value of $MAX1_l$, if $MAX1_l$ mod 2 is equal to $MAX2_l$ mod 2; you need to modify the value of $MAX1_l$. The rules are the same as you embed "1".

4. After wavelet coefficients are adjusted, the wavelet reconstruction is needed to do. The new watermarked audio signal is completed.

1.3.2 Watermark Extracting

Fig. 1.5 is the procession of the watermark extracting. The fragmentation procession is added to the detected audio signal. *L* level discrete wavelet transform is taken for each data segment of the part Fe. We find the absolute value of the maximum value and the second maximum value from each L-level low-frequency component, and the last nonzero number of which is extracted. If $MAX1_l$ mod 2 is equal to $MAX2_l$ mod 2, the extracted watermark information is "1"; otherwise the extracted watermark information is "0". A binary disorder watermark sequence can be obtained through these processions.

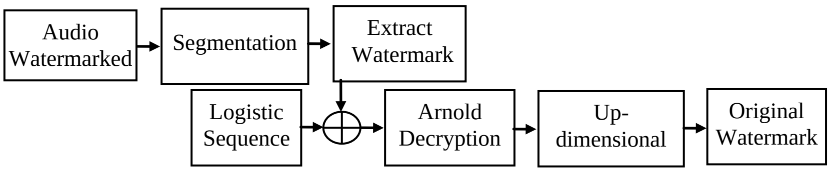


Fig. 1.5 Watermark extracting process

The logistic chaotic sequence is implemented by the key K （$u=4$, $x_0=0.8$） and select the former 1024 data. A series of binary disordered watermark sequence is obtained by taking *XOR* procession. The first decrypted one-dimensional watermark signal is made. Up-dimensional procession is added to this one-dimensional watermark signal and the two-dimensional watermark is obtained. The watermark, which we need, can be obtained.

1.4 Results and Analysis of Experiments

Two kinds of audio are used in the simulation. Its format is .wav, length is 5 seconds, sampling frequency is 22.05 *kHz*, and resolution is 16 bit. Watermark image is binary gray-scale image which size is *32*32*. Db4 wavelet is used for three-level wavelet decomposition.

Imperceptibility and robustness is mainly tested. By hearing tests, the audio signals have no significant difference with embedded watermark.

Fig. 1.6 and **Fig. 1.7** are pop and rock music with the watermarked audio waveform and coefficients. Watermarked audio signal is attacked by following attacks in robustness tests.

1. White Gaussian noise: Gaussian noise is normal distribution function, with 0.01 of expected value, 0.05 of variance, and it noise contains with embedded audio signals information;
2. Colored noise: Colored noise is achieved by a cut-off frequency of the low-pass filter of 10 *kHz*. It is generated and added to embedded audio signals information;
3. Re-quantization: 16 *bit* audio signal is quantized to 8 *bit* firstly, then is quantized to 16 *bit*;
4. Re-sampling: Up-sampling: Audio signal sampling frequency is become from 22.05 *kHz* to 44.1 *kHz*, then using extraction technology restore to 22.05 *kHz* of the original sampling frequency. Down-sampling: Audio signal sampling frequency is become from 22.05 *kHz* to 11.025 *kHz*, then using extraction technology restore to 22.05 *kHz* of the original sampling frequency.
5. Random cut: 10 positions are selected randomly from the embedded audio signals. 200 samples are cut at these positions. Then we take watermark extraction procedure.
6. Low-pass filtering: Butterworth low-pass filter's order is 6; and its cutoff frequency is 10 *kHz*. Low-pass filtering operation is taken for the audio signal with the watermark information.

Fig. 1.6 and **Fig. 1.7** show that audios have small changes with watermark embedded. Human ear is difficult to detect this different, and this algorithm has better imperceptibility.

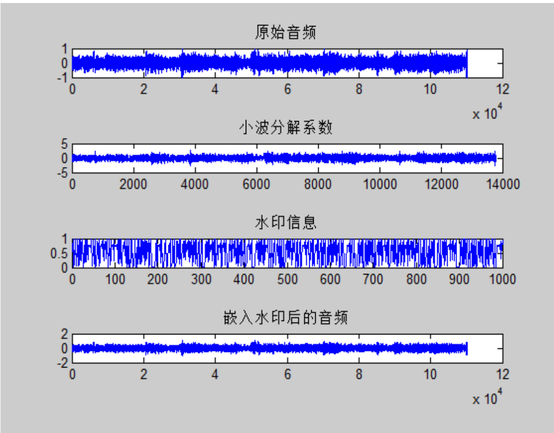


Fig. 1.6 Pop waveform comparison chart

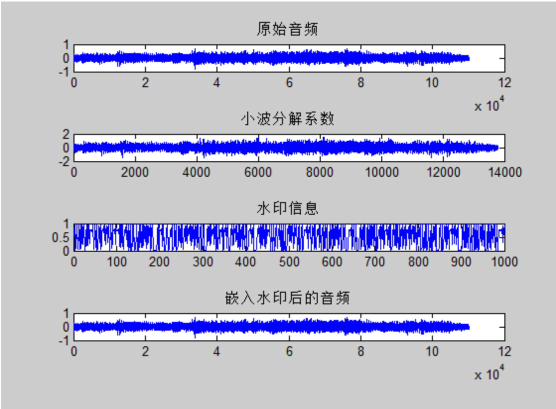


Fig. 1.7 Rock music waveform comparison chart

Table 1.1 Extracted watermark after various attacks

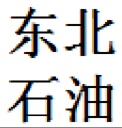
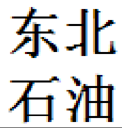
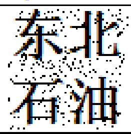
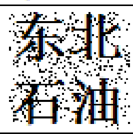
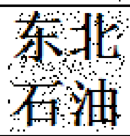
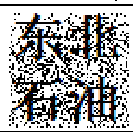
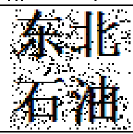
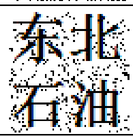
Attack Type	Music Type	
	Pop music	Rock music
No attack		
Gauss noise		
Colored Noise		
Re-quantization		
Up-sampling		
Down-sampling		

Table 1.1 (continued)

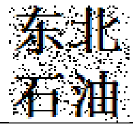
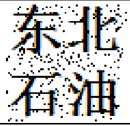
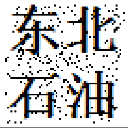
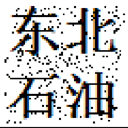
Random cut		
Low-pass filter-ing		

Table 1.2 Watermark correlation coefficient and contrast

Attack Type	Music Type			
	Pop music		Rock music	
	NC	*SNR*	*NC*	*SNR*
No attack	1.0000	34.45	1.0000	34.57
Gauss noise	0.8796	24.86	0.8814	25.03
Colored Noise	0.9035	24.57	0.9057	24.96
Re-quantization	0.7826	21.54	0.7819	21.68
Up-sampling	0.8568	27.66	0.8525	26.98
Down-sampling	0.7457	23.68	0.7521	24.81
Random cut	0.8515	21.79	0.8512	22.06
Low-pass filtering	0.9223	20.17	0.9247	20.45

The correlation coefficient and the signal to noise ratio is from **Table 1.1** and **Table 1.2**. The watermarked audio is attacked by common attacks; although the clarity of the extracted watermark is different, but they can still be identified, and that this algorithm has better robustness.

1.5 Conclusions

A blind audio watermarking algorithm by modifying the low-frequency coefficients of *DWT* is proposed. The watermark takes secondary encryption process, which is to improve the security of watermark. The algorithm is simple and fast. Simulation results show that audios have small changes with watermark embedded. Human ear is difficult to detect the different. The watermarked audio is attacked by common attacks; and the watermark extracted can be identified. It proves that this algorithm has better robustness. If the keys are not correct, the watermark information is not extracted correctly. The algorithm's imperceptibility is good, but its robustness is relatively poor. How to balance the imperceptibility and robustness is the direction of our future research work.

This work is supported by the Science & Technology Project of Heilongjiang province under Grant No. 12521056.

1.6 References

1. Chen Licong, Yao Zhiqiang (2005) Based on Local Extreme Points of the Audio Watermarking Algorithm. Computer Engineering and Applications, 41 (19): 41-43
2. Xu Dacheng, Shi Minghui (2002) DWT-based Digital Audio Watermarking Technology Research. In: Communications Technology, (11): 86-87
3. Zhang Zhijie, Ji Qichun, Wang Huiqin (2004) A New Digital Audio Watermarking Algorithm . Xi'an University of Architecture and Technology, (2): 203-206
4. Zhu Youzhong, Xiang Jun (2010) A Blind Audio Watermarking Algorithm Based on DWT and Relatively Small Values. In: Journal of Jilin University, (2): 203-208