Audio Zero-Watermarking Algorithm based on Beat Tracking

2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP) | 978-1-6654-3904-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSIP52628.2021.9688849

Xiaoman Dou
School of Software Engineering
Xi'an Jiaotong University
Xi'an, China
d434367594@stu.xjtu.edu.cn

Chen Li*
School of Software Engineering
Xi'an Jiaotong University
Xi'an, China
lynnlc@126.com

Lihua Tian
School of Software Engineering
Xi'an Jiaotong University
Xi'an, China
lhtian@xjtu.edu.cn

Abstract—**In order to protect the copyright of digital products effectively, digital watermarking technology is proposed. At present, synchronization attack is still a challenging problem in audio watermarking technology. In this paper, a robust zero watermarking scheme is proposed. Firstly, the beat activation function is obtained based on the bidirectional Long Short-Term Memory(BLSTM) recurrent neural network, and the beats of music are extracted by peak detection. Then the audio is segmented according to beat points. After framing for each segmentation, the mean value of each frame's normalized sampling values is calculated to obtain the mean value sequence. By encoding, the mean value sequence is converted into a binary sequence as the audio feature, which is xor with the watermarking image to generate the final zero-watermarking. Experiments show that the proposed scheme can effectively resist some synchronization attacks, such as TSM and jittering.**

Keywords—audio watermarking; beat tracking; synchronization attack

I. Introduction

The rapid development of digital multimedia technology and Internet technology makes the transmission and download of multimedia digital products such as images and audio become extremely convenient. But illegal copying, piracy and other phenomena also appear, which seriously infringe upon the legitimate rights and interests of the owners. The emergence of digital watermarking technology provides an effective method to solve copyright disputes[1], [2].

For the purpose of copyright protection, audio digital watermarking focuses on the robustness against various attacks, including common signal processing and synchronization attacks[3]. Noise, resampling, low-pass filtering and MP3 compression are all common signal processing. Synchronization attack can destroy the structure of audio signal watermarking. It requires very high synchronization mechanism of the algorithm. It is the most challenging attack method in the field of audio digital watermarking technology. Synchronization attacks include jittering and time scale modification(TSM). The attack is to increase or delete several sampling points in audio data so as to change the number of time-domain samples of audio. It makes the algorithm unable to extract the watermarking due to the deviation of audio synchronization point. TSM will cause the overall playback time of audio signal to be prolonged or shortened, leading to a great change in the time-domain structure of audio. Because many watermarking extraction algorithms rely on the pre-designed absolute position, and audio signal may appear different degrees of dislocation under synchronization attack, these algorithms may not be able to extract the watermarking. Compared with common signal processing, synchronization attack has great influence on the extraction of traditional watermarking algorithm, but has little influence on the auditory quality of signal[4].

Audio zero-watermarking is one of the developing directions of audio watermarking. It generates a secret key containing information about the host audio and the copyright without physical embedding, so the audio itself does not change. Thus the contradiction between robustness and imperceptibility is solved. S Choudhary proposed a double layer audio zero watermark based on DWT and DSSS, which has high security, but has obvious distortion to the image extracted after noise and low-pass filtering attack[5]. Wu proposed an adaptive blind watermarking algorithm, which dynamically adjusts the energy to determine the embedding strength, so as to adaptively embed watermarks into audio frames. But this algorithm has no strong robustness to MP3 compression[6]. None of the above methods are resistant to synchronization attack. Xiang used histograms to embed audio watermarks in pioneering research[7]. The algorithm based on redistributing the number of samples in adjacent boxes is proposed. The algorithm is effective for synchronization attack, but its robustness for ordinary signal processing needs to be improved. Lu proposed to extract feature points by using the second derivative, and the embedded and extracted fragments are centered on the detected feature points[8].

Since beat is an inherent feature of music, it should be robust to synchronization attacks. Therefore, this paper considers combining the beat and watermark segmentation to improve the robustness of the algorithm. In this paper, a robust zero-watermarking algorithm based on beat tracking is proposed, which can resist common signal processing and synchronization attacks.

This paper is organized as follows: Section 2 introduces the beat extraction algorithm. Proposed watermarking algorithm is

This work is supported by the National Natural Science Foundation of China under Grant No. 61901356 and the HPC Platform of Xi'an Jiaotong University.

described in Section 3. In Section 4, the experimental results are given. In Section 5, the conclusion is provided.

II. Beat Extraction Process

In recent years, various music beat analysis methods have been proposed. Among them, Bock's algorithm has outstanding performance in the accuracy of beat extraction due to the use of deep learning methods, so this article will use this method to extract beats[9]. The following describes the specific steps to extract the beat.

The audio data is transformed to the frequency domain via Short Time Fourier Transforms(STFT). The obtained magnitude spectrum and their first order difference are used as inputs to the BLSTM network, which produces a beat activation function. And then the autocorrelation function is used to determine the period of the beat, which is the rhythm. Finally, the beat sequence is obtained by detecting the peak value according to the rhythm. The flow chart of beat extraction is shown in Fig. 1.

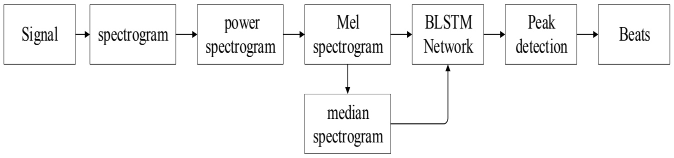


Fig. 1. Beat extraction flow chart

A. Feature Extraction

The sampling rate of the input audio signal is 44.1khz. For the input signal X(n), the window length was 23.2ms, 46.4ms and 92.8ms respectively, and the frame shift was 10ms. The spectrogram X(n,k) can be obtained by applying STFT transformation to the frame after framing and hamming window. The formula is as follows:

$$X(n,k)=\sum_{l=-\frac{W}{2}}^{\frac{W}{2}-1}\omega(l)\cdot x(l+nh)\cdot e^{-2\pi jlk/W} \tag{1}$$

Where n is the index of the frame, k is the frequency, h is the length of frame shift, and W is the window length. Then the energy spectrum is calculated according to the following formula:

$$S(n,k)=\left|X(n,k)\right|^2 \tag{2}$$

To reduce the dimension of the energy spectrum, 20 triangular filter banks equispaced on the Mel scale were used to transform the spectrum S(n, k) to the Mel spectrogram M(n, m).

$$M(n,m)=\log\left(S(n,k)\cdot F(m,k)^T+1.0\right) \tag{3}$$

where n is the index of the frame and m is the frequency in units of Mel. The first-order median difference of M(n, m) is calculated by the following formula:

$$D^+(n,m)=H(M(n,m)-M_{median}(n,m)) \tag{4}$$

$$M_{median}(n,m)=median\left\{M(n-l^*,m),\ldots,\ M(n,m)\right\} \tag{5}$$

Where, H(x) represents half wave rectification. The length of *l** depends on the window length W of STFT. *l** = W/100.

B. Network Model of Beat Extraction

The three M(n, m) of different window lengths and the corresponding three first-order median difference were input into BLSTM. The open source library called madmom is used here, which includes a pre-trained network model [10]. The fully connected network has three hidden layers in each direction, with 25 LSTM units each (6 layers with 150 units in total). The network is trained into a classifier by using the cross entropy error function. There are two units in the output layer, representing two classes: beat and no beat. We can simply think of the neural network as a black box, and the output beat activation function can be obtained by input as required. The independent variable of the function is time, and the dependent variable is the probability that this point is a beat point.

C. Peak Detection

The beat activation function obtained in the previous section may have redundant or missing beats, so further processing is required. The autocorrelation function is used to determine the rhythm, and the formula is as follows:

$$A(\tau)=\sum_n a_b(n+\tau)\cdot a_b(n) \tag{6}$$

The algorithm constrains the possible tempo range of the audio signal from T_{min} = 40 to T_{max}= 220 given in beats per minute. Evaluate to maximize the auto-correlation function as a rhythm, which represents the time interval between beats.

A part of sampling points are intercepted from around the initial beat point, and among them the sampling point p is found when the beat activation function takes the maximum value. Store p in the beat position array (a sequence of time points, each point represents a beat). Add the beat interval to the p to get the next point position. Perform the same operation as before until the point position exceeds the length of the audio segment, ending the recursion.

III. Watermarking Algorithm

A. Zero-Watermarking Generation

After extracting the music beats, segment between adjacent beat points and framing for each segmentation. Calculate the mean value of each frame to obtain the mean value sequence. Then convert the mean value sequence into a binary sequence with only 0 and 1 by encoding, and XOR with the watermark image to generate the zero-watermarking. After being attacked by TSM, it is difficult for most watermarking schemes to extract a complete watermark in the past. Because the TSM attack changes the time scale and the position of the extracted watermarking is misplaced, resulting in poor robustness of the extracted watermarking. After being attacked by TSM, our algorithm extracts the beat points as the basis for segmentation.

Although the absolute position of the beat points changes, the relative position does not change. The designed algorithm can effectively extract the watermarking and reduce the occurrence of dislocation. The generation process of zero-watermarking is shown in Fig. 2.

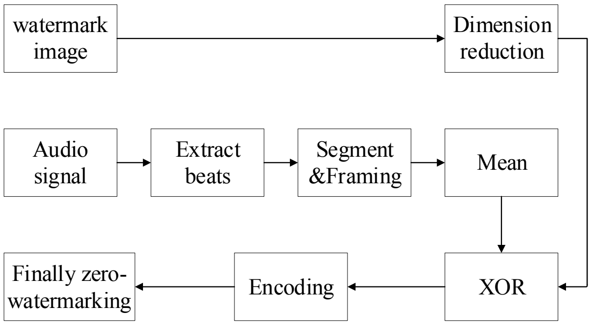


Fig. 2. Zero-watermarking generation process

1) Pretreatment: Input the audio signal, and extract the beat feature of the music audio according to the method described in the previous section. The unit of beat point is second. Input the watermarking image, convert the image into a binary image based on the threshold, and convert the image into a one-dimensional binary sequence S1.

2) Segment and Framing: A segment is divided between every two beat points. The window length is calculated according to the number of watermarking bits v_num embedded in each segment, and each segment is divided into non-overlapping frames with the window length. V_num is a user-defined adjustable parameter.

3) Calculate the mean value: Calculate the mean value of the audio signal normalized amplitude of each frame. The sample mean value of a piece of audio changes very little after the synchronization attack. Such as jittering, only periodically delete or add some sampling points, so as to change the number of audio samples, but the sampling mean value for each beat segment will not change much.

4) Encoding: Due to the stability of the mean, if an appropriate method is used to encode the mean sequence into a binary sequence, its stability can be enhanced. We use the K-means clustering method to encode the mean sequence, the cluster with the larger average value is coded as 1, and the other clusters are coded as 0, so as to obtain a binary sequence S2 representing audio features. During some attacks, the value of the mean sequence may change, but the overall mean sequence keeps a relative trend after the attack. Due to the advantages of K-means, our encoding method is very simple and can maintain the robustness of the encoded sequence to the greatest extent.

5) XOR: The following operation is performed where $\oplus$ means XOR. $Y=S1\oplus S2$, where Y represents the final zero-watermarking sequence.

In the end, the final zero-watermarking sequence Y and the relevant copyright information of the original audio signal are registered to the third party copyright protection center. In the future, copyright disputes can be resolved through this approach.

B. *Watermarking extraction*

The watermarking extraction process is similar to the generation process, as shown in Fig. 3.

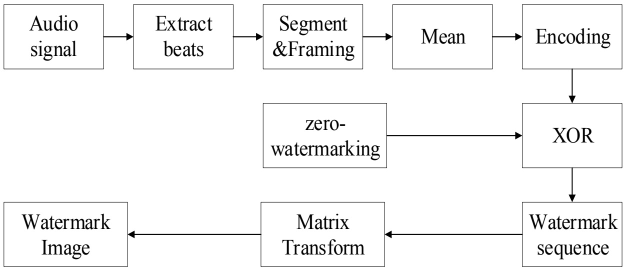


Fig. 3. Watermarking extraction process

Input the audio signal and extract the sequence of beats. Take the zero-watermarking sequence Y from the third party as input. Then segment and framing. Calculate the mean value of the audio signal amplitude of each frame. The mean value sequence is divided into two categories by K-means clustering, and then encoded into a binary sequence S2' representing audio characteristics. $W=S2'\oplus Y$. The watermark sequence W represents copyright. In the end, upgrading the watermarking sequence to generate a watermark image.

IV. Experimental Results

In order to verify the feasibility of our design scheme, two groups of music from different genres are selected for the test, and each group has 10 audio clips. The selected audio is music in WAV format with a sampling frequency of 44100 Hz. The size of the binary image to be embedded is 32*32, as shown in Fig. 4, which is converted into a one-dimensional sequence of 1024 bits.

Fig. 4. Watermark image

We set v_num to 128, which means that 128 bits are embedded in each beat segment. If it is set too large, the window length of each frame will be short, the mean value of each frame will be unstable, and the robustness will be reduced. If it is set too small, the window length of each frame will be longer. After taking the mean value, the gap between each frame will become smaller, and the result of clustering will become worse.

In order to measure the robustness of the algorithm, bit error rate (BER) and normalized correlation coefficient (NC) are used to evaluate. The formulas are as follows:

$$BER=\frac{number\ of\ error\ bits}{nunber\ of\ embedded\ bits}\times 100\% \qquad (7)$$

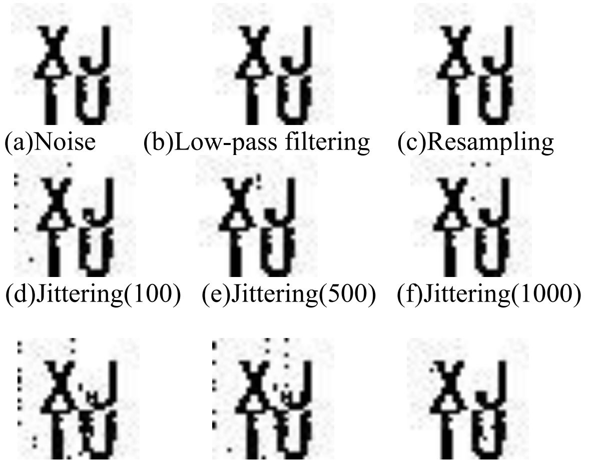
$$NC = \frac{\sum_{i=1}^{L} S_i \times S_i^{'}}{\sqrt{\sum_{i=1}^{L} S_i^2} \times \sqrt{\sum_{i=1}^{L} S_i^{'2}}} \quad (8)$$

where L is the length of the watermarking image sequence, S_i is the original watermarking sequence, and S_i' is the extracted watermark sequence.

Some attacks on audio signals are as follows:

- Noise attack: add 20dB Gaussian white noise in the audio signal.
- Low-pass filtering: the audio signal is passed through a low-pass filter with a cut-off frequency of 10kHz.
- Resampling: resampling the audio signal to 22.05kHz and then restoring it to 44.1kHz.
- MP3 compression: convert audio signals to MP3 format, and then convert them back to WAV format.
- TSM attack: Modify the time scale of the signal to lengthen or shorten the signal in the time domain.
- Jittering: copy or delete a sample point every certain number of sample points.

The extracted watermark image after the attack is shown in Fig. 5. BER and NC of watermarks after different attacks are shown in Table I.



(a)Noise (b)Low-pass filtering (c)Resampling

(d)Jittering(100) (e)Jittering(500) (f)Jittering(1000)

(g)TSM-10% (h)TSM+10% (i)MP3 compression

Fig. 5. The extracted watermark image

The lower the BER and the higher the NC value, the better the extraction quality of the watermark image. To some extent, even if the watermark image is destroyed, the human eye can still understand the content of the watermark image. Such as the watermark image shown in Fig. 5, we can still clearly identify the content of the image. The BER of the watermark is approximately 0 after common signal processing. After the TSM, the BER does not exceed 0.05, and the normalized correlation coefficient is above 0.97. After the jittering, the BER is close to 0, and the NC is close to 1. In other words, the proposed method has good robustness against some synchronization attacks. The rhythm of pop music is weaker than that of rock music, so the accuracy of beat extraction is relatively low, resulting in a decrease in algorithm robustness. So the robustness of our watermarking algorithm is better for music with stronger beats.

In Table II, we compare the BER of the proposed method with the methods in [8] and [11] under different attacks. All the three methods have good robustness in common signal processing. In the case of synchronization attack, the method [11] only mentions resistance to TSM attack of less than 4% due to the limitation of quantitative index modulation, while the method [8] can resist 10% TSM attack, but the BER is higher than that of our proposed method. When the audio is attacked by TSM, the absolute position of the beat point changes, but the relative position remains the same. In our method, the extracted audio features remain synchronized by extracting the beat point again . In addition, what we retain in coding is the transformation trend of the mean. Even if the mean changes slightly, the overall trend will not change significantly, so it can resist various attacks.

TABLE I. ROBUSTNESS TO DIFFERENT MUSIC GENRES

Attacks	Rock music		Pop music	
	BER	*NC*	*BER*	*NC*
AWGN (20db)	0	1	0	1
Low-pass filter	0	1	0	1
Resample	0	1	0	1
MP3 compression	0	1	0	1
TSM+1%	0.0273	0.9827	0.0312	0.9783
TSM-1%	0.0244	0.9846	0.0352	0.9776
TSM+10%	0.0361	0.9771	0.0439	0.9720
TSM-10%	0.0322	0.9795	0.0412	0.9745
Jittering(100)	0.0088	0.9945	0.0098	0.9938
Jittering(500)	0.0029	0.9982	0.0059	0.9963
Jittering(1000)	0.0029	0.9982	0.0020	0.9988

TABLE II. COMPARISON OF BER UNDER DIFFERENT ATTACKS

Attacks	Method in[8]	Method in[11]	Proposed method
AWGN(20db)	0	-	0
Low-pass filter	0	0.14	0
Resample(22.05kHz)	0	0.14	0
MP3 compression (128kbps)	0	0	0
MP3 compression (64kbps)	0	0.14	0.0039
TSM+1%	0.0625	0.19	0.0303
TSM-1%	0.1786	0.19	0.0244
TSM+10%	0.1518	-	0.0361

TSM-10%	0.3125	-	0.0322
Jittering(500)	0.125	-	0.0029
Jittering(1000)	0.125	-	0.0029
Jittering(1500)	0.0625	-	0

V. Conclusion

A robust audio zero-watermarking algorithm based on beat tracking is proposed in this paper. First, the music beats are extracted by the BLSTM network. The beat feature is used as the segmentation basis for watermarking generation, and the beat segment is divided into frames. The mean value of each frame is calculated, and then converted into a binary sequence by encoding. And the binary sequence XOR with the watermark image to generate a zero-watermarking. Finally, the zero-watermarking is uploaded to a third-party copyright protection center. Experiments show that the watermarking algorithm proposed in this paper can not only resist common signal processing attacks such as noise, low-pass filtering, resampling, and MP3 compression, but also has good robustness against synchronization attacks such as TSM and jittering.

Due to the limitation of the model for extracting the beat, the algorithm has strong robustness to music with strong rhythm, and the robustness of the watermarking algorithm is reduced for music with a weak rhythm. We will improve the accuracy of the beat extraction algorithm, and we will improve the watermarking algorithm to resist more malicious attacks in the future work.

References

[1] Y. Himeur and B. Boudraa, "Secure and robust audio watermarking system for copyright protection," Proc. 2012 24th International Conference on Microelectronics (ICM), Algiers, Dec. 2012, pp. 1-4, doi: 10.1109/ICM.2012.6471449.

[2] W. Li, X. Xue and P. Lu, “Localized audio watermarking technique robust against time-scale modification, ” IEEE Trans. Multimedia, Feb.2006, vol.8, no.1, pp.60-69, doi: 10.1109/TMM.2005.861291.

[3] Y. Xiang, I. Natgunanathan, D. Peng, G. Hua and B. Liu, "Spread Spectrum Audio Watermarking Using Multiple Orthogonal PN Sequences and Variable Embedding Strengths and Polarities," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 3, March 2018, pp. 529-539, doi: 10.1109/TASLP.2017.2782487.

[4] M. Fan and H. Wang, "Time-Scale Invariant Zero-Watermarking Scheme for Audio," 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, 2009, pp. 157-160, doi: 10.1109/IIH-MSP.2009.138.

[5] S. Choudhary, K. Nath and J. Panda, "Double layered audio zero-watermarking using DWT & DSSS," Proc. 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 0419-0423, doi: 10.1109/ICCSP.2017.8286390.

[6] W. Weina, "Digital audio blind watermarking algorithm based on audio characteristic and scrambling encryption," Proc. 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2017, pp. 1195-1199, doi: 10.1109/IAEAC.2017.8054203.

[7] S. Xiang and J. Huang, "Histogram-Based Audio Watermarking Against Time-Scale Modification and Cropping Attacks," in IEEE Transactions on Multimedia, vol. 9, no. 7, Nov. 2007, pp. 1357-1372, doi: 10.1109/TMM.2007.906580.

[8] W. Lu, L. Li, Y. He, J. Wei and N. N. Xiong, "RFPS: A Robust Feature Points Detection of Audio Watermarking for Against Desynchronization Attacks in Cyber Security," in IEEE Access, vol. 8, Mar. 2020, pp. 63643-63653, doi: 10.1109/ACCESS.2020.2984283.

[9] S. Bock and M. Schedl. "Enhanced Beat Tracking with Context-Aware Neural Networks," Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx), 2011.

[10] S. Bock, F. Korzeniowski, J. Schluter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," Proc. the 24th ACM international conference on Multimedia, Oct.2016, pp.1174-1178, doi:10.1145/2964284.2973795.

[11] I. D. Irawati, G. Budiman and F. Ramdhani, "QR-based Watermarking in Audio Subband Using DCT," 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia, 2018, pp. 136-141, doi: 10.1109/ICCEREC.2018.8712108.