

# Research on robust audio zero watermarking algorithm based on discrete cosine transform

Xiongwei Wu\* and Ting Fang

Nanchang Institute of Technology, Nanchang, China.

**Keywords:** Robust audio, Watermarking, Algorithm.

**Abstract.** This paper mainly uses the discrete cosine transform algorithm characteristics, combined with the "zero watermarking" embedding features, to improve the robustness of the watermark algorithm, to balance the uncoordinated relationship between robustness and transparency.

## 1 Watermark generation

This algorithm uses binary image as a watermark  $W = \{w(i, j) | 1 \leq i \leq a, 1 \leq j \leq b\}$ , The image size is  $a \times b$ . First, convert the image into one-dimensional sequence:

$$P = \{p(k) = w(i, j) | 1 \leq i \leq a, 1 \leq j \leq b, k = (i - 1) \cdot a + j\} \quad (1)$$

And then scramble the one-dimensional sequence  $P = \{p(k) | 1 \leq k \leq M\}$ , of which  $M = a \cdot b$ , get the embedded watermark sequence Q:

$$Q = \{q(k) | 1 \leq k \leq M\} = \text{Perm}(p), \quad (2)$$

Of which,  $\text{Perm}(\bullet)$  is the scrambling function, which is the watermark length, key1 as the key, save the Key1 for the watermark extraction.

## 2 Watermark embedding

### 2.1 Watermark embedding process

First, the algorithm will segment the original audio, DCT transform each audio segment, and extract the DC coefficient (DCC) of each segment, then, scramble the watermark, and sort the DC coefficients based on absolute value from big to small. The positive and negative feature of DC coefficients in DCT domain is not easy to change, so the positive and negative feature of the DC coefficients of the large absolute values are more stable. Therefore, in order to improve the robustness of the algorithm, orderly extract the direct and negative feature of

---

\* Corresponding author: [nclgdz@foxmail.com](mailto:nclgdz@foxmail.com)

the ranked DC coefficient, to obtain a set of binary feature sequences with the same length as the watermark, then perform the XOR operation in these feature sequence and the watermark bit, store the results as a key for the detection of the watermark, thus realizing the embedding of the zero watermark [1,4].

## 2.2 Embedding algorithm

Assuming that the original audio signal  $S = \{s(i) | i = 1, 2, \dots, N\}$  has an  $N$  sampling point, Each frame of audio contains an  $L$  sample point, the total frame count is  $U = \text{fix}(N/L)$ , ( $\text{fix}(\bullet)$  is the rounding function). To ensure that the audio length is sufficient to embed all watermarks, it requires  $U \geq M$ . The embedding steps for the watermark are as follows:

(1) segment the original audio  $S$ , and DCT transforms each audio to obtain the coefficients of the DCT domain  $f_k = \{f_k(i) | i = 1, 2, \dots, U\}$ .

$$S_k = \{s_k(i) = s_j(i) | i = 1, 2, \dots, 2k, U, j = 1, 2, \dots, 2k, L+i\} \quad (3)$$

$$f_k = DCT(S_k) \quad (k = 1, 2, \dots, U) \quad (4)$$

(1) Extract the DC coefficient in each DCT coefficient, i.e.  $f_k(1)$ , get the DC coefficient sequence  $f^{DC}$

$$f^{DC} = \{f^{DC}(k) = f_k(1) | k = 1, 2, \dots, U\} \quad (5)$$

(2) Sort  $f^{DC}$  by absolute value from large to small, and extract the  $M$  coefficients of higher absolute value, to get the DC coefficient matrix  $F^{DC}$ .

$$f^{DC} = \{f^{DC}(k) = f_k(1) | k = 1, 2, \dots, U\} \quad (6)$$

$$\text{key2} = \{l(k) | l \in (k)[1, u] = k, 1\} \quad (7)$$

$$F^{DC} = \{F^{DC}(k) = f^{DC}(l(k)) = k, 1, 2\} \quad (8)$$

Of which, the  $\text{sort}(\bullet)$  is sort function,  $DESC$  represents descending order, the element in  $\text{key2}$ , is the serial number  $F^{DC}(k)$  in  $f^{DC}$ , restore  $\text{key2}$  as the key to extract the watermark.

(4) Extract the positive and negative feature of each coefficient in  $F^{DC}$ , generate binary feature sequences  $H = \{h(k) | k = 1, 2, \dots, M\}$ , shown in the following formula:

$$h(k) \begin{cases} 1 & \text{if } F^{DC}(k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (k = 1, 2, \dots, M) \quad (9)$$

(5) Perform the XOR operation on watermark sequence  $Q$  and the feature sequence  $H$ , save the results  $\text{key3}$  as the key to extract the watermark:

$$key3 = Q \oplus H \quad (10)$$

### 3 Watermark extraction

Watermark detection is the inverse process of watermark embedding, which can be carried out according to the following steps:

(1) Assuming that the audio signal to be measured is  $\hat{S} = \{\hat{s}(i) | i = 1, 2, \dots, N\}$ , divide the  $\hat{S}$  into U frames, Each frame of L sample points.

(2) DCT transform for each frame of audio to obtain  $\hat{f}_k$ , extract the DC coefficients of each frame to obtain:

$$\hat{f}^{DC} = \{\hat{f}^{DC}(k) = \hat{f}_k(1) | k = 1, 2, \dots, M\} \quad (11)$$

(3) using the key  $key2$ , get feature sequences of audio  $\hat{H} = \{\hat{h}(k) | k = 1, 2, \dots, M\}$

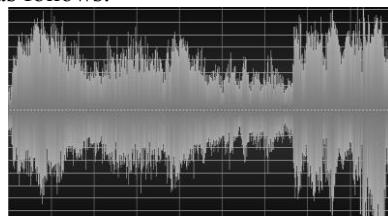
$$\hat{h}(k) = \begin{cases} 1 & \text{if } f^{DC}l(k > 0) \\ 0 & \text{otherwise} \end{cases} \quad (k = 1, 2, M) \quad (12)$$

(4) perform the XOR operation on key  $key2$  and feature sequences  $\hat{H}$ , extract the watermark sequences  $\hat{Q}$ , as shown in formula (4-13). Then use the  $key1$  to perform the adverse scrambling, get a one-dimensional sequence  $\hat{P}$ . Finally, extend the  $\hat{P}$  into the two-dimensional image  $\hat{W}$ , then the  $\hat{W}$  is the watermark image we extracted.

$$\hat{Q} = key3 \oplus \hat{H} \quad (13)$$

### 4 Simulation results and analysis

In the simulation experiment of this chapter, we use Stir mark for Audio V02 to variously deal with of the music and speech embedded the watermark respectively to simulate the attack of audio in the actual application [4]. Using mono-channel, and use music and voice sections with sampling rate of 1kHz as the original audio signal, audio length of 100s, take a binary image as a watermark, the length of each frame is l=20. In this section, the transparency and robustness of the algorithm are simulated and analyzed. The original audio and the original image are as follows:



**Fig. 1.** Original audio image.

#### 4.1 Transparency test

We adopt the standard based on signal-to-noise ratio (SNR) to measure the audio signal quality of embedded watermark. The formula for SNR is as follows:

$$SNR = 10 \log \frac{\sum_k s^2(k)}{\sum_k [s(k) - s'(k)]^2} \quad (14)$$

In the formula,  $S(k)$  and  $S'(k)$  respectively represent the original audio signal to be embedded with the watermark part and the audio signal embedded with the watermark part[2].

In order to prove the feasibility of audio watermarking, in the transparency test, we do not choose to use the above mentioned audio as the original audio, instead, we choose classical music, pop music, jazz and rock music as the original file, to test the transparency[1,3].

According to the test, the waveform file is almost no difference, in terms of signal-to-noise ratio, the largest SNR value of its classical music is 32.6863dB, the lowest SNR value of rock music is 29.5657dB.

Because this algorithm uses zero watermark embedding method, the watermark embedding does not bring the original audio sampling point coefficient change. As a result, the quality of audio is almost unaffected. The calculated algorithm, that is, the signal-to-noise ratio is infinitely large.

#### 4.2 Robustness test

In the robust simulation experiment of this algorithm, perform the adverse watermark on all kinds of attacked audio and display the restored watermark image. Simultaneously calculates the NC and BER under various attacks. The bandwidth calculated according to

$Bandwidth = \frac{M}{N} \cdot Fs$  is  $Bandwidth = 200bps$ , far exceeding the minimum requirement of 20bps of IFPI to the bandwidth.

In order to highlight the robustness of the algorithm, we will compare the robustness between this algorithm and that which is presented in the "DCT Domain Audio Watermark: Watermark Algorithm and Non-Perceptual Test" published in the electronics newspaper by the Wen Quan, Wang Shuxun, and Nian Guijun. Many experiments show that to achieve better robustness, the bandwidth of embedded watermarking needs to be reduced[1]. So in this experiment, using music and voice sections with 100s as the original audio signal, using a 100-bit binary sequence as a watermark. The parameters compared with the algorithm in the reference literature are set as follows: 441 sampling points per frame, the F(1) is divided into 100 sequences, the length of each sequence is 100, and the embedding strength d=20. This algorithm sets the frame length of L = 20. According to the formula

$$Bandwidth = \frac{M}{N} \cdot Fs,$$

The bandwidth of this experiment can be calculated as Bandwidth=1bps.

### Summary

The simulation results show that under the same embedded bandwidth, the algorithm has better robustness and better transparency under most attacks. The relationship between

transparency, bandwidth and robustness is also well balanced. It is proved that this algorithm achieves the desired design goal well.

## References

1. Wang Bingxi,Chen Qi Digital Watermarking Technology [M] Xi'an University of Electronic Science and Technology Press 2003
2. Huang Wu,Cheng Weidong DCT Domain Image Watermarking: Embedding Countermeasure and Algorithm [J] Electronic Journals 2000.28
3. Arnold M. Audio watermarking: Bury Information in The Data [J]. Dr. Dobb's Journal, 11
4. Fu Xiaojuan,Yang Wanquan Speech Signal Compression Scheme Using Discrete Cosine Transform [M] Information Technology 2006.11