Computers and Electrical Engineering 106 (2023) 108580

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Advanced audio coding steganography algorithm with distortion minimization model based on audio beat☆

Xue Zhang, Chen Li *, Lihua Tian

School of Software Engineering, Xi'an Jiaotong University, Xi'an, China

ARTICLE INFO

Keywords:
Minimal distortion
Content-adaptive embedding
The small value area
AAC

ABSTRACT

Currently, most advanced audio coding (AAC) steganography methods are content-non-adaptive without considering the characteristics of audio, and there are several limitations in imperceptibility and steganalysis. In this paper, we use an audio feature beat as the anchor point to identify the cover elements, group the quantized modified discrete cosine transform (QMDCT) coefficients in the small value area, and finally use the syndrome-trellis codes (STCs) framework for content-adaptive embedding to obtain the minimum distortion. In the STCs framework, we comprehensively consider auditory and data distortions. Experimental results demonstrate that the proposed steganography algorithm has a 10% improvement over the compared algorithms in terms of imperceptibility and steganalysis, and it can accurately extract secret information in face of frame loss and misalignment.

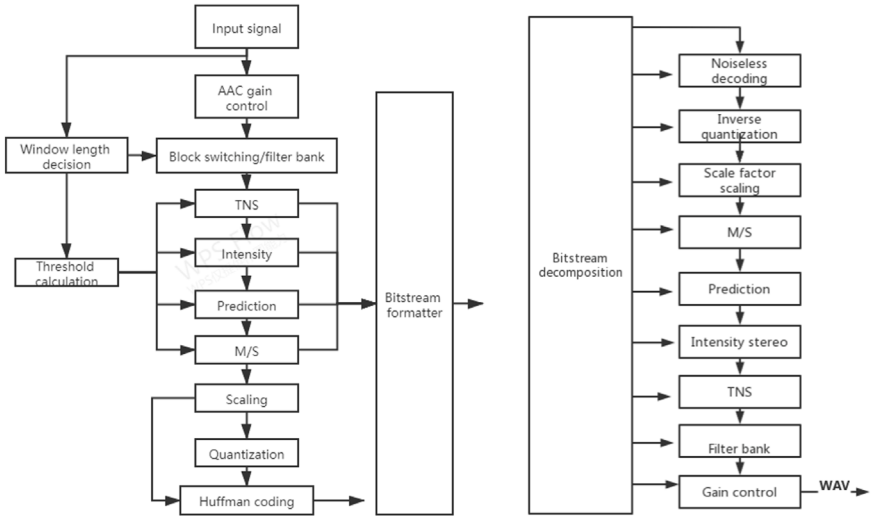
1. Introduction

Currently, advanced communications and networks have significantly enhanced user experiences, and they have a major impact on all aspects of people's lifestyles in terms of work, society, and the economy [1]. Steganography is a technology that embeds secret information in cover data and transmits it. An excellent steganography algorithm should prevent the monitor from discovering anomalies and achieve secret communication [2]. Currently, popular steganography covers include digital images and digital media. Because human hearing is more sensitive than vision, slight embedding introduces noise to the audio, thus, the embedding location of digital media is limited and steganography of digital media is more difficult [3]. As more than 90% of the digital media distributed online are music, it is important to study the information hiding of audio. Current studies on steganography algorithms for audio focus on uncompressed audio. However, uncompressed audio is relatively large and inconvenient for transmission to the network, and is rarely used in real life. The method in [4] has proposed a visual audio steganography model based on convolutional neural network (CNN), it brings a new direction to steganography, but this method performed discrete wavelet transform(DWT) transformation on the sample points and then embedded information while discrete cosine transform(DCT) transformation is selected in the AAC operation. DWT transformation is obtained by discretizing the scale and displacement of the continuous wavelet transform according to the power of 2 while DCT transformation divides elements into small blocks of different frequencies, which are then quantized. During the quantization process, high frequency components are discarded, and the remaining low frequency components are preserved for later reconstruction. The method in [4] is currently not available for AAC audio. The method in [5] has proposed a novel LSB-BMSE method that enhances LSB audio steganography. But in the process of AAC encoding, the sampling points need to be quantized, lsb-based algorithms would lose precision during the quantization process, resulting in the failure of secret information

☆ This paper is for regular issues of CAEE. Reviews processed and recommended for publication to the Editor-in-Chief by Prof. H. Huimin Lu.

* Corresponding author.
E-mail address: lynnlc@126.com (C. Li).

https://doi.org/10.1016/j.compeleceng.2023.108580
Received 16 February 2022; Received in revised form 29 August 2022; Accepted 4 January 2023
Available online 16 January 2023



(a) AAC encoding flowchart (b) AAC decoding flowchart

Fig. 1. AAC encoding and decoding flowcharts.

embedding. Compressed domain audio has gradually become a popular steganography carrier owing to its better auditory effects and smaller volume, among which the advanced audio coding (AAC) and MP3 formats are the most widely used. AAC is an advanced audio coding standard proposed based on MP3 [6]. It uses a brand-new algorithm for coding and decoding, which is more efficient and has better sound quality and smaller volume than MP3, making it an ideal steganographic carrier. In summary, with the rapid development of network technology, effective and secure management of work on the internet is important [7]. Therefore, research on AAC audio steganography is necessary [8]. The AAC audio coding process is illustrated in Fig. 1(a). The decoding process of AAC is the inverse of encoding, as shown in Fig. 1(b). The main contributions of this paper are as follows:

1. A new distortion function of the syndrome-trellis codes (STCs) framework is proposed to ensure the operation of the minimum distortion model. The distortion function considers both auditory distortion and data distortion of embedded elements during AAC encoding, thus, the algorithm performs better in anti-steganography.

2. In this paper, the quantized modified discrete cosine transform (QMDCT) coefficients in the small-value area are the candidate cover elements, and slight changes to them do not cause auditory loss to the audio. Therefore, the imperceptibility of the algorithm is improved.

3. A beat point extraction method based on deep learning is applied to the steganography algorithm. The extracted beat points are used to group and filter candidate cover elements such that the algorithm can still effectively extract hidden information in the case of frame loss.

According to the different locations used to embed private messages, steganography algorithms can be divided into two situations: before and after quantization embedding. The current before-quantization embedding methods include the least significant bit embedding algorithm [9], echo hiding-based embedding algorithm [10], a modification of the MDCT coefficient algorithm [11]. In the algorithm [9], an audio watermarking algorithm based on the least significant qubit is proposed. The algorithm in [10] embeds a time-delay sequence in the main signal. Ren [11] modifies the spectral coefficients to embed secret information by calculating the difference between them before quantization. However, the spectral coefficients should go through the quantization module. Quantization means quantizing the floating-point spectral coefficients into integer spectral coefficients and it reduces the accuracy, therefore, embedding the spectral coefficients before quantization is not recommended. Generally, the before-quantization embedding algorithms are not recommended.

After-quantization embedding algorithms embed private messages after the quantization module. Mostly, they modify the QMDCT coefficients, including those in [12,13] and [14], and some embeds messages in AAC bitstreams [15] or in the sign bits of the Huffman codeword [16]. The algorithm in [15] embeds secret messages into the frames of AAC bitstreams, and uses a secure hash algorithm 1 pseudo-random generator to randomly choose frames for embedding. The algorithm in [16] embeds secret messages using the characteristics of the sign bits of the Huffman codeword. However, conventional hashing methods usually represent content by handcrafted features [17], AAC encoding has strict requirements, and changes in the bitstream or Huffman codeword may cause the encoding to fail. In method in [14], nonzero QMDCT coefficients are selected in each frame to embed secret information with the STCs framework. The algorithm performs well in steganalysis, but the QMDCT coefficients are changed at low frequencies, which significantly impacts the audibility of the audio. Additionally, when calculating the data distortion of the cover element, the algorithm considers only the influence of the previous and the last elements, which is not sufficiently comprehensive.The method in [13] has proposed a large capacity secure steganography algorithm based on the compression parameters of AAC QMDCT. This

method sacrificed audio audibility for improving embedding capacity, the sound quality of the stego music was worse after the secret information was embedded, which was not suitable for types of audio such as music. In [12], a genetic algorithm is used to modify the QMDCT coefficients located in a small area to embed a secret message. For the QMDCT coefficients in the small value area, every four QMDCT coefficients form one group for encoding during the Huffman coding, and the algorithm calculates the parity of the group. When the parity is different from the parity of the secret message, the QMDCT coefficient is added or subtracted by 1, and the optimal sequence for adding or subtracting 1 is determined according to the genetic algorithm. The algorithm performs well in terms of imperceptibility, but the genetic algorithm is not adaptive. The search for the sequence of adding or subtracting 1 is not the optimal sequence, therefore, the effect of embedding cannot be guaranteed. Based on the exposed, imperceptibility and steganalysis can be further improved. In [12] and [14], the QMDCT coefficients are adjusted to embed a secret message. Because the Huffman coding and bit stream in the AAC coding process are lossless operations, they do not have any effect on the secret message, which can guarantee the accuracy of extracting the secret message.

In summary, we should select the QMDCT coefficients in the small value area for adaptive minimum distortion embedding. A more comprehensive data distortion is designed in the STCs framework to ensure the embedding effect. In addition, when audio is divided into frames during transmission and storage, overlapping data occur between frames. Frame drop and misalignment may occur during the timefrequency conversion. Moreover, embedding secret messages inevitably introduces noise to the audio, which also causes audio frame drop and frame misalignment. To solve these issues, audio beat points are used as anchor points. In this paper, we design more comprehensive data and auditory distortions and adopt the STCs framework to embed secret messages adaptively. The proposed algorithm guarantees the capacity of secret message embedding, has good imperceptibility and steganalysis, and can effectively solve the problem of frame misalignment.

The remainder of this paper is organized as follows: The specific process of the beat tracking algorithm and the STCs framework are introduced in Section 2. Specific embedding and extraction processes are presented in Section in 3. The experimental results of the algorithm are presented in Section 4. Finally, discussions on future directions of this paper are presented in Section 5.

2. Related technology

2.1. Beat tracking algorithm based on deep network

In music, time is divided into equal basic units, and each unit is called a “beat”, which is expressed in the regular and periodic recurrence of upbeats and downbeats. Beats are a unique characteristic of each audio. Conventional beat tracking algorithms include beat tracking methods based on music knowledge [18] and a symbolic model [19]. In [18], MIDI is chosen as the input signal, using multiple agent competition methods to infer and track the beat from the inter-onset intervals. In [19], a music hierarchy is designed and a bottom-up method is used to obtain the accent of the music and find the beat. The conventional beat-tracking algorithm is relatively complicated, the use of scenes is limited, and the accuracy is low. With the development of neural network technology, a probability framework adopting the “reverse” viterbi algorithm to estimate the probability of the downbeat and using machine learning methods to estimate the most discerning beat linearly is proposed in [20]. In [21], the activations of a unidirectional recurrent neural network are fed into an enhanced Monte-Carlo localization model to infer beat positions, but the ‘do not look back’ algorithm is used, which may miss beats. In [22], a bilateral recurrent network is combined with long short-term memory neurons to form a bilateral long- and short-term memory cyclic neural network that can model the time context of the input data, eliminate erroneously detected beats, and replenish missing beats. Because the process is simple and efficient, and the accuracy of beat points is better than that of other current beat tracking algorithms, we choose the Madmom framework based on [22] to track beats.

2.2. STCs framework

The current steganography methods based on AAC are not adaptive to audio content, therefore, the algorithm works better in one style of audio but works poorly in another style. The STCs framework, which depends on the content of the cover elements for embedding, can effectively solve this problem and also the problem of minimal distortion. Different adjustments can be made according to the value of each cover element with a certain embedding capacity, thus, the overall distortion of the embedding can reach the theoretical minimum [23,24]. The method in [25] specially designed an adaptive parity-check matrix to replace the designed distortion cost to restrict the embedding position. However, when embedding secret information, each audio needed to undergo a lot of computation due to the flexible selection of parity check matrix width, the effect of AAC encoding needed further study. Normally overly complex up-front computation would lead to a lack of practicality.

Under the assumption that each cover element is independent of the others, we can define the overall distortion of embedding secret information as the sum of the embedding cost of each individual cover element. Suppose that the binary vector $X = (x_1, x_2, \dots, x_n) \in 0,1^n$ is the vector of cover elements, $Y = (y_1, y_2, \dots, y_n) \in 0,1^n$ is the vector of stego elements, and $m = (m_1, m_2, \dots, m_k) \in 0,1^k$ is the binary vector of the secret message. The additive distortion function can be defined according to Eq. (1). Package hyperref Warning: Token not allowed in a P

$$Package hyperref Warning : Token not allowed in a PD(X,Y) = \sum_{1}^{n} C(x_i, y_i), \tag{1}$$

where C(xi,yi) represents the cost of changing the ith cover element from $x_i to y_i$.

The goal of the STCs framework is to obtain the Y sequence, which makes the value of $D(x, y)$ the smallest. This process is realized using the parity matrix[18,19] and the Viterbi algorithm. The parity matrix H consists of the sub-check matrix h' arranged in sequence in the order of the main matrix. The sub-parity matrix h' has h rows and w columns, where $h \in [2, 32]$; considering coding complexity and computer computing power, usually $h \in [6, 15]$. w depends on the embedding rate(ER).

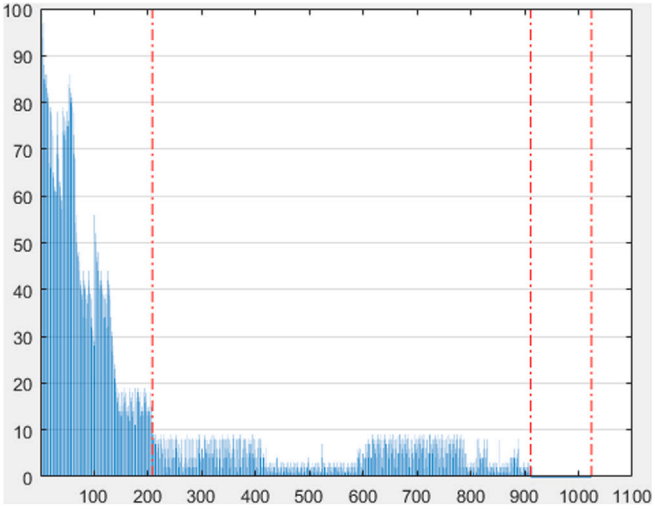


Fig. 2. MDCT coefficient diagram.

Table 1
Huffman codebook.

Codebook number	Number of data	*Maximum absolute value*
0	/	0
1 2	4	1
3 4	4	1
5 6	2	4
7 8	2	7
9 10	2	12
11	2	16(ESC)

3. Proposed algorithm

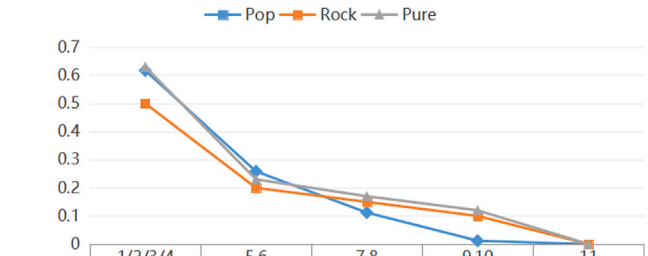
3.1. Determine the cover elements

In this paper, a steganographic embedding algorithm based on the AAC low-complexity (LC) framework is proposed, and other frameworks can be processed similarly. During AAC encoding, the MDCT coefficients are arranged from low to high frequency after the time–frequency conversion. Herein, we test 10 audio clips with duration of 10 s, and calculate the average value of the MDCT coefficients using their 4th to 10th frames. As shown in Fig. 2, the abscissa represents the order of 1024 coefficients in a frame, and the ordinate represents the MDCT coefficient value. As observed, the MDCT values of at low frequencies are larger, and remained stable as the frequency increases. Finally, they tends to be 0 in the high-frequency part, therefore, we can divide the MDCT coefficient values into large, small, and zero value areas [26]. The MDCT coefficients located in the large value area contain most of the audio energy. As a slight modification would reduce the quality of the audio, it is not suitable for information hiding. The MDCT coefficients located in the zero-value area do not participate in encoding. Thus, modifying coefficients in this area is likely to cause embedding failure, thus, this area is not suitable for information hiding. The MDCT coefficients located in the small value area cover the smaller energy of the audio and they have a relatively stable value. Therefore, a small change would generally not cause auditory loss. This area has good imperceptibility and can resist steganalysis, therefore, it is suitable for information hiding.

The QMDCT coefficients should be noise-free coding to further reduce data redundancy and increase compression efficiency. Codebook selection is required when performing the Huffman coding. The AAC standard contains 12 codebooks, as listed in Table 1. The maximum absolute value of the QMDCT coefficients in the small-value area is two, which corresponds to the MDCT coefficients in codebooks 1, 2, 3, and 4. Because four coefficients are encoded as a group in the codebook 1/2/3/4, the codeword index value is calculated according to Eq. (2), and the corresponding codebook is encoded according to the index value [26].

$$I = 27 \times X1 + 9 \times X2 + 3 \times X3 + X4 + 40 \tag{2}$$

where I is the codeword index; $X1$, $X2$, $X3$, and $X4$ are the first, second, third, and fourth coefficients in a set of quantized coefficients, respectively. As observed, the change to $X4$ has the least impact on the final codeword index value, and has a small overall impact on the audio. Therefore, when choosing the cover elements, the fourth QMDCT coefficient of each group should be selected. The Huffman codebook is listed in Table 1.



	1/2/3/4	5,6	7,8	9,10	11
Pop	0.6169	0.2586	0.1121	0.0122	0.0001
Rock	0.5	0.2	0.15	0.1	0.0003
Pure	0.63	0.23	0.17	0.12	0.0003

Fig. 3. Usage rate of codebooks.

Three styles of music and ten pieces of each style with duration of 10 s are tested, and the usage frequency of their codebooks is calculated. As shown in Fig. 3, the usage rate of codebook 1/2/3/4 is significantly higher than that of the other codebooks. Therefore, this codebook has a sufficient embedding space.

3.2. New distortion function

In the STCs framework, the determination of the distortion function is particularly crucial. To ensure the overall quality of audio, the distortion function definition in this paper comprehensively considers auditory and data distortions. A psychoacoustic model is used to estimate the auditory distortion function, as it can estimate the maximum allowable distortion value of the audio signal. The calculation equation is as follows:

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4, \tag{3}$$

where f is the frequency of the audio signal.

After obtaining the auditory distortion of the cover elements, the chosen cover element is the fourth element of each group in the Huffman code. Therefore, the data distortion of each cover element should comprehensively consider the effects of the other three QMDCT coefficients in the group and the next QMDCT coefficient of the cover element. Data distortion is defined according to Eq. (4).

$$D(Z_i^j) = \sum_{k\in K} (|Z_i^{j+1}| + |z_i^k| + \alpha)^{-1}, \tag{4}$$

where $D(Z_i^j)$ represents the data distortion of the jth element of the ith frame, and $|Z_i^{j+1}|$ represents the absolute value of the j+1-th element of the ith frame. $|z_i^k|$ belongs to set K, which represents the set of the other three elements in the Huffman coding group. α is the adjustment parameters, which is determined experimentally.

After obtaining the auditory and data distortions, we have combined the auditory and data distortions to accurately obtain the overall distortion of each cover element. For a precise experiment, we first normalize the auditory distortion to min–max. The overall distortion function is given according to Eq. (5)

$$L(Z_i^j) = D(Z_i^j) - log_2 T_q'(f) + \beta, \tag{5}$$

where $L(Z_i^j)$ is the overall distortion of the jth element of the ith frame and β is a fixed value to ensure that $L(Z_i^j)$ is always positive, the specific value is determined experimentally.

3.3. Embedding process

The cover elements are modified to embed secret messages. The embedding process is illustrated in Fig. 4. The specific steps are as follows:

(1) The secret message is preprocessed. To increase confidentiality, the original binary sequence of the secret message is transformed. Suppose that the number of cover elements is C and the length of the binary sequence of the secret message is M. According to the STCs framework described, it should be guaranteed that $C > M$.
(2) During AAC encoding, when the Huffman encoding module obtains the fourth QMDCT coefficients of each group in codebook 1/2/3/4 and their overall distortion, the pre-cover element set is obtained.
(3) The audio is input to the Madmom framework to obtain n beats of the audio: $R = \{R_1, \ldots, R_n\}$, and the locations of the QMDCT coefficient corresponding to each beat are calculated: $P = \{P_1, \ldots, P_n\}$. Notably, the QMDCT coefficient corresponding to the beat does not necessarily belong to the pre-cover element set, in this case, the nearest QMDCT close to the QMDCT that belongs to the set of pre-cover elements should be selected to replace the anchor point. The n anchor points divide the

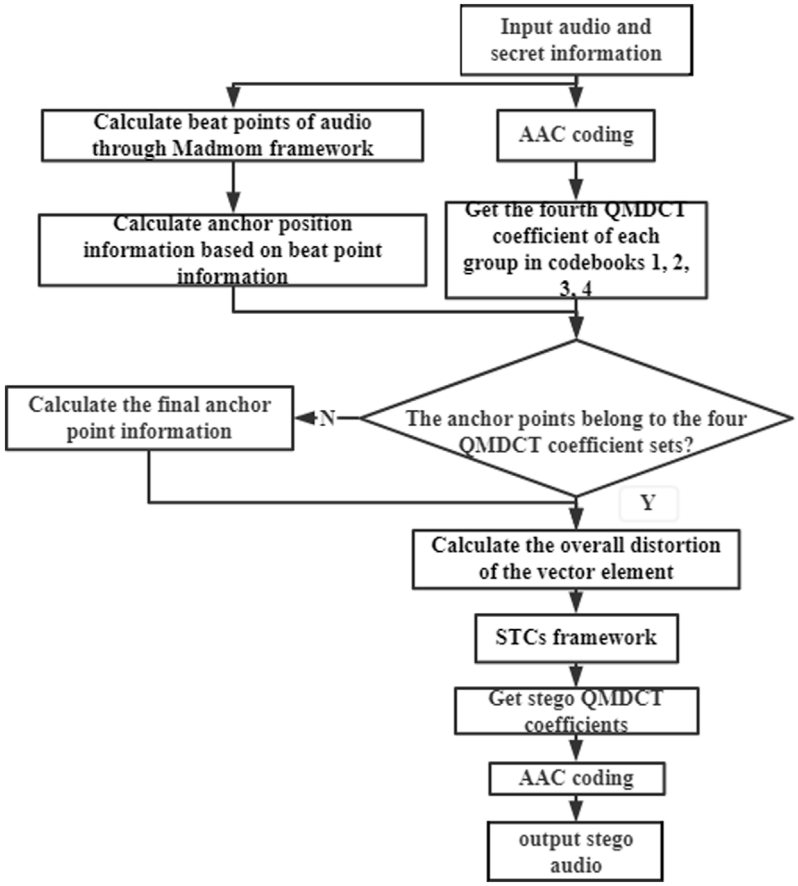


Fig. 4. Flowchart of secret information embedding.

pre-cover elements into unequal $n+1$ groups, and the number of pre-cover elements in each group is $S = \{S_1, \ldots, S_{n+1}\}$. The rules for determining the final cover element set are as follows.

A. Calculate and obtain the $Smax$ group, where the largest number of cover elements exists in this group. All elements of this group are included in the final cover element set, $Cover$.
B. The rule of selecting the remaining n groups is as follows: In the ith group, $D_i = [S_i * (\frac{M}{C})]$ elements need to be contributed to the $Cover$. When $D_i \leq \frac{S_i}{2}$,every other element is chosen into the $Cover$; otherwise it directly chooses elements of this group in sequence into the $Cover$.

(4) The LSB bits of the cover elements in the $Cover$ and the corresponding overall distortion are obtained and sent to the STCs framework. Thereafter, the stego coefficients are obtained and sent back to the AAC encoding. Finally, the stego audio of the AAC format is obtained.

3.4. Extraction process

The extraction process is completed using AAC decoding. During AAC decoding, after the corresponding codebook is selected, only the index information in the codebook can be obtained. Thus, the index must be decomposed into a set of four quantization coefficients. The extraction process is illustrated in Fig. 5. When AAC decoding proceeds to Huffman decoding, codebook 1/2/3/4 is selected, the index value in the codebook is obtained, and the group of quantized coefficients is obtained according to the decomposition method. The specific steps of the extraction are as follows:

(1) During AAC decoding, we obtain the fourth QMDCT coefficient of each group in codebook 1/2/3/4 to form a set of pre-cover elements;
(2) The stego audio is input into the Madmom framework to obtain n beat points of the audio: $R = \{R_1, \ldots, R_n\}$. The position of the QMDCT coefficient corresponding to each beat point is calculated according to the sampling rate: $P = \{P_1, \ldots, P_n\}$,

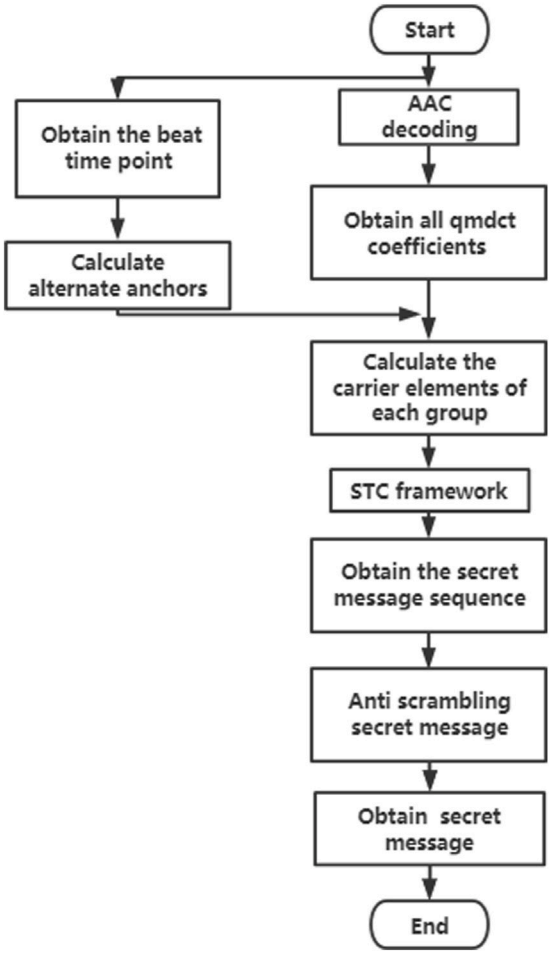


Fig. 5. Flowchart of secret information extraction.

similar to step (3) of embedding, after calculating the position of the QMDCT. If it is an element that does not belong to the pre-cover element set, the nearest QMDCT located in the pre-cover element set as the anchor point is selected. The n anchor points divide the cover elements into unequal $n+1$ groups, and the number of QMDCT in each group is $S = \{S_1, \ldots, S_{n+1}\}$. The rules for choosing the cover elements in each group are as follows:

A. Calculate and obtain $Smax$ group, where the largest number of cover elements exists in this group. All elements of this group are included in the final cover element set, $Cover$;
B. The rule of selecting the remaining n groups is as follows: In the ith group, $D_i = [S_i * (\frac{M}{C})]$ elements need to be contributed to the $Cover$. When $D_i \leq \frac{s_i}{2}$, every other element is chosen into the $Cover$; otherwise, it directly chooses elements of this group in sequence into the $Cover$.

(3) The stego QMDCT coefficients are input into the STCs framework and the secret message is obtained.

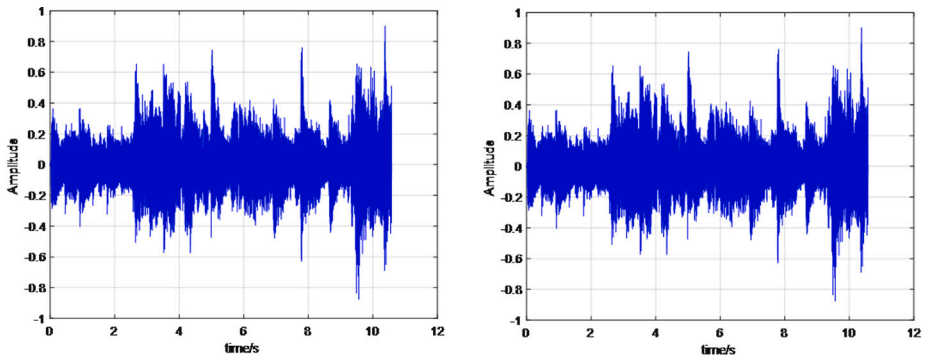
4. Experiment and analysis

4.1. Experiment environment

In this paper, related experiments are conducted on four aspects: embedding capacity, imperceptibility, steganalysis, and integrity of secret information. The experiments are conducted on a public dataset containing 1000 stereo WAV files with different types of music, including 330 pop, 330 rock, and 340 classic songs [27]. The sampling rate is 44.1 KHz, and each test audio is 10 s and two-channel. The Faac-1.28 and faad-2.27 frames is used for encoding and decoding, respectively. Visual Studio 2015 is used to embed and extract the secret messages. Comparative experiments are presented using MATLAB 2021b.

Table 2
Integrity of extracted message.

Bit rate	64 kb/s	128 kb/s	192 kb/s
[12]	0.95	0.96	0.968
[14] $ER = 0.5, w = 2$	0.95	0.96	0.99
[14] $ER = 0.5, w = 4$	0.97	0.987	0.994
[14] $ER = 1, w = 2$	0.96	0.978	0.996
[14] $ER = 1, w = 4$	0.98	0.984	0.992
Proposed($ER = 0.5, w = 2$)	1	1	1
Proposed($ER = 0.5, w = 4$)	1	1	1
Proposed($ER = 1, w = 2$)	1	1	1
Proposed($ER = 1, w = 4$)	1	1	1



(a) Cover audio waveform (b) Stego audio waveform

Fig. 6. Comparison of waveforms.

4.2. Integrity of secret information

During decoding, the secret message is extracted and the normalized correlation (NC) coefficient is used to calculate the integrity of the secret message. NC describes the difference between the extracted and original secret message. When the value is 1, there is no difference. The calculation equation is as follows:

$$NC = \frac{\sum_{i=1}^{N} X_i \times X_i'}{\sqrt{\sum_{i=1}^{N} X_i^2} \times \sqrt{\sum_{i=1}^{N} X_i'^2}}, \tag{6}$$

where X_i and X_i' represent the original and extracted secret messages, respectively. N represents the length of the secret message. The value of NC is between 0 and 1. When it is 1, the extracted secret message is exactly the same as the original message. To verify the effectiveness of the proposed algorithm in correctly extracting the secret message in the face of frame misalignment and frame loss, we have removed a sampling point every 10 frames in the stego audio and extracted the secret message. The integrity results are summarized in Table 2.

As shown in Table 2, the integrity of the secret message extracted by the proposed algorithm in the face of frame misalignment and frame loss is better than that of the reference algorithm because we use the beat as the anchor point to select the cover elements.

4.3. Imperceptibility

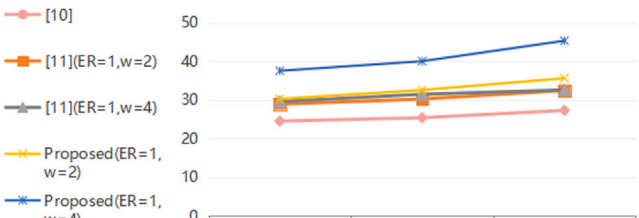
Imperceptibility refers to the auditory impact of embedding secret messages into the AAC audio. We test the algorithm from both the subjective and objective perspectives. Subjectively, we compare the waveforms before and after embedding the messages. Objectively, we calculate the signal-to-noise ratio (SNR) and objective difference grade (ODG) of the stego audio. The waveforms of the stego audio in Fig. 6(a) are compared with those of the cover audio in Fig. 6(b).

Subjectively, the difference between the waveforms is minimal. To measure the effectively imperceptibility, an objective calculation method is used to calculate the effect of the stego audio quantitatively. First, the ODG value of stego audio is calculated by the perceptual evaluation of the audio quality (PEAQ) algorithm [28]. The ODG value represents the similarity between the stego and cover audios, and its value is between −4 and 0, where 0 indicates no distortion. The results are summarized in Table 3.

From Table 3, the imperceptibility of the proposed algorithm is better compared to that of the algorithms in [12] and [14], and the ODG value is closer to 0 in the case of a high bit rate. However, using the PEAQ algorithm to calculate ODG is limited to the sound channel, as this algorithm can only measure two-channel stego audio, which must be in WAV format. To directly obtain the imperceptibility of stego AAC audio, we calculate the SNR of the stego audio to measure the imperceptibility of the algorithm. The

Table 3
ODG results of stego audio.

Bit rate	Proposed ($ER = 1, w = 2$)	Proposed ($ER = 1, w = 2$)	[14] $ER = 1, w = 2$	[14] $ER = 1, w = 4$	[12]
64 kb/s	−1.224	−1.103	−1.425	−1.205	−1.685
128 kb/s	−0.492	−0.485	−0.514	−0.510	−0.528
192 kb/s	−0.013	−0.010	−0.019	−0.012	−0.018



	64kb/s	128kb/s	192kb/s
[10]	24.562	25.43	27.33
[11](ER=1,w=2)	28.956	30.233	32.452
[11](ER=1,w=4)	29.6	31.472	32.568
Proposed(ER=1,w=2)	30.265	32.568	35.622
Proposed(ER=1,w=4)	37.542	40.024	45.322

Fig. 7. SNR of stego audio.

larger the SNR is, the better the imperceptibility of the stego audio is. According to the requirements of IFPI, the human ear cannot perceive audio differences when the SNR is greater than 20. The SNR was calculated as in Eq. (7):

$$SNR(X, X') = 10log_{10}\frac{\sum_{i=1}^{L} X_i^2}{\sum_{i=1}^{L}(X_i' - X_i)^2}, \tag{7}$$

where X_i and X_i' represent the ith sample values of the original audio signal and stego audio, respectively, and L represents the length of the audio signal. The results are shown in Fig. 7.

As observed from Fig. 7, the proposed algorithm provides significantly better results compared to those from [12,14] at the three bit rates. Additionally, the proposed algorithm has good imperceptibility, where the SNR can be 45.322 at 192 kb/s. This is mainly because when we use the STCs framework for embedding, we obtain the optimal solution with minimal distortion. Moreover, we use the audio beat to choose the cover elements such that the embedding position is random, and the double effect makes the audio imperceptible.

4.4. Capacity

Embedding capacity refers to the number of bits of secret messages that can be embedded in AAC audio per second. Because the embedding of the STCs framework depends on the size of the parity check matrix, and according to Fig. 3, the number of MDCT coefficients in the small value area is significantly large, the ER is determined according to Eq. (8).

$$ER = \frac{m}{n}, \tag{8}$$

where n represents the QMDCT coefficient in the small value area and m represents the number of cover elements chosen from n. The width of the h' matrix in the STCs framework is w. According to the embedding process in Section 2.2, when we input the cover element of w bits into the STCs framework, one bit of the message is embedded. Therefore, the algorithm can achieve different embedding capacities according to the different values of w. To evaluate the embedding capacity of the algorithm, we compare it with those in [12,14], and the corresponding average embedding capacities for AAC audio at different bit rates are listed in Table 4.

From Table 4, the embedding capacity of the proposed algorithm is significantly better than that of the algorithm in [12], but slightly worse than that of the algorithm in [14]. This is mainly because the proposed algorithm chooses the QMDCT coefficients in the small value area as the cover element and it uses the beat as the anchor point to choose the cover elements in more detail. Therefore, the embedding capacity is reduced, however, in real life, this embedding capacity is sufficient, and the gap between the proposed algorithm and the algorithm when ER = 1 and w = 2 is extremely small. Thus, the proposed algorithm chooses to moderately sacrifice embedding capacity to ensure better imperceptibility and steganalysis. Compared to that in [12], the embedding capacity of the proposed algorithm is better, which indicates that this capacity is sufficient in practical applications.

Table 4
Capacity results.

	Capacity(bit/s)		
	64 kb/s	128 kb/s	192 kb/s
[12]	2560.47	1471.00	692.73
[14] $ER = 0.5, w = 2$	2171.63	5576.64	9554.14
[14] $ER = 0.5, w = 4$	1060.75	2766.84	4755.59
[14] $ER = 1, w = 2$	4386.25	11196.16	19150.84
[14] $ER = 1, w = 4$	2171.73	5576.64	9554.10
Proposed($ER = 0.5, w = 2$)	2231.06	2001.26	1720.55
Proposed($ER = 0.5, w = 4$)	1045.25	1000.96	720.56
Proposed($ER = 1, w = 2$)	4310.23	4012.37	3750.66
Proposed($ER = 1, w = 4$)	2321.42	2000.32	1620.44

Table 5
Steganalysis results.

Bit rate	Proposed ($ER = 1, w = 2$)	Proposed ($ER = 1, w = 2$)	[14] $ER = 1, w = 2$	[14] $ER = 1, w = 4$	[12]
64 kb/s	0.0114	0.0104	0.0386	0.0211	0.09833
128 kb/s	0.0112	0.0102	0.0320	0.0160	0.04213
192 kb/s	0.0103	0.0093	0.0200	0.0120	0.01942

4.5. Steganalysis

In actual applications, stego audio must be tested using a steganalysis algorithm. Different steganalysis methods exist for different embedding positions, and the proposed algorithm modifies the QMDCT coefficients to embed the secret information. According to the embedding rules, the modified value of the QMDCT coefficients is 1 or 0, therefore, the most advanced steganalysis algorithm is used to analyze the statistical characteristics of the QMDCT coefficients. First, the histograms of the QMDCT coefficients of the cover and stego audio are drawn, and then the Manhattan distance of the two histograms is calculated. The calculation equation for the Manhattan distance is as follows:

$$d_{manh} = \sqrt{\sum_{i}^{d} |C_i - O_i|}, \tag{9}$$

where d is the number of histogram bars, and C and O represent the probability densities of the QMDCT coefficient histogram of the cover and stego audios, respectively. The smaller the d_{manh} is, the smaller the difference between the QMDCT coefficients of the cover and stego audios, that is, the proposed algorithm resists steganalysis better. The results of steganalysis are summarized in Table 5.

From Table 5, the proposed algorithm can resist steganalysis well. The value of the Manhattan distance is infinitely close to 0 at 192 kb/s, indicating that the algorithm has minimal changes to QMDCT and it can resist the current common statistical analysis. In the proposed algorithm, the STCs framework theoretically minimizes the cost of embedding, and the beats are used to choose the cover elements, thus, the algorithm performs well in steganalysis.

5. Conclusion

An adaptive AAC steganography algorithm based on an audio beat is proposed in this paper. First, we innovatively adopt a new distortion function to ensure the operation of the minimum distortion model and consider the effect of AAC coding on data distortion. In addition, we use a deep learning algorithm to obtain the audio beat points, and utilize the beat points as the anchor points to determine the cover elements. Therefore, the algorithm can still effectively extract the secret information in the case of frame misalignment. Finally, the QMDCT coefficients in the small value area are determined as the cover element. They cover the small energy of the audio, and slight changes do not cause auditory loss to the audio. The experimental results indicate that the two innovative improvements can significantly improve the overall performance of the algorithm.

Although the algorithm effectively improve imperceptibility and anti-steganography, these aspects can be further improved. In this paper, when considering the data distortion function, the degree of influence of all data that may affect the cover elements is set to the same size, however, in actual coding, the influence of the coefficients at different positions on the cover element should be different. Moreover, the QMDCT coefficients in the small value area are determined for embedding, and the embedding capacity problem may occur in the case of a high code rate. These aspects need to be further studied, which would be addressed in future research.

CRediT authorship contribution statement

Xue Zhang: Methodology, Software, Data curation, Writing – original draft. **Chen Li:** Conceptualization, Methodology, Writing – review & editing, Formal analysis. **Lihua Tian:** Resources, Supervision, Formal analysis.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.compeleceng.2023.108580.

Data availability

Data will be made available on request.

Acknowledgments

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, China.

References

[1] Lu H, Zhang Y, Li Y, Jiang C, Abbas H. User-oriented virtual mobile network resource management for vehicle communications. IEEE Trans Intell Transp Syst 2021;22(6):3521–32. http://dx.doi.org/10.1109/TITS.2020.2991766.
[2] Zhu J, Wang R, Yan D. The sign bits of huffman codeword-based steganography for AAC audio. In: 2010 International conference on multimedia technology. 2010, p. 1–4. http://dx.doi.org/10.1109/ICMULT.2010.5629745.
[3] Shelke R, Nemade M. Audio encryption algorithm using modified elliptical curve cryptography and arnold transform for audio watermarking. In: 2018 3rd International conference for convergence in technology. 2018, p. 1–4. http://dx.doi.org/10.1109/I2CT.2018.8529329.
[4] Zhang R, Dong H, Yang Z, Ying W, Liu J. A CNN based visual audio steganography model. In: Sun X, Zhang X, Xia Z, Bertino E, editors. Artificial intelligence and security. Cham: Springer International Publishing; 2022, p. 431–42.
[5] Mahmoud MM, Elshoush HT. Enhancing LSB using binary message size encoding for high capacity, transparent and secure audio steganography–An innovative approach. IEEE Access 2022;10:29954–71. http://dx.doi.org/10.1109/ACCESS.2022.3155146.
[6] Luo W, Zhang Y, Li H. Adaptive audio steganography based on advanced audio coding and syndrome-trellis coding. In: International workshop on digital watermarking. Springer; 2017, p. 177–86.
[7] Lu H, Tang Y, Sun Y. DRRS-BC: Decentralized routing registration system based on blockchain. IEEE/CAA J Autom Sin 2021;8(12):1868–76. http://dx.doi.org/10.1109/JAS.2021.1004204.
[8] Wang XQ. Implementation of voice encryption technology based on AAC audio coding and chaotic encryption. Modern Electron Tech 2012.
[9] Nejad MY, Mosleh M, Heikalabad SR. An LSB-based quantum audio watermarking using MSB as arbiter. Internat J Theoret Phys 2019;58(11):3828–51.
[10] Zong T, Xiang Y, Natgunanathan I, Gao L, Hua G, Zhou W. Non-linear-echo based anti-collusion mechanism for audio signals. IEEE/ACM Trans Audio Speech Lang Process 2021;29:969–84.
[11] Ren Y, Xiong Q, Wang L. A steganalysis scheme for AAC audio based on MDCT difference between intra and inter frame. In: International workshop on digital watermarking. Springer; 2017, p. 217–31.
[12] Li C, Zhang X, Luo T, Tian L. Audio steganography algorithm based on genetic algorithm for mdct coefficient adjustment for AAC. In: 2020 IEEE international symposium on multimedia. IEEE; 2020, p. 111–2.
[13] Ren Y, Cai S, Wang L. Secure AAC steganography scheme based on multi-view statistical distortion (SofMvD). J Inform Secur Appl 2021;59:102863. http://dx.doi.org/10.1016/j.jisa.2021.102863, URL https://www.sciencedirect.com/science/article/pii/S2214212621000946.
[14] Zhang Z, Yi X, Zhao X. An AAC steganography scheme for adaptive embedding with distortion minimization model. Multimedia Tools Appl 2020;79(37):27777–90.
[15] Wei Y, Guo L, Wang Y. Controlling bitrate steganography on AAC audio. In: 2010 3rd International congress on image and signal processing, vol. 9. IEEE; 2010, p. 4373–5.
[16] Zhu J, Wang R, Yan D. The sign bits of huffman codeword-based steganography for aac audio. In: 2010 International conference on multimedia technology. IEEE; 2010, p. 1–4.
[17] Lu H, Zhang M, Xu X, Li Y, Shen HT. Deep fuzzy hashing network for efficient image retrieval. Trans Fuz Sys 2021;29(1):166–76. http://dx.doi.org/10.1109/TFUZZ.2020.2984991.
[18] Dixon S. Mirex 2006 audio beat tracking evaluation: Beatroot. In: MIREX 2006. Citeseer; 2006, p. 27.
[19] Eck D, Lamere P, Bertin-Mahieux T, Green S. Automatic generation of social tags for music recommendation. Adv Neural Inf Process Syst 2007;20:385–92.
[20] Peeters G, Papadopoulos H. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. IEEE Trans Audio Speech Lang Process 2010;19(6):1754–69.
[21] Heydari M, Duan Z. Don't look back: An online beat tracking method using RNN and enhanced particle filtering. In: ICASSP 2021-2021 IEEE International conference on acoustics, speech and signal processing. IEEE; 2021, p. 236–40.
[22] Böck S, Schedl M. Enhanced beat tracking with context-aware neural networks. In: Proc. int. conf. digital audio effects. 2011, p. 135–9.
[23] Filler T, Judas J, Fridrich J. Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Trans Inf Forensics Secur 2011;6(3):920–35.
[24] Sedighi V, Cogranne R, Fridrich J. Content-adaptive steganography by minimizing statistical detectability. IEEE Trans Inf Forensics Secur 2015;11(2):221–34.
[25] Ying K, Wang R, Lin Y, Yan D. Adaptive audio steganography based on improved syndrome-trellis codes. IEEE Access 2021;9:11705–15. http://dx.doi.org/10.1109/ACCESS.2021.3050004.
[26] Shin D, Hong Y, Kim J, Choi J. Audio blind watermarking robust against HE-AAC. In: Proceedings of the 8th international conference on signal processing systems. 2016, p. 114–8.
[27] Wang Y, Yang K, Yang Y, Zhang J, Zhao X. Audio steganalysis dataset. 2019, http://dx.doi.org/10.21227/rab0-vf56.
[28] Thiede T, Treurniet WC, Bitto R, Schmidmer C, Sporer T, Beerends JG, Colomes C. PEAQ-The ITU standard for objective measurement of perceived audio quality. J Audio Eng Soc 2000;48(1/2):3–29.