EURASIP Journal on
Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Context-based adaptive arithmetic coding in time and frequency domain for the lossless compression of audio coding parameters at variable rate

Jing Wang*, Xuan Ji, Shenghui Zhao, Xiang Xie and Jingming Kuang

Abstract

This paper presents a novel lossless compression technique of the context-based adaptive arithmetic coding which can be used to further compress the quantized parameters in audio codec. The key feature of the new technique is the combination of the context model in time domain and frequency domain which is called time-frequency context model. It is used for the lossless compression of audio coding parameters such as the quantized modified discrete cosine transform (MDCT) coefficients and the frequency band gains in ITU-T G.719 audio codec. With the proposed adaptive arithmetic coding, a high degree of adaptation and redundancy reduction can be achieved. In addition, an efficient variable rate algorithm is employed, which is designed based on both the baseline entropy coding method of G.719 and the proposed adaptive arithmetic coding technique. Experiments show that the proposed technique is of higher efficiency compared with the conventional Huffman coding and the common adaptive arithmetic coding when used in the lossless compression of audio coding parameters. For a set of audio samples used in the G.719 application, the proposed technique achieves an average bit rate saving of 7.2% at low bit rate coding mode while producing audio quality equal to that of the original G.719.

Keywords: Adaptive arithmetic coding, Time-frequency context, Lossless compression, Variable rate, MDCT

1. Introduction

Natural digital audio signals require large bandwidth for transmission and enormous amounts of storage space. Developments in entropy coding, i.e., Huffman coding [1,2] and arithmetic coding [3,4], have made it practical to reduce these requirements without information loss. They employ non-stationary statistical behavior which exploits redundant information in the source signal. Compared with lossless compression methods, vector quantization methods and lossy compression methods are adopted in audio coding system to remove irrelevancy inaudible to humans and to improve the coding efficiency. Many audio codecs only use lossy compression methods to quantize and encode the audio parameters. In fact, when further combined with lossless entropy coding for the quantization and encoding procedure, audio codec can achieve better performance on the coding efficiency compared with using the lossy compression alone.

With the development of modern multimedia communication, high-quality full-band speech and audio coding becomes significant and is needed more at low bit rate. Besides the lossy compression through parametric and transform coding, many audio codecs introduce lossless coding algorithm to further compress the coding bits, such as Moving Picture Experts Group-4 advanced audio coding (MPEG-4 AAC) [5], MPEG unified speech and audio coding (USAC) [6], and ITU-T G.719 [7]. ITU-T G.719 is a low-complexity full-band (20 Hz to 20 kHz) audio codec for high-quality speech and audio, which operates from 32 to 128 kbps [7]. As with most of the transform audio coding, G.719 uses modified discrete cosine transform (MDCT) to realize the time-frequency transform and to avoid artifacts stemming from the block boundaries. In the MDCT domain [8], statistical and

* Correspondence: wangjing@bit.edu.cn
Research Institute of Communication Technology (RICT), School of Information and Electronic Engineering, Beijing Institute of Technology, Beijing 100081, China

subjective redundancies of the signals can be better understood, exploited, and removed in most cases. After the lossy compression with vector quantization, removing irrelevancy inaudible to humans, the further compression performance is largely determined by the entropy coding efficiency of the quantized MDCT coefficients. In G.719, Huffman coding is applied, and the coding procedure has to be driven by an estimated probability distribution of the quantized MDCT coefficients along with the norms (frequency band gains).

Although Huffman coding removes some of the quantized MDCT coefficients' redundancy, it suffers from several shortcomings which limit further coding gains. For instance, in Huffman code, the distribution of MDCT coefficients is pre-defined from training statistics, and the adaptation mechanism is not flexible enough to combat the possible statistics mismatch, such as the techniques of switching between different codebooks and multi-dimensional codebooks which are exploited in AAC. Furthermore, if the symbols are not grouped into blocks, the symbols whose probabilities greater than 0.5 cannot be efficiently coded due to the intrinsic limit of 1 bit per symbol of Huffman code. Hence, the entropy coding schemes based on the adaptive arithmetic coding [9] are involved in the audio codec like MPEG USAC. The adaptive model measures the statistics of source symbols and is updated continuously with the encoding and decoding processes. In addition, the context from the point of view of the neighboring symbols is taken into account in order to further improve the coding efficiency.

For the context, it is firstly introduced in image and video coding. Here, context-based adaptive binary arithmetic coding (CABAC) in H.264/AVC [10] is taken as an example. CABAC is one of the two entropy coding methods of the new ITU-T/ISO/IEC standard for video coding, i.e., H.264/AVC, and plays a very important role in the efficiency improvement of the video coding. Through combining an adaptive binary arithmetic coding technique with context modeling of the neighboring symbols in binary bit stream and macro block, a high degree of adaptation and redundancy reduction is achieved. The encoding process of CABAC consists of three elementary steps: binarization, context model selecting, and adaptive binary arithmetic encoding. The last step consists of probability estimation and binary arithmetic encoder.

In the second step of CABAC [10], a context model is chosen, and a model probability distribution is assigned to the given symbols. In the subsequent coding stage, the binary arithmetic coding engine generates a sequence of bits that represent the symbols. The model determines the coding efficiency in the first place, so it is of paramount importance to design an adequate model that explores the statistical dependencies to a large degree. At the same time, this model needs to be continuously updated during encoding. Suppose one pre-defined set T of the past symbols, a so-called context template, and one related set $C = \{0,\ldots,C\text{-}1\}$ of the contexts are given, where the contexts are specified by a modeling function $F{:}T \rightarrow C$ operating on the template T. For each symbol x to be coded, a conditional probability $p(x|F(z))$ is estimated by switching between different probability models according to the already coded neighboring symbols $z \in T$. Generally speaking, the context model makes use of the information related to the encoded symbols and describes the mapping between a sequence of symbols and the assignment of the symbols' probability distribution.

Lately, arithmetic coding schemes based on bit-plane context are also involved in the field of audio coding such as USAC, like the applications in video coding. The spectral noiseless coding scheme is based on an arithmetic coding in conjunction with a dynamically adaptive context. The noiseless coding is fed by the quantized spectral values and uses context-dependent cumulative frequency tables derived from the two previously decoded neighboring two-tuple quantized spectral coefficients. The coding separately considers the sign, the two most significant bits (MSBs) and the remaining least significant bits. The context adaptation is applied only to the two MSBs of the unsigned spectral values. The sign and the least significant bits are assumed to be uniformly distributed.

By now, entropy coding schemes based on arithmetic coding are quite frequently involved in the field of none block-based video coding. The CABAC design is based on the key elements of binarization, context modeling, and binary arithmetic coding. Binarization enables efficient binary arithmetic coding via a unique mapping of non-binary syntax elements to a sequence of bits, which are called bins. Now, the arithmetic coding as a lossless data compression scheme also plays an essential role in the chain of processing of audio signal coding. The correlation in bit plane of the quantized MDCT coefficients is employed in the USAC [11]. However, the concept of context model for the adaptive arithmetic coding has been neither deeply investigated nor widely used in audio coding especially for the efficient compression by setting up context model from the point of view of the quantized audio parameters. When using the arithmetic coding to compress the coding parameters directly, the probability estimation based on the bit-plane context model may not be suitable. In this situation, the correlation of audio coding parameters leading to lower information entropy could be considered both in time and frequency domain which can be deeply investigated in theory and carefully designed in practice. Thus, a novel time-frequency plane context model will be given in this paper, and the adaptive arithmetic coding will be used directly for the audio coding parameters. Furthermore, variable coding scheme is introduced to advance the efficiency.

In our work on arithmetic coding, the entropy coding method of an adaptive arithmetic coding technique with a time-frequency plane context model (both time and frequency domain are taken into account) was developed, which has led to the improvement of coding the quantized MDCT coefficients and the frequency band gains. The adaptive arithmetic coding will be applied to further compress the coding parameters in audio codec frame by frame and the probability estimation of which will make use of the inter-frame (time domain) correlation and the intra-frame (frequency domain) correlation of the coding parameters. In fact, most of alternative approaches to audio coding are on the basis of MDCT. One of its main distinguishing features is related to the time-frequency plane: Given a source of the quantized transform coefficients for instance, it was found to be useful to utilize the correlation in the time domain and frequency domain to increase the probability of the encoding symbol for arithmetic coding. The experiment on G.719 is carried out as an application of the proposed technique, in which the compatibility with the G.719 baseline is required. The good compression performance is achieved. Adopting this method, the allocated bits for coding the quantized parameters vary in consecutive analysis frames, while the quality of decoded audio remains constant. Therefore, the average bit rate is lower than that of the fixed bit rate codec while sustaining the same audio quality. Hence, a variable rate operation is introduced into the novel context-based adaptive arithmetic coding algorithm, which achieves better performance in terms of the coding efficiency.

This paper is organized as follows. Section 2 outlines the novel adaptive arithmetic coding of the parameters produced in the audio encoding. Section 3 describes in detail the novel techniques and the underlying ideas of our entropy coding modules. Section 4 presents the experimental results and the performance comparison. Section 5 concludes this paper with a summary.

2. Modules of the novel adaptive entropy coding

2.1. Preliminary principle

The information entropy of a discrete memoryless source X which has different symbols (x_0, ……, x_{I-1}) is given by [12,13]

$$H(X) = -\sum_{i=0}^{I-1} p(x_i)\log_2(p(x_i)), \tag{1}$$

where $p(x_i)$ is the probability of the symbol x_i.

The entropy establishes the lower bound of the average bit rate achieved by source coding. However, when the source is correlated, this bound can be further lowered by taking into account a higher order of the entropy like the conditional entropy

$$H(|S) = -\sum_{j=0}^{J-1} p(s_j) \sum_{i=0}^{I-1} p(x_i|s_j)\log_2(p(x_i|s_j)), \tag{2}$$

where s_j, a so-called context, is a specific state of the source and J represents the total number of the considered states. For the application of the so-called context, the distribution of the symbols (x_0, ……, x_{I-1}) is more concentrated in the vicinity of the encoding symbol, which means the probability of the encoded symbol can be increased through establishing the context model. Consequently, a suitable context design considering the correlation of the source means the lower entropy. In the applications of audio coding, because of the similarity of the sequential frames as well as the adjacent frequency bands, some audio parameters like frequency band gains and frequency spectral values have the correlation in time, and frequency domain and the context model with the neighboring parameters can be designed to make the entropy of the coding source lower, thus the compression efficiency can be higher. In Sections 2.3 and 2.4, the proposed context model and the way to utilize it will be mentioned in theory. The practical behavior and design in the case of G.719 codec will be investigated in Section 3.

2.2 Integer arithmetic coding

The performance of arithmetic coding is optimal without the need for blocking of input data. It encourages a clear separation between the probability distribution model and the encoding of information. For example, the model may assign a predetermined probability to each symbol. These probabilities can be determined by counting frequencies of representative samples to be transmitted. Such a fixed model is communicated in advance for both encoder and decoder. Alternatively, the probabilities that an adaptive model assigns may change as each symbol is transmitted. The encoder's model changes as each symbol is transmitted and the decoder's model changes as each symbol is received. If the context is involved, the adaptive model is based on the context.

In the arithmetic coding, a message is represented by an interval of real numbers between 0 and 1. As the message becomes longer, the interval becomes smaller, and it is necessary for the decoder to know the final interval at the end of the arithmetic coding. However, the integer arithmetic coding [14-16] can be employed without knowing the final interval, i.e., the decoding algorithm can be carried out even if the encoding procedure has not been completed. Meanwhile, the interval required to represent the message can grow in the process of encoding. Integer arithmetic coding [14] is done by subdividing the current interval initialized to [0, N-1] according to the symbol probabilities, where N is the upper limit of the 32-bit integer in the computer. The probabilities in the model are represented as integer frequency counts [16], and the cumulative counts are stored in the array $c()$. When a symbol comes each time, we take its subinterval as current interval. To put it simply, the subinterval to be encoded is represented in the form [l, u], where l is called the base or starting point of the subinterval and u is the ending point of the subinterval. The subintervals in the arithmetic coding process are defined by the equations as follows:

$$\Phi_0 = [l_0, u_0] = [0, N{-}1], \tag{3}$$

$$\Phi_i = [l_i, u_i] = \left[l_{i-1} + \frac{c(x_{i-1})}{c(x_{I-1})}(u_{i-1}{-}l_{i-1}+1), l_{i-1} + \frac{c(x_i)}{c(x_{I-1})}(u_{i-1}{-}l_{i-1}+1){-}1\right]. \quad i = 0, 1, \ldots\ldots, I{-}1 \tag{4}$$

The properties of the intervals guarantee that $0 \le l_i \le l_{i+1} < N$, and $0 \le u_i \le u_{i+1} < N$. The expression $\frac{c(x_i)-c(x_{i-1})}{c(x_{I-1})}$ is equivalent to $p(x_i)$ in Equation 1. To have incremental output, i.e., coded word, during the encoding process and to resolve the need for high-precision computations, the algorithm is performed through three mappings as follows. ‘Scale’ is defined as an intermediate variable in the calculation process to count the number of the three mappings, which represents the bit following the previous output bit in steps I and II.

- I: If the subinterval [l, u] lies entirely in the lower half part of [0, N − 1], i.e., [0, N/2 − 1], then the coder emits a bit 0 and scale outputs a bit 1 until it is successively reduced to 0, and linearly expands [l, u] to [2l, 2u + 1]. Scale is reset to 0.
- II: If the subinterval [l, u] lies entirely in the upper half part of [0, N − 1], i.e., [N/2, N − 1], then the coder emits a bit 1 and scale outputs a bit 0 until it is successively reduced to 0, and linearly expands [l, u] to [2l − N, 2u − N + 1]. Scale is reset to 0.
- III: If the subinterval [l, u] lies entirely in the interval [N/4, 3N/4 − 1], then the coder linearly expands [l, u] to [2l − N/2, 2u − N/2 + 1] and increases the value of scale by 1.

The three mapping steps will be ended until the interval [l, u] meets with none of the above looping conditions. As the subinterval shortens, the number of loops increases which lead to more bits output. Thus, the larger the subinterval is, the smaller bits the coder output. Since the context model can be established to increase the probability of the encoded symbol, the subinterval representing the probability will correspond to be enlarged.

2.3. Time-frequency context model

Generally, in the current applications, the context consists of neighbors of the current symbol to be encoded. In the application of CABAC, the context models the neighboring symbols of the binary bit, and in the application of USAC, the adaptive arithmetic coding is established based on the bit plane. In this paper, it deals with the correlation of neighboring parameters in the transform audio coding and some basic rules are designed to help with selecting the proper context model for the adaptive arithmetic coding. The time-frequency context associated with the current coded element is shown in Figure 1 which is different from the bit-plane context used in CABAC and USAC. In the proposed model, the time-frequency context-based arithmetic coding only makes use of the neighboring parameters in the past frames when considering the time domain, so there is no extra algorithmic delay when the arithmetic coder accesses to the time-frequency plane, shown in Figure 1.

A family of contexts is defined by means of the function $T(m)$. The parameter m represents the number of symbols lying in the vicinity of the present coded symbol with $0 \le m \le 2$. For each symbol C to be coded, the conditional probability $p(C|T(m))$ is estimated by switching between different probability models according to the already coded neighboring symbols. In Figure 1, $T(0)$ represents no context, $T(1)$ = A or B, and $T(2)$ = A or B. A represents the context in the frequency domain, while B represents the context in the time domain, and they correspond with the quantized parameters in the transform audio codec. Their

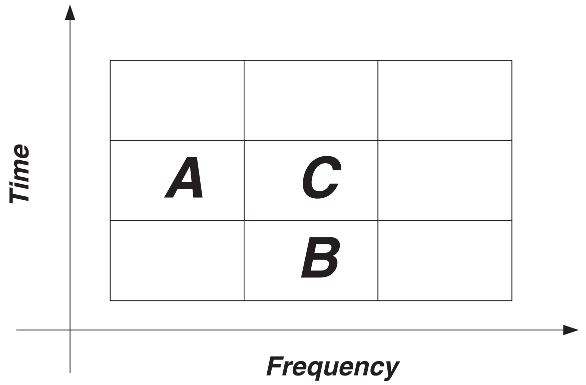

Figure 1 A context template consisting of two neighboring elements A and B. Which are on the left and on the bottom of the current element C, respectively. The x-axis represents frequency, and the y-axis represents time.

conditional probabilities are estimated by different methods which will be introduced in the following sections.

2.3.1. Context model in the frequency domain

When the neighboring elements satisfy the following equations

$$x = \{x_0, x_1, ..., x_{I-1}\}, \tag{5}$$

$$s = \{s_0, s_1, ..., s_{J-1}\}, \tag{6}$$

$$\frac{c(x_i|s_j)-c(x_{i-1}|s_j)}{c(x_{I-1}|s_j)} > \frac{c(x_i)-c(x_{i-1})}{c(x_{I-1})}, \tag{7}$$

where x represents the symbols and s represents the context. Then, the context dependence in the frequency domain is given a primary consideration. This guarantees a larger subinterval, which we explain as follows:

$$l1 = l + \frac{(u-l+1) \times c(x_{i-1}|s_j)}{c(x_{I-1}|s_j)} \tag{8}$$

$$u1 = l + \frac{(u-l+1) \times c(x_i|s_j)}{c(x_{I-1}|s_j)} - 1, \tag{9}$$

$$l2 = l + \frac{(u-l+1) \times c(x_{i-1})}{c(x_{I-1})}, \tag{10}$$

$$u2 = l + \frac{(u-l+1) \times c(x_i)}{c(x_{I-1})} - 1. \tag{11}$$

In the case of Equation 7, the result $u1 - l1 > u2 - l2$ can be obtained. Since the subinterval calculated by the conditional probability is larger, smaller bits are obtained. The conditional entropy can be smaller than the entropy.

$$\begin{aligned} H(X|S) = &-\sum_{j=0}^{J-1} \frac{c(s_j)-c(s_{j-1})}{c(s_{J-1})} \sum_{i=0}^{I-1} \frac{c(x_i|s_j)-c(x_{i-1}|s_j)}{c(x_{I-1}|s_j)} \log_2 \\ &(\frac{c(x_i|s_j)-c(x_{i-1}|s_j)}{c(x_{I-1}|s_j)}) \\ &< H(X) = -\sum_{i=0}^{I-1} \frac{c(x_i)-c(x_{i-1})}{c(x_{I-1})} \log_2(\frac{c(x_i)-c(x_{i-1})}{c(x_{I-1})}). \end{aligned} \tag{12}$$

The length of the context-based sequence is defined as the order of the context model. A key issue in context modeling for the input symbol sequence is to balance the usage of the model order and the model cost. Higher order means higher cost of the computation. To solve this problem, one order context model [17] can be chosen in the frequency domain regarding its good compression and low complexity in the audio coding application.

2.3.2. Context model in the time domain

When the neighboring elements are correlated and the current symbol C distributes around the encoded symbol B, i.e., $C \in (B - \delta, B + \delta)$, where δ represents the rescaling parameter, the model probability distribution is reassigned to the current symbol C.

For the m-ary (m is the number of symbols) adaptive arithmetic coding, the encoded symbol B is taken as the center; 2δ symbols, which are located in the vicinity of B, would be chosen to add a large number λ on the basis of the original frequency, leading to rearrange the distribution of the model. λ is the cumulative counts of all symbols which can change the subinterval adaptively.

That is,

$$f(x_i) = c(x_i) - c(x_{i-1}), \tag{13}$$

$$\lambda = \sum_{i=0}^{I-1} f(x_i), \tag{14}$$

$$f'(x_i) = \begin{cases} f(x_i) + \lambda; & i = B - \delta + 1, ..., B, ..., B + \delta \\ f(x_i); & other \end{cases}, \tag{15}$$

where $f(x_i)$ is the original frequency counts of the symbol and $f'(x_i)$ represents the final frequency counts distribution assigned to drive the arithmetic coder. The subinterval is changed to $(l'1, u'1)$

$$l'1 = l + \frac{(u-l+1) \times c'(x_{i-1})}{c'(x_{I-1})}, \tag{16}$$

$$u'1 = l + \frac{(u-l+1) \times c'(x_i)}{c'(x_{I-1})} - 1, \tag{17}$$

$$l2 = l + \frac{(u-l+1) \times c(x_{i-1})}{c(x_{I-1})}, \tag{18}$$

$$u2 = l + \frac{(u-l+1) \times c(x_i)}{c(x_{I-1})} - 1. \tag{19}$$

As $f'(x_i)$ increases for $i = B - \delta + 1, ..., B, ..., B + \delta$, the inequality $\frac{c'(x_i)-c'(x_{i-1})}{c'(x_{I-1})} > \frac{c(x_i)-c(x_{i-1})}{c(x_{I-1})}$ can be obtained. The subinterval $u'1 - l'1$ is then larger than $u2 - l2$ under the above condition. Consequently, the higher the encoding symbol's frequency counts value is, the better the designed coding scheme performs with the larger subinterval of the encoding symbol.

As to the context model in time domain, we only consider one state context which models the past symbol B close to the current symbol C because the state before B has a weaker correlation with C while more states mean higher complexity.

3. Scheme of the novel context adaptive arithmetic coding in G.719

3.1 State-of-the-art techniques of G.719

ITU-T G.719 codec [7] makes use of the transform coding technique for low-complexity full-band conversational speech and audio, operating from 32 up to 128 kbps. The input signal sampled at 48 kHz is firstly processed through a transient detector based on the energy ratio between the short-term energy and the long-term energy. An adaptive window switching technique is used depending on the detection of transient and stationary signal. Then, time domain aliasing and MDCT techniques are designed to process the different kind of input signal. The transformed spectral coefficients are grouped into subbands of unequal lengths. The gain of each band (i.e., norm) is estimated, and the resulting spectral envelope consisting of the norms of all bands is quantized and encoded. The quantized norms are further adjusted based on adaptive spectral weighting and used as the input for bit allocation. The spectral coefficients are normalized by the quantized norms, and the normalized MDCT coefficients are then lattice vector quantized and encoded based on the allocated bits for each frequency band. In the process of bit allocation, Huffman coding is applied to encode the indices of both the encoded spectral coefficients and the encoded norms. The saved bits by Huffman coding are used for the following bit allocation and the noise adjustment in order to generate better audio quality. Finally, the fixed bit stream is obtained and transmitted to the decoder.

3.2 The novel structure of G.719

In this section, the novel context-based adaptive arithmetic coding is introduced to improve the coding scheme in G.719, and the probability statistic of the entropy coding is established for the transient and the stationary audio separately. The key elements will be discussed in the next section.

Figure 2 shows the basic structure of the proposed method. The input signals (sampled at 48 kHz) are firstly processed through a transient detector [7] to be classified into transient and stationary signals, which are assigned with different statistical models for the adaptive arithmetic coding. After the modified discrete cosine transform, the obtained spectral coefficients are firstly grouped into subbands of unequal lengths. Then, the norm of each band, i.e., the frequency band gain, is estimated and the resulting spectral envelope, consisting of the norms of all bands, is quantized and encoded. Regarding the good correlation of the quantized norms of neighbor bands, we apply the time-frequency context-based adaptive arithmetic coding. The time-frequency context aims to remove the redundancies in the frequency domain and in the time domain.

When the coding procedure of the quantized norms is over, the coefficients are normalized by the quantized norms, and then, the normalized spectral coefficients are lattice vector quantized according to the bit allocation which leads to different dynamic range in subbands. For the so-called bit allocation, the maximum number of bits assigned to each normalized transform coefficient is set to

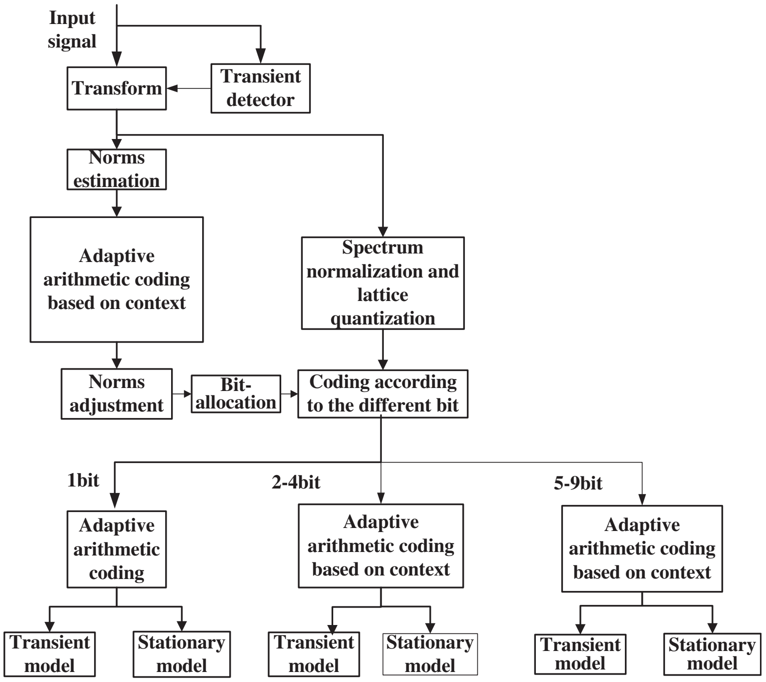
Input signal
Transform
Transient detector
Norms estimation
Adaptive arithmetic coding based on context
Spectrum normalization and lattice quantization
Norms adjustment
Bit-allocation
Coding according to the different bit
1bit
2-4bit
5-9bit
Adaptive arithmetic coding
Adaptive arithmetic coding based on context
Adaptive arithmetic coding based on context
Transient model
Stationary model
Transient model
Stationary model
Transient model
Stationary model

Figure 2 The basic structure of the proposed method.

$R_{\max}$ = 9 in G.719 by default. Thus, nine statistical models for the adaptive arithmetic coding to be updated are employed, and all bands will be rearranged in order from low band to high band for the arithmetic coding so that the quantized coefficients in the subbands with the same allocated bits are encoded continuously. Considering that the 1-bit subband, the 2- to 4-bit subband, and the 5- to 9-bit subband have different correlations in the time domain and in the frequency domain, we use different context models when the bit allocation is different. The subbands of 5 to 9 bits are designed to exploit the correlation in the time domain for compression, while the subbands of 2 to 4 bits make good use of the correlation in frequency domain. Finally, the subband of 1 bit uses the normal adaptive arithmetic coding.

3.3 Time-frequency context model in G.719

Through a large number of experiments, we have found that the quantized norms and the quantized MDCT coefficients with 2 to 4 bits have the context statistical characteristic in the frequency domain, while the quantized norms and the quantized MDCT coefficients with 5 to 9 bits have the characteristic in the time domain, as is discussed in Section 2.3.

In the frequency domain, if the spectral parameters have the correlation, the conditional probability of the current encoding symbol will be larger than its probability. For the 2- to 4-bit subbands of the quantized MDCT coefficients, Figure 3 describes an example of the probability $p(C)$ ($C \in \{0,\ldots 15\}$) and the conditional probability $p(C|A)$ (A = 0 is the neighboring encoded symbol) of the current encoding symbol C, i.e., the code indexes of the quantized MDCT coefficients. The solid line represents the conditional probability $p(C|A)$, and the dotted line describes the probability $p(C)$. It can be found in Figure 3 that the conditional probability distribution $p(C|A)$ is more concentrated in the vicinity of the current encoding index 0 than the probability $p(C)$, and the relationship of the two kinds of probability (shown by dotted line and solid line) satisfies the Equation 7.

Thus, for the 2- to 4-bit subbands, the context in the frequency domain is defined as the encoded symbol A before the input one C, as is shown in Figure 1. Then, the conditional cumulative counts $c(C|A)$ can be obtained. Let $c(C|A)$ be the estimated conditional cumulative counts to drive the integer arithmetic coder.

In the time domain, $\gamma_j(n)$ is defined as the correlation coefficient in the previous adjacent subbands with the same bit allocation

$$\gamma_j(n) = 1 - \frac{\sum_{i=1}^{n}\left[\frac{\left|D_{i,j}^{'(t)} - D_{i,j}^{'(t+1)}\right|}{2^b/8}\right]}{n}, \tag{20}$$

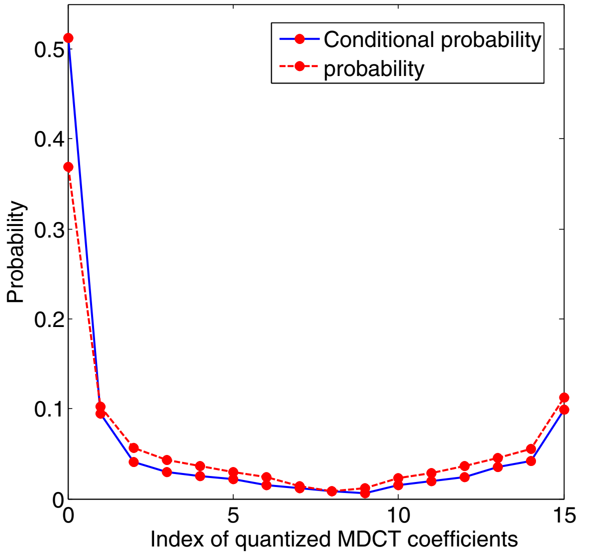

Figure 3 The probability and conditional probability of the encoding symbol MDCT indexes. The *x*-axis represents the index of quantized MDCT coefficients, and the *y*-axis represents the probability of the encoding symbol. The solid line represents the conditional probability, and the dotted line describes the probability.

$$D_{i,j}^{'(t)} = \begin{cases} D_{i,j}^{(t)}, & 0 \le D_{i,j}^{(t)} \le 2^{b-1} - 1 \\ 2^{b-1} - D_{i,j}^{(t)} & 2^{b-1} < D_{i,j}^{(t)} \le 2^b - 1 \end{cases}, \tag{21}$$

$$D_{i,j}^{'(t+1)} = \begin{cases} D_{i,j}^{(t+1)}, & 0 \le D_{i,j}^{(t+1)} \le 2^{b-1} - 1 \\ 2^{b-1} - D_{i,j}^{(t+1)} & 2^{b-1} < D_{i,j}^{(t+1)} \le 2^b - 1 \end{cases}, \tag{22}$$

where $D_{i,j}^{(t)}$ represents subband index with $1 \le j \le 44$ and j means the number of the subbands. The subbands have different sizes n = 8, 16, 24, 32 that increase with the increasing frequency. The character b represents the bits allocated for the current frame and 2^b is just the number of symbols for the m-ary (m symbols) adaptive arithmetic coding, i.e., $m = 2^b$.

If $\gamma_j(n) \ge 0.5$, then the context in the time domain is employed in the present adjacent subbands with the same bit allocation. By statistical analysis, we have found that the audio coding parameters for music signal have higher correlation than the speech signal in time domain. As to the quantized norms in G.719, a large percentage, 98.9%, of all the frames have the correlation (i.e., the correlation coefficient is higher than 0.5) between the adjacent frames which enables larger compression.

Given the encoded symbol in the previous frame, referred to as B, there is a large possibility of the input symbol C distributing around B. In G.719, for the m-ary

(m symbols) adaptive arithmetic coding, the encoded symbol B is the center; $m/2$ symbols, which are located in the range of B and $m - B$ (provided by $- B$ to avoid negative symbol), would be chosen to add $\lambda = \sum_{i=1}^{m} f(i)$ on the basis of the original frequency, and $\delta = m/8$, which can guarantee that the probability of half of all symbols is increased.

That is,

$$f^{'}(i) = \begin{cases} f(i)+\lambda; & i = B–m/8+1, \ldots, B, \ldots, B+m/8 \\ f(i); & others \\ f(i)+\lambda; & i = (m–B)–m/8+1, \ldots, m–B, \ldots, (m–B)+m/8 \end{cases} . \tag{23}$$

As is depicted in Figure 4 which gives the behavior of MDCT parameters with 5 to 9 bits, the dotted line describes the original frequency counts ($f(i)$in formula (23)) of all symbols, while the solid line presents the final frequency counts ($f^{'}(i)$in formula (23)) of all symbols. It is shown that the solid line is higher than the dotted line which indicates that the subinterval for $i = B - m/8 + 1, \ldots, B, \ldots, B + m/8$ will be larger resulting from their higher final frequency counts. After the operation of encoding, the model frequency distribution returns to the original probability distribution, then its updating takes place.

3.4 Variable rate in G.719

Variable rate coding methods [18-20] are important for source compression, and they have been studied for many years especially in speech codec. This paper introduces an efficient variable rate algorithm for G.719 based on the proposed adaptive arithmetic coding together with the original Huffman coding module. Figure 5 shows the block diagram of the variable rate scheme.

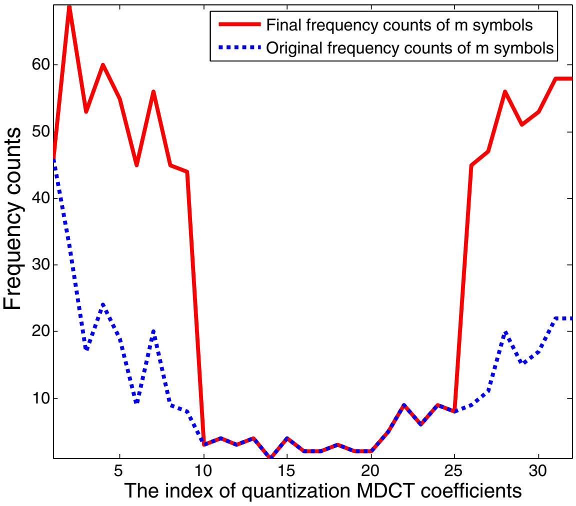

Figure 4 The estimation of the probability by the context model in time domain. The *x*-axis represents the index of quantization MDCT coefficients, and the *y*-axis represents frequency counts. The solid line represents the final frequency counts of all symbols and the dotted line indicates the final frequency counts of all symbols.

The bit rate is determined through three steps. The module of Huffman coding is kept to calculate the saving bits and prepare for the bit allocation. Let Sum be the total bits at a fixed bit rate. Firstly, the norms are coded by both the original Huffman coding consuming $h1$ bits and the context-based adaptive arithmetic coding consuming $a1$ bits simultaneously. Compared to the Huffman coding, the context-based adaptive arithmetic coding can save bits $L1 = h1 - a1$. The remaining bits $num1 = Sum - h1$ are used for bit allocation of the quantized MDCT coefficients. In the second step, the subbands with different bits assigned by the bit allocation are encoded by the proposed adaptive arithmetic coding. The quantized MDCT coefficients are also encoded by Huffman coding consuming $h2$ bits to calculate the remaining bits $num2 = Sum - h1 - h2$ used for the noise level adjustment. Compared to the Huffman coding, the number of bits used for coding the quantized MDCT coefficients with the context-based adaptive arithmetic coding is $a2$, which can save bits $L2 = h2 - a2$. Finally, the noise level is adjusted according to $num2$. The total bits and the bits used for the bit allocation and noise level adjustment in the improved encoder remain the same as those in the primary fixed rate G.719; hence, the saving bits $L1 + L2$ (provided by the context-based adaptive arithmetic coding compared to the original Huffman coding) lead to the variable rate of G.719. To ensure the correct decoding, the header in G.719 [7] which specifies the number of bits used for encoding is changed to indicate variable bits instead of fixed bits.

4. Experimental results

4.1 Bit rate comparison

In this section, the performance of the variable rate coder, which employs the novel context-based adaptive arithmetic coding, is evaluated from the point of view of the average bit rate. The samples used in the bit rate measurement are ten speech and 29 music, including three classical music, ten mixed music (music and speech), three orchestras, one folk, one guitar, two harps, two percussions, one pop, three saxophones, and three trumpets. Each sample is sampled at the rate of 48 kHz and lasts 10 s. Table 1 summarizes the average bit rates of the improved variable rate G.719 at low bit rate mode compared with those of the fixed rate G.719 at 32 kb/s.

As is shown in Table 1, our scheme achieves an average bit rate from 29.4817 to 29.9606 kb/s at low bit rate coding mode, compared with the fixed rate 32 kb/s. The coding gains of the three types of signal have a range from 6.4% to 7.9%, and it shows a coding gain on average 7.2% for all the test samples. Particularly, the bit rate saving for music signal is the largest compared with the mixed music

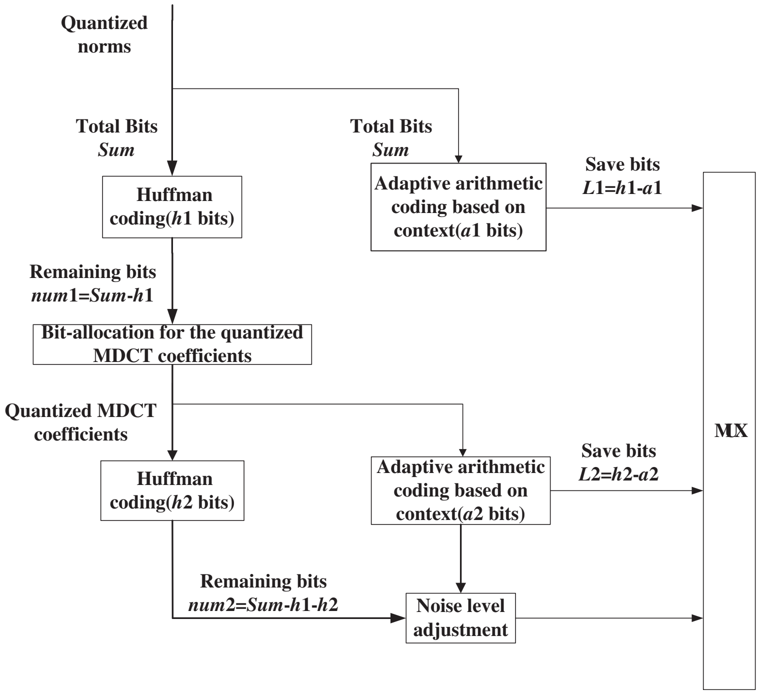
Quantized norms
Total Bits Sum
Total Bits Sum
Huffman coding(h1 bits)
Adaptive arithmetic coding based on context(a1 bits)
Save bits L1=h1-a1
Remaining bits num1=Sum-h1
Bit-allocation for the quantized MDCT coefficients
Quantized MDCT coefficients
MUX
Huffman coding(h2 bits)
Adaptive arithmetic coding based on context(a2 bits)
Save bits L2=h2-a2
Remaining bits num2=Sum-h1-h2
Noise level adjustment

Figure 5 **The basic structure of the variable rate encoder with introducing the context-based adaptive arithmetic coding.**

signal and speech signal because of its good correlation in time domain and frequency domain.

Table 2 shows the coding modes in G.719, and we carried out experiments at all coding modes. As the bit rate increases, the context-based adaptive arithmetic coding scheme achieves a better coding gain compared with the original Huffman coding especially for the highest bit rate coding mode. The test shows an average coding gain of 9.1% at the highest bit rate (coding mode 7 in Table 2). Specifically, music processing shows an average coding gain of 10.9% at the highest bit rate, which indicates the good statistical characteristic for pure music.

In order to have a good knowledge of the performance of the proposed adaptive arithmetic coding, we also carried out experiments to compare the different improved coders. Figure 6 presents the bit rate of G.719 fixed rate coder with Huffman coding, G.719 variable rate coder with the adaptive arithmetic coding and the context-based adaptive arithmetic coding at different coding modes, as is shown in Table 2. Compared with the adaptive arithmetic coding, the context-based adaptive arithmetic coding has a better performance. The lower the bit rate is, the higher the average coding gain is achieved when the context-based adaptive arithmetic coding is compared with the common adaptive arithmetic coding. The test shows a gain of 2.3% with the context-based adaptive arithmetic coding at the lowest bit rate (coding mode 1 in Table 2).

Table 1 Average bit rate of different signal type

Signal type	Average bit rate in fixed rate G.719 (kb/s)	Average bit rate in variable rate G.719 (kb/s)
Music	32	29.4817
Mixed music	32	29.6640
Speech	32	29.9606
Total	32	29.6512

4.2 Investigation of the short-term coding efficiency

The bit rate comparison in Section 4.1 shows the overall bit rate reduction that reflecting the long-term average of coding efficiency performance. In order to investigate the short-term coding efficiency of the proposed variable rate arithmetic coding, the bit allocation is evaluated frame by frame, and the performance is shown in Table 3.

As it can be seen from Table 3, the minimum bits of each frame in the variable rate G.719 are less than that in the fixed rate G.719, and the maximum bits of each frame

Table 2 Coding modes in G.719

Coding mode	Fixed rate (kb/s)	Variable rate (kb/s)
1	32	29.6512
2	48	44.8484
3	64	59.8988
4	80	74.5056
5	96	88.6831
6	112	102.5974
7	128	116.3237

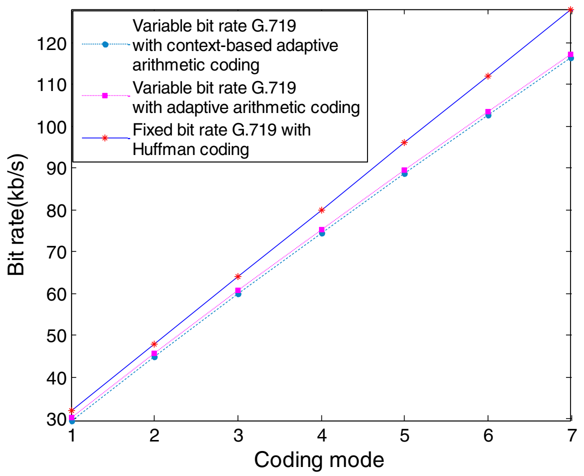

Figure 6 Average bit rate at different coding modes. The *x*-axis represents the coding modes (1 to 7), and the *y*-axis represents the average bit rate (kb/s). The solid line represents the fixed bit rate of G.719 using Huffman coding, the dotted line represents the variable bit rate of G.719 using the adaptive arithmetic coding, and the dash line represents the variable bit rate of G.719 using the context-based adaptive arithmetic coding.

in the variable rate G.719 are more than that in the fixed rate G.719 only because the context model tends to be stable after the first several input frames. Through statistical analysis, there is an extraordinarily large percentage, 99.1%, of all the frames needing less than the fixed 640 bits, which guarantees the short-term coding efficiency of the proposed variable rate arithmetic coding. Since the good correlation in the time domain and in the frequency domain, the minimum bits in the variable rate G.719 for music signal have the best performance.

4.3 The performance comparison of different entropy coding

A comparative study of different entropy coding schemes will be presented in this section, which includes Huffman coding, the adaptive arithmetic coding, and the context-based adaptive arithmetic coding, respectively. Table 4 shows the average number of bits to code the quantized norms using different coding schemes under different coding modes, while Table 5 presents the average number of bits to code the quantized MDCT coefficients using different coding schemes under different coding modes.

Table 3 The performance of bit allocation of each frame

Signal type	Bits of each frame in fixed rate G.719 (bits/frame)	The minimum bits in variable rate G.719 (bits/frame)	The maximum bits in variable rate G.719 (bits/frame)
Music	640	495	725
Mixed music	640	540	718
Speech	640	550	714

Table 4 The average number of bits when coding the quantized norms

Coding bits for the quantized norms	Huffman coding	Adaptive arithmetic coding	Context-based adaptive arithmetic coding
Modes 1 to 7	147.1909	132.0607	119.8233

As it can be seen from the two tables, the coding bits required for the quantized norms and the quantized MDCT coefficients are the least using the proposed context-based adaptive arithmetic coding. Since the energy of all subbands will not change at different coding mode, the coding bits of the quantized norms remain the same along the different modes.

In order to further understand the compression degree between the adaptive arithmetic coding and Huffman coding, and the compression degree between the context-based adaptive arithmetic coding and Huffman coding, the compression percentage can be calculated according to the following formulas:

$$\Delta 1 = \frac{\text{h bits}-\text{a bits}}{\text{h bits}} \times 100\%, \tag{24}$$

$$\Delta 2 = \frac{\text{h bits}-\text{ca bits}}{\text{h bits}} \times 100\%, \tag{25}$$

where h_bits represents the bits for encoding the audio parameters by Huffman coding, a_bits represents the bits for encoding parameters by the adaptive arithmetic coding, and ca_bits represents the bits for encoding parameters by the proposed context-based adaptive arithmetic coding. Tables 6 and 7 present the compression percentage of the quantized norms and the quantized MDCT coefficients which are calculated by Equations 24 and 25.

As it can be seen from Tables 6 and 7, the compression percentage of the quantized norms is higher than that of the quantized MDCT coefficients. Since the variation of the quantized norms is less than that of the quantized MDCT coefficients, the conditional probability of the encoding symbol of the quantized norms is bigger than that of the quantized MDCT coefficients. Moreover, the

Table 5 The average number of bits when coding the quantized MDCT coefficients

Coding bits for the quantized MDCT coefficients	Huffman coding	Adaptive arithmetic coding	Context-based adaptive arithmetic coding
Mode 1	429.8683	418.1427	417.3993
Mode 2	720.0894	695.6415	694.4265
Mode 3	1,011.484	972.396	970.7839
Mode 4	1,305.252	1,250.046	1,247.866
Mode 5	1,604.597	1,532.503	1,530.054
Mode 6	1,860.674	1,773.074	1,770.578
Mode 7	2,239.736	2,131.121	2,128.619

Table 6 The compression percentage of the quantized norms

Compression percentage (%)	Adaptive arithmetic coding	Context-based adaptive arithmetic coding
Modes 1 to 7	10.27932	18.59329

correlation in the time domain of the quantized norms is higher than that of the quantized MDCT coefficients because of the less variation of norms. As a result, the scheme of the context-based adaptive arithmetic coding used for the quantized norms has a better performance than that used for the quantized MDCT coefficients.

Figure 7 presents the compression percentage of all kinds of the parameters with different entropy coding. The solid line presents the compression percentage of the quantized norms coded by the context-based adaptive arithmetic coding compared to Huffman coding. The dashed line presents the compression percentage of the quantized norms coded by the adaptive arithmetic coding compared to Huffman coding. The dotted line presents the compression percentage of the quantized MDCT coefficients coded by the context-based adaptive arithmetic coding compared to Huffman coding. The dash dotted line presents the compression percentage of the quantized MDCT coefficients coded by the adaptive arithmetic coding compared to Huffman coding. It can be seen that the proposed context-based adaptive arithmetic coding performs better than the adaptive arithmetic coding when coding both norms and MDCT coefficients, especially when the frequency band gains are coded.

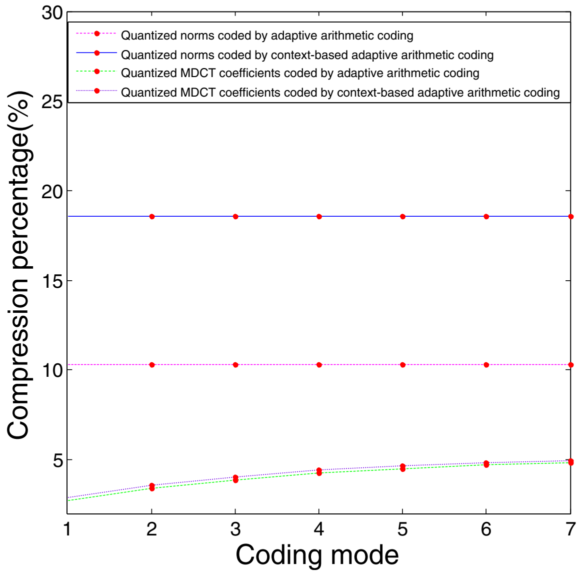

Figure 7 Average compression percentages of quantized norms and quantized MDCT coefficients. The *x*-axis represents the coding modes (1 to 7), and the *y*-axis represents the compression percentage of all kinds of the parameters with different entropy coding. The solid line presents the compression percentage of the quantized norms coded by the context-based adaptive arithmetic coding compared to Huffman coding. The dashed line presents the compression percentage of the quantized norms coded by the adaptive arithmetic coding compared to Huffman coding. The dotted line presents the compression percentage of the quantized MDCT coefficients coded by context-based adaptive arithmetic coding compared to Huffman coding. The dash dotted line presents the compression percentage of the quantized MDCT coefficients coded by the adaptive arithmetic coding compared to Huffman coding.

4.4 Audio quality

The proposed context-based arithmetic coding is performed directly on the quantized audio parameters, and the technique is lossless, so the decoded parameters using the proposed arithmetic coding method should have no distortion. In the quality tests to evaluate the arithmetic coding, objective comparison tests would be firstly used to verify the lossless coding. By the objective comparison, i.e., PEAQ [21] over a large number of speech and music samples, all samples generated by the proposed variable rate G.719 appear the same as those of the fixed rate G.719. Secondly, we carry out the preferable listening tests to verify that the proposed scheme does not introduce any kind of undesirable effects although there is no need to use subjective listening tests if the sample values are not changed. It is thus verified that the proposed variable rate coder has the same audio quality as the original G.719 under the different coding modes. Besides, we use the audio comparing tool 'CompAudio' [22] to check if all the sample values are equal before and after the arithmetic coding.

Table 7 The compression percentage of the quantized MDCT coefficients

Compression percentage (%)	Adaptive arithmetic coding	Context-based adaptive arithmetic coding
Mode 1	2.727712	2.900664
Mode 2	3.395124	3.563846
Mode 3	3.864423	4.023802
Mode 4	4.229515	4.396522
Mode 5	4.492942	4.645568
Mode 6	4.707953	4.842141
Mode 7	4.849440	4.961144

Table 8 Average complexity comparison test results in terms of WMOPS

Signal type	Fixed rate G.719		Variable rate G.719		The proposed modules	
	Encoder	Decoder	Encoder	Decoder	Encoder	Decoder
Music	6.5986	6.2988	9.6376	8.8378	3.0390	2.5390
Mixed music	6.6356	6.3944	9.6641	8.9429	3.0285	2.5485
Speech	6.6854	6.3825	9.7223	9.2516	3.0369	2.8691
Total	6.6298	6.3288	9.6643	8.9808	3.0345	2.6520

Through careful audio quality evaluation and the value comparison, the proposed context-based adaptive arithmetic coding actually leads to lossless compression used for the quantized audio parameters. It is verified that the proposed technique is lossless and the detailed test results need not to be reported. As to the audio qualities of the full codec (e.g., ITU-T G.719), the formal test results can be found in [23,24].

4.5 Complexity test

The computational complexity obtained per frame can be specified in terms of the weighted million operations per second (WMOPS) and can be evaluated by the average running time. The coding rate is set to 32 kbps. The processor of the computer is the Intel Core 2 Duo processor (Intel, Santa Clara, CA, USA). The basic frequency is 1.8 GHz. Each frame has a length of 960 samples. Table 8 shows the average complexity of the original fixed rate G.719, the proposed variable rate G.719, and the proposed adaptive arithmetic coding modules based on context. The encoder and decoder complexities are computed separately. In fact, the proposed adaptive arithmetic coding itself results in the increase of the complexity in the new scheme. The additive complexity of the proposed entropy coding modules can be acceptable in some applications because of the intrinsic low complexity of G.719 codec. However, almost 50% increase in total complexity should be considered to be optimized if very low complexity is actually needed.

5. Conclusions

The novel context-based adaptive arithmetic coding technique proposed in this paper behaves promising and significant for the lossless compression when both the time and frequency plane of the audio coding parameters are considered. The proposed technique has been introduced to compress the quantized MDCT coefficients and the quantized norms in G.719. Variable rate coding structure has also been investigated and adopted to obtain high coding efficiency compared with the original fixed rate G.719. Experiments have shown that the new technique achieves a coding gain of 6% to 10% at all coding modes for different types of signals, appearing to be advantageous over the conventional Huffman coding. To evaluate the performance of the proposed algorithm, objective and subjective quality tests have been done for a variety of speech and audio samples. The average bit rates and computation complexity have also been computed at different coding modes. It is verified that the proposed variable rate coder with the adaptive arithmetic coding based on the time-frequency context produces the same audio quality as the original G.719 coder while achieving a high coding gain. The proposed method in this paper can be easily used in other audio codecs which need to lower the coding bit rate by means of entropy coding.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank the reviewers for their suggestions which have contributed a lot to the great improvement of the manuscript. The work in this paper is supported by the National Natural Science Foundation of China (no.11161140319), and the corporation between BIT and Ericsson.

Received: 24 September 2012 Accepted: 2 May 2013
Published: 21 May 2013

References

1. PM Fenwick, Huffman code efficiencies for extensions of sources. IEEE Trans. Commun. **43**(234), 163–165 (1995). doi:10.1109/26.380027
2. DA Huffman, A method for construction of minimum redundancy codes. Proc. IRE **40**(9), 1098–1101 (1952). doi:10.1109/JRPROC.1952.273898
3. GG Langdon, An introduction to arithmetic coding. IBM J. Res. Dev. **28**(2), 135–149 (1984). doi:10.1147/rd.282.0135
4. K Hyungjin, W Jiangtao, JD Villasenor, Secure arithmetic coding. IEEE Trans. Signal Process. **55**(5), 2263–2272 (1987). doi:10.1109/TSP.2007.892710
5. Information technology, Coding of Audio-Visual Objects - Part 3, Audio, Subpart 4: Time/Frequency Coding, in *International Organization for Standardization*. ISO/IEC 14496–3:1999, 1999
6. M Neuendorf, P Gournay, M Multrus, J Lecomte, B Bessette, R Geiger, S Bayer, G Fuchs, J Hilpert, N Rettelbach, R Salami, G Schuller, R Lefebvre, B Grill, Unified speech and audio coding scheme for high quality at low bitrates, in *Proc of IEEE Int Conf Acoustics, Speech and Signal Processing*, 2009, pp. 1–4. doi:10.1109/ICASSP.2009.4959505
7. ITU-T Recommendation, *G.719 (06/08), Low-complexity full-band audio coding for high-quality conversational applications* (Int Telecomm Union, Geneva, 2008)
8. L Zhang, X Wu, N Zhang, W Gao, Q Wang, D Zhao, Context-based arithmetic coding reexamined for DCT video compression, in *IEEE International Symposium on Circuits and Systems* (New Orleans, 2007), pp. 3147–3150. doi:10.1109/ISCAS.2007.378098
9. B Ryabko, J Rissanen, Fast adaptive arithmetic code for large alphabet sources with asymmetrical distributions. IEEE Commun. Lett. **7**(1), 33–35 (2003). doi:10.1109/LCOMM.2002.807424
10. D Marpe, H Schwarz, T Wiegand, Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. IEEE T Circ Syst Vid **13**(7), 620–636 (2003)
11. Information technology - MPEG audio technologies, *International Organization for Standardization* (ISO/IEC). ISO/IEC 23003–3: 2012
12. CE Shannon, A mathematical theory of communications. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
13. G Fuchs, V Subbaraman, M Multrus, Efficient context adaptive entropy coding for real-time applications, in *Proc of IEEE Int Conf Acoustics, Speech and Signal Processing*, 2011, pp. 493–496. doi:10.1109/ICASSP.2011.5946448
14. H Moradmand, A Payandeh, MR Aref, Joint source-channel coding using finite state integer arithmetic codes, in *IEEE International Conference on Electro/Information Technology* (Windsor, 2009), pp. 19–22. doi:10.1109/EIT.2009.5189577
15. YM Huang, YC Liang, A secure arithmetic coding algorithm based on integer implementation, in *International Symposium on Communications and Information Technologies* (Hangzhou, 2011), pp. 518–521. doi:10.1109/ISCIT.2011.6092162
16. IH Witten, RM Neal, JG Cleary, Arithmetic coding for data compression. Communication of the ACM **30**(6), 520–540 (1987). doi:10.1145/214762.214771
17. Y Chen, H Zhu, H Jin, X-H Sun, *Improving the effectiveness of context-based prefetching with multi-order analysis* (International Conference on Parallel Processing Workshops, San Diego, 2010), pp. 428–435. doi:10.1109/ICPPW.2010.64
18. O Pasi, Toll quality variable-rate speech codec. Int Conf Acoust Spee **2**, 747–750 (1997)
19. E Dong, H Zhao, Y Li, Low bit and variable rate speech coding using local cosine transform. Proceedings of TENCON. on Computers, Communications, Control and Power Engineering. **1**, 28–31 (2002)
20. S McClellan, JD Gibson, Variable rate CELP based on subband flatness. IEEE T Speech Audi P **5**(2), 120–130 (1997). doi:10.1109/89.554774

21. ITU-R Recommendation, *BS.1387-1 (11/01), Method for Objective Measurements of Perceived Audio Quality* (Int Telecomm Union, Geneva, 2001)
22. P Kabal, CompAudio. (1996). http://www.csee.umbc.edu/help/sound/AFsp-V2R1/html/audio/CompAudio.html. Accessed 20 January 2013
23. M Xie, P Chu, A Taleb, M Briand, ITU-T G.719, A new low-complexity full-band (20 kHz) audio coding standard for high-quality conversational applications, in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (New Paltz, 2009), pp. 265–268. doi:10.1109/ASPAA.2009.5346487
24. A Taleb, S Karapetkov, G.719: The first ITU-T standard for high-quality conversational full-band audio coding. IEEE Communication Magazine **47**(10), 124–130 (2009). doi:10.1109/MCOM.2009.5273819

doi:10.1186/1687-4722-2013-9
Cite this article as: Wang *et al.*: Context-based adaptive arithmetic coding in time and frequency domain for the lossless compression of audio coding parameters at variable rate. *EURASIP Journal on Audio, Speech, and Music Processing* 2013 **2013**:9.