Embedded High-Quality Multichannel Audio Coding

Dai Yang, Hongmei Ai, Chris Kyriakakis and C.-C. Jay Kuo *

Integrated Media Systems Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, CA 90089-2564, USA

ABSTRACT

An embedded high-quality multichannel audio coding algorithm is proposed in this research. The Karhunen-Loeve Transform (KLT) is applied to multichannel audio signals in the pre-processing stage to remove inter-channel redundancy. Then, after processing of several audio coding blocks, transformed coefficients are layered quantized and the bit stream is ordered according to their importance. The multichannel audio bit stream generated by the proposed algorithm has a fully progressive property, which is highly desirable for audio multicast applications in heterogeneous networks. Experimental results show that, compared with the MPEG Advanced Audio Coding (AAC) algorithm, the proposed algorithm achieves a better performance with both the objective MNR (Mask-to-Noise-Ratio) measurement and the subjective listening test at several different bit rates.

Keywords: Multichannel audio, Progressive coding, Karhunen-Loeve Transform, Successive Quantization, Context-based QM Coder

1. INTRODUCTION

Multichannel audio technologies have become much more mature these days partially pushed by the need of the film industry and home entertainment systems. Starting from the monophonic technology, new systems such as stereophonic, quadraphonic, 5.1 channels and 10.2 channels are penetrating into the market very quickly. Compared with the mono- or stereo-sound, multichannel audio provides end-users a more compelling experience and becomes more appealing to music producers.[1] As a result, an efficient coding scheme for multichannel audio's storage and transmission becomes a more challenging problem. Among several existing multichannel audio compression algorithms, Dolby AC-3 and MPEG Advanced Audio Coding (AAC) are two most prevalent perceptual digital audio coding systems. However, they can only provide fixed bit rate perceptually lossless coding at about 64 kbits/sec/ch. In order to transfer high quality multichannel audio through the variable bandwidth network, a fully-embedded audio compression algorithm, which is able to transfer audio signals from coarse to fine qualities progressively, is highly desirable.

With a deeper understanding of wavelet and DCT transforms and the bit-layer coding principle, progressive image coding has progressed significantly during the last decade. The emerging wavelet-based JPEG2000 image compression standard can provide a quality-scalable bit stream of coded images over the Internet. The embedded coding and the streaming technologies allow users to display images of different qualities in a progressive fashion. That is, with more and more bits arriving, the quality of the image is gradually refined until it is indistinguishable from the original one perceived by the human visual system. Embedded coders also find applications in image multicast when the server only has to store one copy of the image file and truncate the bit stream to a proper location depending on the bandwidth of each receiver.

In the scenario of progressive audio coding, quality scalability can be used effectively in an audio multicast environment. A single audio bit stream can be stored in the server and transmitted over the network to users of different receiving bandwidth. For users who have a broad-band connection, they can receive more data and enjoy reconstructed audio of better quality. In contrast, for users with only the narrow-band connection, they can still enjoy the target audio but with poorer quality.

Compared with progressive image coding, the difficulty of progressive audio coding lies in the role played by the human auditory system. The SNR measure, which is commonly used in evaluating the performance of a coded image, is not very useful to evaluate the quality of compressed audio. How to determine a new scheme that achieves

* Email: daiyang@usc.edu, ahm@costard.usc.edu, ckyriak@imsc.usc.edu, cckuo@sipi.usc.edu

quality scalability according to the psychoacoustic model is the key issue in the design of a progressive audio codec. Another challenge is how to organize and transmit audio signals in different channels for multichannel audio coding.

Being inspired by progressive image coding and the MPEG AAC (advanced audio coding) system, we develop a novel fully-embedded multichannel audio compression algorithm in this work. First, a Karhunen-Loeve Transform (KLT) is performed in the pre-processing stage to remove inter-channel redundancy inherent in original physical channels. Then, we exploit most coding blocks of the AAC main profile encoder to generate spectral coefficients. Finally, a progressive transmission strategy and a context-based QM coder are adopted to obtain the fully quality-scalable multichannel audio bit stream. The key feature of the proposed algorithm is that it incorporates the successive approximation quantization (SAQ), the context-based arithmetic coder and the unique Rate-Distortion metric, Mask-to-Noise-Ration (MNR), in audio coding to achieve perceived quality scalability of multichannel audio coding.

This paper is organized as follows. An inter-channel redundancy removal method is briefly introduced in Section 2. Then, two unique embedded codec design techniques are explained in Section 3. Section 4 describes the channel and subband selection rules adopted in our proposed algorithm. Section 5 discussed some implementation issues. And section 6 illustrates the complete compression system. Finally some experimental results and conclusion remarks are given in section 7 and section 8 respectively.

2. INTER-CHANNEL DECORRELATION

It has been observed that multichannel audio sources, especially those captured and recorded in a real event, have high inter-channel correlation.[2] In order to efficiently compress multichannel audio, a pre-processing stage which consists of KLT across channels is performed to reduce the inter-channel dependency. Figure 1 illustrates how KLT is performed on multichannel audio signals, where columns of the KL transform matrix are composed of eigenvectors of the covariance matrix associated with the original multichannel audio signals.

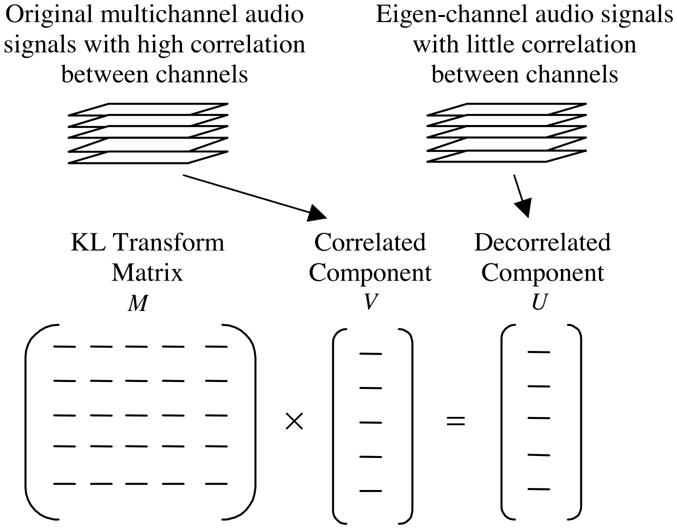

Original multichannel audio signals with high correlation between channels
Eigen-channel audio signals with little correlation between channels
KL Transform Matrix
M
Correlated Component
V
Decorrelated Component
U
×
=

Figure 1. Removal of inter-channel correlation via KLT.

After the pre-processing stage, signals in these relatively independent channels called eigen-channels are further processed. For more details of the KLT pre-processing technique, we refer to work of Yang *et al.*[2].[3] In the next section, we examine efficient methods to encode the signal in each eigen-channel independently.

3. MAIN FEATURES OF EMBEDDED AUDIO CODEC

The major difference between the proposed progressive audio codec and other existing audio codecs such as AAC lies in the quantization module and the entropy coding module. The dual iteration loop used in AAC to calculate the quantization step size for each frame's and each channel's coefficients is replaced by a progressive quantization block. The huffman coding module used in the AAC to encode quantized data is replaced by a context-based QM coder. They will be explained in detail below.

3.1. Successive Approximation Quantization (SAQ)

The most important component of the quantization module is called successive approximation quantization (SAQ). The SAQ scheme, which is adopted by all embedded wavelet coders for progressive image coding, is crucial to the design of embedded coders. The motivation for successive approximation is built upon the goal of developing an embedded code that is in analogy to find an approximation of binary-representation to a real number.[4] Instead of coding every quantized coefficient as one symbol, SAQ processes the bit representation of coefficients via bit layer sliced in the order of their importance. Thus, SAQ provides a coarse-to-fine, multiprecision representation of the amplitude information. The bit stream is organized such that a decoder can immediately start reconstruction based on the partial received bit stream. As more and more bits are received, more accurate coefficients and higher quality multichannel audio can be reconstructed.

3.1.1. Description of the SAQ Algorithm

SAQ sequentially applies a sequence of thresholds $T_0, T_1, \ldots, T_{N+1}$ for refined quantization, where these thresholds are chosen such that $T_i = T_{i-1}/2$. The initial threshold T_0 is selected such that $|C(i)| < 2T_0$ for all transformed coefficients in one subband, where $C(i)$ represents the i^{th} spectral coefficient in the subband. To implement SAQ, two separate lists, the dominant list and the subordinate list, are maintained both at the encoder and the decoder sides. At any point of the process, the dominant list contains the coordinates of those coefficients that have not yet been found to be significant. While the subordinate list contains magnitudes of those coefficients that have been found to be significant. The process that updates the dominate list is called the significant pass, and the process that updates the subordinate list is called the refinement pass.

In the proposed algorithm, SAQ is adopted as the quantization method for each spectral coefficient within each subband. This algorithm, which is borrowed from Kuo *et al.*'s work,[5] is listed below.

Successive Approximation Quantization (SAQ) Algorithm

1. Initialization:
   For each subband, find out the maximum absolute value $C_{\max}$ for all coefficients $C(i)$ in the subband, and set the initial quantization threshold to be $T_0 = C_{\max}/2 + BIAS$, where $BIAS$ is a small constant.

2. Construction of the significant map (significance identification):
   For each $C(i)$ contained in the dominant list, if $C(i) \geq T_k$, where T_k is the threshold of the current layer (layer k), add i to the significant map, remove i from the dominant list and encode it with $'1s'$, where $'s'$ is the sign bit. Moreover, modify the coefficient's value to

$$C(i) \leftarrow \begin{cases} C(i) - 1.5 \times T_k, & \forall C(i) > 0 \\ C(i) + 1.5 \times T_k, & \text{otherwise} \end{cases}$$

3. Construction of the refinement map (refinement):
   For each $C(i)$ contained in the significant map, encode the bit at layer k with a refinement bit $'D'$ and change the value of $C(i)$ to

$$C(i) \leftarrow \begin{cases} C(i) - 0.25 \times T_k, & \forall C(i) > 0 \\ C(i) + 0.25 \times T_k, & \text{otherwise} \end{cases}$$

4. Iteration:
   Set $T_{k+1} = T_k/2$ and repeat Steps 2-4 for $k = 0, 1, 2, \ldots$

3.1.2. Analysis of Error Reduction Rates

The following two points have been observed in Kuo *et al.*'s work[5]:

- The coding efficiency of the significant map is always better than that of the refinement map at the same layer.
- The coding efficiency of the significant map at the k^{th} layer is better than that of the refinement map at the $(k-1)^{th}$ layer.

In the following, we would like to provide a formal proof by analyzing the error reduction capability due to the significant pass and the refinement pass, respectively.

First, let us consider the error reduction capability for the bit-layer coding of coefficient $C(i)$, $\forall i$, in the significant pass. Since the sign of each coefficient will be coded separately, we will assume $C(i) > 0$ below without loss of generality. Suppose that $C(i)$ becomes significant at layer k. This means $T_k \leq C(i) < T_{k-1} = 2T_k$ and its value is modified accordingly. Then, error reduction Δ_1 due to the coding of this bit can be found as

$$\Delta_1 = C(i) - |C(i) - 1.5 \times T_k|.$$

Note that, at any point of the process, the value of $|C(i)|$ is nothing else but the remaining coding error. Since $T_k \leq C(i) < 2T_k$, $-0.5T_k < C(i) - 1.5T_k \leq 0.5T_k$, we have $|C(i) - 1.5T_k| \leq 0.5T_k$. Consequently,

$$\Delta_1 = C(i) - |C(i) - 1.5 \times T_k| \geq 0.5T_k.$$

Now, let us calculate the error reduction for the bit-layer coding of coefficient $C(j), \forall j$, in the refinement pass. Similar to the previous case, we assume $C(j) > 0$. At layer k, suppose $C(j)$ is being refined, and its value is modified accordingly. The corresponding error reduction is

$$\Delta_2 = C(j) - |C(j) - 0.25 \times T_k|.$$

Two cases have to be considered:

1. If $C(j) \geq 0.25T_k$,
$$\Delta_2 = C(j) - C(j) + 0.25T_k = 0.25T_k.$$
2. If $C(j) < 0.25T_k$,
$$\Delta_2 = C(j) + C(j) - 0.25T_k = 2C(j) - 0.25T_k < 0.5T_k - 0.25T_k = 0.25T_k.$$

Thus, we conclude that

$$\Delta_2 = C(j) - |C(j) - 0.25 \times T_k| \leq 0.25T_k < 0.5T_k \leq \Delta_1.$$

Thus, the error reduction for significant pass is always greater than that of the refinement pass at the same layer.

Similarly, at layer $(k-1)$, the error reduction for coefficient $C(j), \forall j$, caused by the refinement pass is

$$\Delta_3 = C(j) - |C(j) - 0.25 \times T_{k-1}| \leq 0.25T_{k-1} = 0.5T_k \leq \Delta_1,$$

which demonstrates that error reduction in the significant pass at layer k is actually greater than or equal to that of the refinement pass at layer $(k-1)$.

According to the above analysis, a refinement-significant map coding is proposed and adopted in our progressive multichannel audio codec. That is, the transmission of k^{th} refinement map of subband i is followed immediately by the transmission of $(k+1)^{th}$ significant map of subband i.

3.1.3. Analysis of Error Bounds

Suppose the i^{th} coefficient $C(i)$ has a value $T_0/2^{R+1} \leq |C(i)| < T_0/2^R$. Then, its binary representation can be written as

$$\begin{aligned} C(i) &= \text{sign} \times [a_0(\frac{T}{2^0}) + a_1(\frac{T}{2^1}) + a_2(\frac{T}{2^2}) + \ldots] \\ &= \text{sign} \times \sum_{k=0}^{\infty} a_k(\frac{T}{2^k}), \end{aligned}$$

where $T = T_0/2^{R+1}$, T_0 is the initial threshold, and a_0, a_1, a_2, ... are binary values (either 0 or 1).

In the SAQ algorithm, $C(i)$ is represented by:

$$\begin{aligned} C(i) &= \text{sign} \times [1.5a_0(\frac{T}{2^0}) + 0.5b_1(\frac{T}{2^1}) + 0.5b_2(\frac{T}{2^2}) + \ldots] \\ &= \text{sign} \times [1.5a_0(\frac{T}{2^0}) + 0.5\sum_{k=1}^{\infty} b_k(\frac{T}{2^k})], \end{aligned}$$

where a_k and b_k are related via

$$a_k = 0.5(b_k + 1), \ \forall k = 1, 2, 3, \ldots,$$

or

$$b_k = \begin{cases} 1, & a_k = 1, \\ -1, & a_k = 0, \end{cases} \quad \forall k = 1, 2, 3, \ldots$$

Based on the first $M+1$ bits $a_0, a_1, a_2, \ldots, a_M$, the reconstructed value $R_1(i)$ by using the binary representation is

$$\begin{aligned} R_1(i) &= \text{sign} \times [a_0(\frac{T}{2^0}) + a_1(\frac{T}{2^1}) + a_2(\frac{T}{2^2}) + \ldots + a_M\frac{T}{2^M}] \\ &= \text{sign} \times [\sum_{k=0}^{M} a_k(\frac{T}{2^k})]. \end{aligned}$$

Based on the first $M+1$ bits $a_0, b_1, b_2, \ldots, b_M$, the reconstructed value $R_2(i)$ by using SAQ is

$$\begin{aligned} R_2(i) &= \text{sign} \times [1.5a_0(\frac{T}{2^0}) + 0.5b_1(\frac{T}{2^1}) + 0.5b_2(\frac{T}{2^2}) + \ldots + 0.5b_M\frac{T}{2^M}] \\ &= \text{sign} \times [1.5a_0(\frac{T}{2^0}) + 0.5\sum_{k=1}^{M} b_k(\frac{T}{2^k})]. \end{aligned}$$

Thus, the error introduced by the binary representation for this coefficient is

$$\begin{aligned} E_1(i) &= |C(i) - R_1(i)| = |\sum_{k=M+1}^{\infty} a_k\frac{T}{2^k}| \\ &\leq \sum_{k=M+1}^{\infty} \frac{T}{2^k} = \frac{T}{2^M}. \end{aligned}$$

Similarly, the error introduced by SAQ for this coefficient is

$$\begin{aligned} E_2(i) &= |C(i) - R_2(i)| = |0.5 \times \sum_{k=M+1}^{\infty} b_k\frac{T}{2^k}| \\ &\leq 0.5\sum_{k=M+1}^{\infty} \frac{T}{2^k} = \frac{T}{2^{M+1}}. \end{aligned}$$

We conclude that the upper bound of both error $E_1(i)$ caused by the binary representation and error $E_2(i)$ cause by SAQ are decaying exponentially when the incoming number M of bits is increasing linearly.

3.2. Context-based QM coder

The QM coder is a binary arithmetic-coding algorithm designed to encode data formed by a binary symbol set. It was the result of the effort by JPEG and JBIG committees, in which the best features of various arithmetic coders are integrated. The QM coder is a lineal descendent of the Q-coder, but significantly enhanced by improvements in the two building blocks, i.e. interval subdivision and probability estimation.[6] Based on the Bayesian estimation, a state-transition table, which consists of a set of rules to estimate the statistics of the bit stream depending on the next incoming symbols, can be derived. The efficiency of the QM coder can be improved by introducing a set of context rules. The QM arithmetic coder achieves a very good compression result if the context is properly selected to summarize the correlation between coded data.

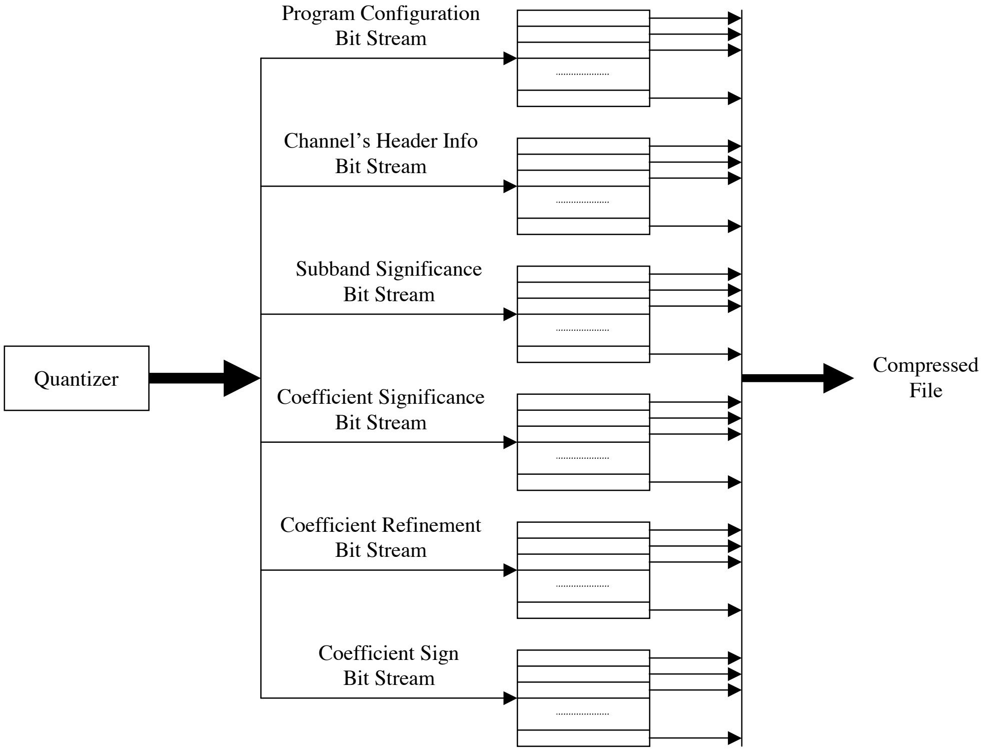

Program Configuration Bit Stream
Channel's Header Info Bit Stream
Subband Significance Bit Stream
Quantizer
Coefficient Significance Bit Stream
Coefficient Refinement Bit Stream
Coefficient Sign Bit Stream
Compressed File

Figure 2. The adopted context-based QM coder with six classes of contexts.

Six classes of contexts are used in the proposed embedded audio codec as shown in Fig 2. They are the general context, the constant context, the subband significance context, the coefficient significance context, the coefficient refinement context and the coefficient sign context. The general context is used in the coding of the configuration information. The constant context is used to encode different channel's header information. As their names suggest, the subband significance context, the coefficient significance context, the coefficient refinement context and the coefficient sign context are used to encode the subband significance, coefficient significance, coefficient refinement and coefficient sign bits, respectively. These contexts are adopted because different classes of bits may have different probability distribution. In principle, separating their contexts should increase the coding performance of the QM coder.

4. CHANNEL AND SUBBAND SELECTION RULES

4.1. Channel Selection Rule

In the embedded multichannel audio codec, we should put the most important bits (in the rate-distortion sense) to the cascaded bit stream first so that the decoder can reconstruct the optimal quality of multichannel audio given a fixed number of bit received. Thus, the importance of channels should be determined for an appropriate ordering of the bit stream. For the normal 5.1 channel configuration, it was observed[3] that the channel importance will be eigen-channel 1, followed by eigen-channels 2 and 3, and then followed by eigen-channels 4 and 5. Between each channel pair, the importance is determined by their energy. This policy is used in this paper.

4.2. Subband Selection Rule

In principle, any quality assessment of an audio channel can be either performed subjectively by employing a large number of expert listeners or done objectively by using an appropriate measuring technique. While the first choice tends to be an expensive and time-consuming task, the use of objective measures provides quick and reproducible results. An optimal measuring technique would be a method that produces the same results as subjective tests while avoiding all problems associated with the subjective assessment procedure.

Unlike image/video coding, techniques for audio quality measurement should deviate from pure signal analysis and move towards a new methodology based on properties of the human auditory system.[7] It is well known that the usual Signal-to-Noise-Ration (SNR) measure completely fails in predicting the quality of coded audio, since quantization noise of perceptual audio coding is spectrally shaped according to psychoacoustic requirements. Nowadays, the most prevalent objective measurement is the Mask-to-Noise-Ratio (MNR) technique, which was first introduced by Brandenburg[8] in 1987. It is the ratio of the masking threshold with respect to the error energy. In our implementation, the masking is calculated from the general psychoacoustic model of the AAC encoder. The psychoacoustic model calculates the maximum distortion energy which is masked by the signal energy, and outputs the Signal-to-Mask-Ratio (SMR).

A subband is masked if the quantization noise level is below the masking threshold so the distortion introduced by the quantization process is not perceptible to human ears. As discussed earlier, SMR represents the human auditory response to the audio signal. If SNR of an input audio signal is high enough, the noise level will be suppressed below masking threshold and the quantization distortion will not be perceived. Since SNR can be easily calculated by

$$SNR = \frac{\sum_i |S_{\text{original}}(i)|^2}{\sum_i |S_{\text{original}}(i) - S_{\text{reconstruct}}(i)|^2},$$

where $S_{\text{original}}(i)$ and $S_{\text{reconstruct}}(i)$ represent the i^{th} original and the i^{th} reconstructed audio signal value, respectively. Thus, MNR is just the difference of SNR and SMR (in dB), or

$$SNR = MNR + SMR.$$

A side benefit of the SAQ technique is that an operational rate vs. distortion plot (or, equivalently, an operational rate vs. the current MNR value) for the coding algorithm can be computed on-line.

The basic ideas behind choosing the subband selection rules are simple. They are:

1. The subband with a better rate deduction capability should be chosen earlier to improve the performance.
2. The subband with a smaller number of coefficients should be chosen earlier to reduce the computational complexity, if the rate reduction performances of two subbands are close.

The first rule implies that we should allocate more bits to those subbands with larger SMR values or smaller MNR values. In other words, we should send out bits belonging to those subbands with larger SMR or smaller MNR values first. The second rule tells us how to decide the subband scanning order. As we know about the subband formation in MPEG AAC, the number of coefficients in each subband is non-decreasing with the increase of the subband number. Figure 3 shows the subband width distribution used in AAC for 44.1 kHz and 48 kHz sampling frequencies and long block frames. Thus, a sequential subband scanning order from the lowest number to the highest number is adopted in this work.

A dual-threshold coding technique is proposed here. One is the MNR threshold, which is used in subband selection. The other is the magnitude threshold, which is used for coefficients' quantization in each selected individual subband. A subband which has its current MNR value smaller than the current MNR threshold is called significant subband. Similar as the SAQ process for coefficient quantization, two lists, i.e. the dominant subband list and the subordinate subband list, are maintained in the encoder and the decoder sides. The dominant subband list contains the indices of those subbands that have not become significant, and the subordinate subband list contains the indices of those

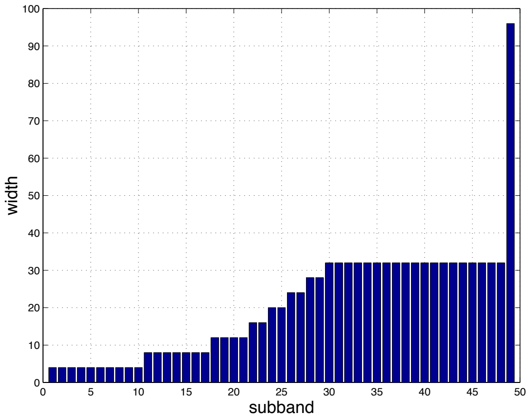


Figure 3. Subband width

subbands that have already become significant. The process that updates the subband dominant list is called the subband significant pass, and the process that updates the subband subordinate list is called the subband refinement pass.

Different coefficient magnitude thresholds are maintained in different subbands. Since we would like to deal with the most important subbands at first and get the best result with only a little information from the resource. Moreover, since sounds in different subbands have different sensibilities to human ears according to the psychoacoustic model, it is worthwhile to consider each subband independently instead of all subbands in one frame simultaneously.

We summarize the subband selection rule below.

1. MNR threshold calculation
   Determine empirically the MNR threshold value $T_{i,k}^{\text{MNR}}$ for channel i at layer k. Subbands with the smallest MNR value at the current layer are given the highest priority.

2. Subband dominant pass
   For those subbands that are still in the dominant subband list, if subband j in channel i has the current MNR value $\text{MNR}_{i,j}^{k} < T_{i,k}^{\text{MNR}}$, add subband j of channel i into the significant map, remove it from the dominant subband list, send 1 to the bit stream, indicating this subband is selected. Then, do coefficient SAQ for this subband. For subbands that have $\text{MNR}_{i,j}^{k} \geq T_{i,k}^{\text{MNR}}$, send 0 to the bit stream, indicating this subband is not selected at this layer.

3. Subband refinement pass
   For subband already in the subordinate list, do coefficient SAQ.

4. MNR values update
   Re-calculate and update MNR values for selected subbands.

5. Repeat Steps 1-4 until the bit stream meets the target rate.

5. IMPLEMENTATION ISSUES

5.1. Frame, subband or channel skipping

As mentioned earlier, each subband has its own initial coefficient magnitude threshold. This threshold has to be included in the bit stream as the overhead so that the decoder can start to reconstruct these coefficients once the layered information is available. In our implementation, the initial coefficient magnitude threshold $T_{i,j}(0)$ for channel i and subband j will be truncated to the nearest power of 2 that is no smaller than $C_{i,j}^{max}$, i.e.

$$T_{i,j}(0) = 2^{p_{i,j}},\ p_{i,j} = \lceil \log_2 C_{i,j}^{max} \rceil,$$

where $C_{i,j}^{max}$ is the maximum magnitude for all coefficients in channel i and subband j.

In order to save bits, the maximum power $p_i^{max} = max(p_{i,j}), \forall j$, for all subbands in channel i will be included in the bit stream at the first time when channel i is selected. A relative value of each subband's maximum power, i.e. the difference $\Delta p_{i,j} = p_i^{max} - p_{i,j}$ between p_i^{max} and $p_{i,j}$, will be included in the bit stream at the first time when the selected subband becomes significant.

For a frame with its maximum value $C_{i,j}^{max}$ equal to 0, i.e. $max(C_{i,j}^{max}) = 0, \forall j$, which means all coefficients in channel i in this frame have value 0, then a special indicator will be set to let the decoder know it should skip this frame. Similarly, if $C_{i,j}^{max}$ has value 0, another special indicator is set to tell the decoder that it should always skip this subband. In some cases when the end user is only interested in reconstructing some channels, channel skipping can also be adopted.

5.2. Determination of the MNR threshold

At each layer, the MNR threshold for each channel is determined empirically. Two basic rules are adopted when calculating this threshold.

1. The MNR threshold should allow a certain number of subbands to pass at each layer.
   Since the algorithm sends 0 to the bit stream for each un-selected subband which is still in the significant subband list, if the MNR threshold is so small that it allows too few subbands to pass, too many overhead bits will be generated. As a result, this will degrade the performance of the progressive audio codec.

2. Adopt a maximum MNR threshold.
   If the MNR threshold calculated by using the above rule is greater than a pre-defined maximum MNR threshold T_{max}^{MNR}, then the current MNR threshold for channel i at k^{th} layer $T_{i,k}^{\mathrm{MNR}}$ will be set to T_{max}^{MNR}. This is based on the assumption that a higher MNR value does not provide higher perceptual audio quality perceived by the human auditory system.

6. COMPLETE ALGORITHM DESCRIPTION

The block diagram of a complete encoder is shown in Figure 4. The perceptual model, filter bank, temporal noise shaping (TNS), and intensity blocks in our progressive encoder are borrowed from the AAC main profile encoder. The inter-channel redundancy removal procedure via KLT is implemented after the input audio signals are transformed into the MDCT domain. Then, a dynamic range control block follows to avoid any possible data overflow in later compression stages. The progressive quantization and lossless coding parts are finally used to construct the

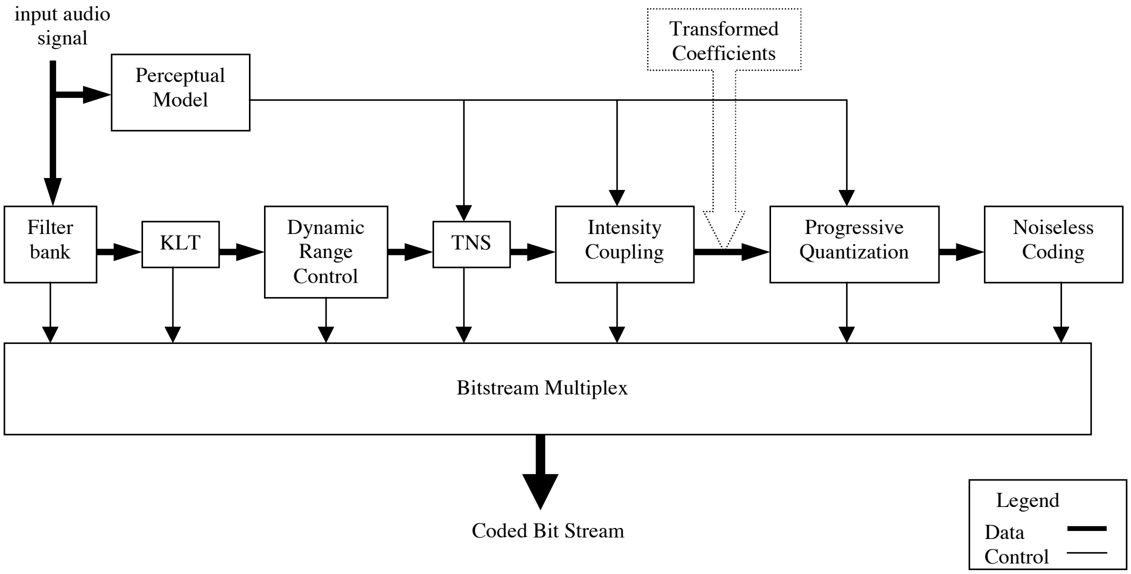


Figure 4. The block diagram of the proposed embedded encoder.

compressed bit stream. The information generated at the first several coding blocks will be sent into the bit stream as the overhead.

Figure 5 provides more details of the progressive quantization block. As mentioned in previous sections, the channel and the subband selection rules are used to determine which subband in which channel should be encoded at this point, and then coefficients within this selected subband will be quantized via SAQ. Finally, based on several different contexts, the layered information together with all overhead bits generated during previous coding blocks will be losslessly coded by using the context-based QM coder.

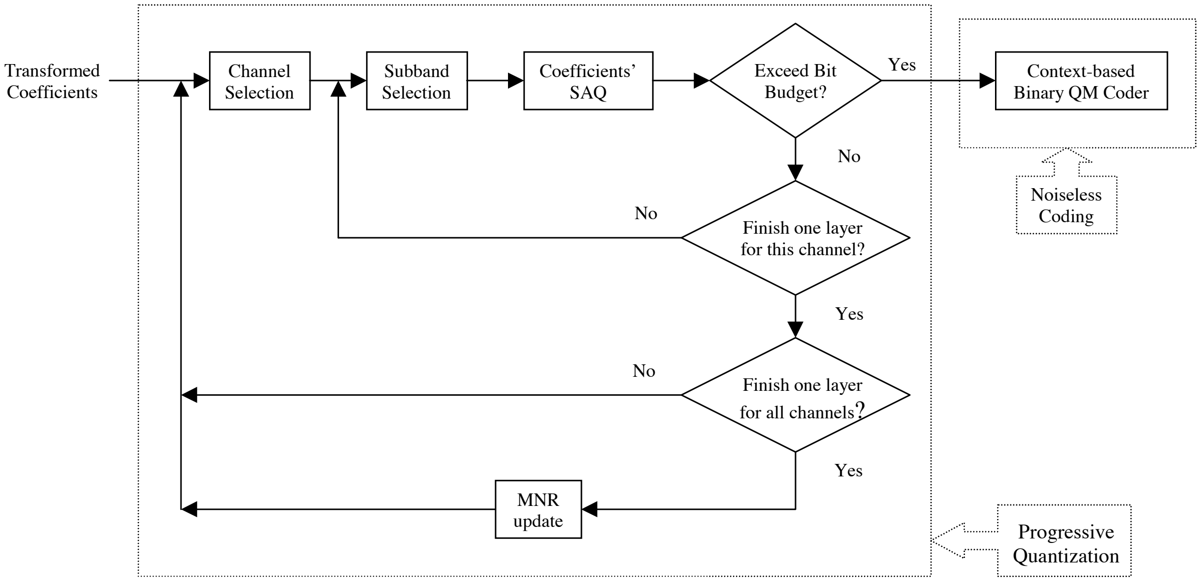


Figure 5. Illustration of the progressive quantization and lossless coding blocks.

The encoding process performed by using the proposed algorithm will stop when the bit budget is exhausted. It can cease at any time, and the resulting bit stream contains all lower rate coded bit streams. This is the so-called fully embedded property. It is worthwhile to mention that if the bit stream is truncated at an arbitrary point, there may be bits at the end of the code that do not provide a valid symbol since a codeword could be truncated. In that case, these bits do not reduce the width of an uncertainty interval or any distortion function. In fact, it is very likely that the first L bits of the bit stream will produce exactly the same reconstructed audio as the first $L + 1$ bits, which occurs when the additional bit is insufficient to complete the decoding of another symbol. Nevertheless, the capability to terminate the decoding of an embedded bit stream at any specific point is extremely useful in systems that are either rate-constrained or distortion-constrained.

7. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented and tested. The basic audio coding blocks,[9] including the psychoacoustic model, filter bank, temporal noise shaping, and intensity/coupling, of the MPEG AAC main profile encoder are still adopted. Furthermore, an inter-channel removal block, a progressive quantization block and a context-based QM coder block are added to construct the proposed progressive multichannel audio codec. Two test data sets are used in this experiment. One is a one-minute long audio material called "Messiah", which is a piece of classical music recorded live in a real concert hall. Another one is an eight-second long music called "Herre", which was used in MPEG-2 AAC standard (ISO/IEC 13818-7) conformance work. Both of them have the typical 5 channel configuration, including Center, Left, Right, Left surround and Right surround sounds.

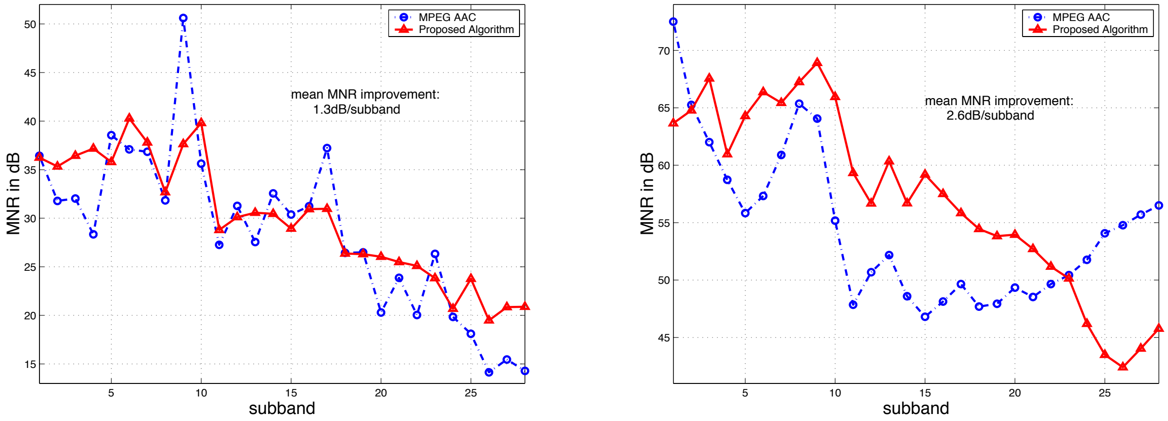
The performance comparison of MPEG AAC and the proposed embedded multichannel audio codec are shown in Figures 6 and 7. The mean MNR values and the mean MNR improvement shown in Figure 6 are calculated by

$$\text{mean MNR}_{subband} = \frac{\sum_{channel} \text{MNR}_{channel,subband}}{\text{number of channels}},$$

$$\text{mean MNR improvement} \;=\; \frac{\sum_{subband}(\text{mean MNR}_{subband}^{\text{Proposed Algorithm}} - \text{mean MNR}_{subband}^{\text{MPEG AAC}})}{\text{number of subband}}.$$

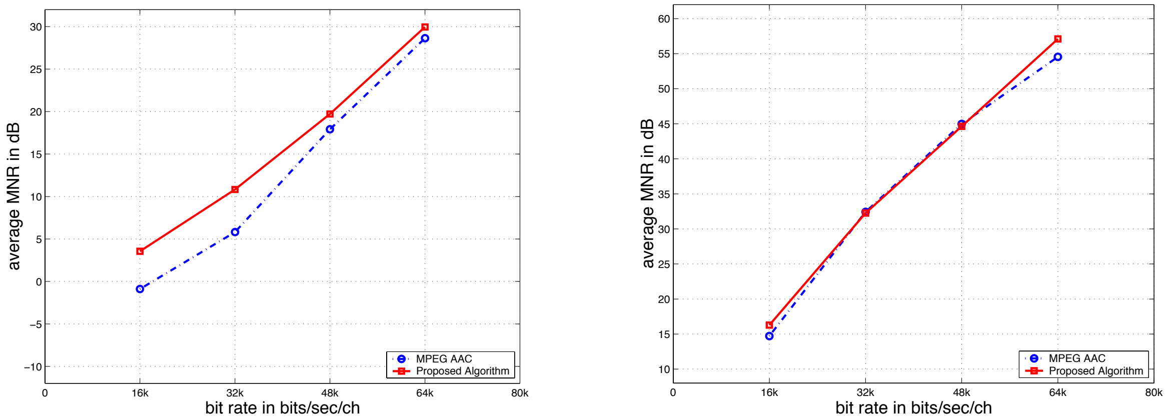
The average MNR shown in Figure 7 is calculated by

$$\text{average MNR} = \frac{\sum_{subband} \text{mean MNR}_{subband}}{\text{number of subband}}.$$



(a) (b)

Figure 6. The MNR comparison at the bit rate of 64 kbits/sec/ch for 5-channel test audio: (a) "Herre" and (b) "Messiah".



(a) (b)

Figure 7. The MNR comparison at different bit rates for 5-channel test audio: (a) "Herre" and (b) "Messiah".

Figures 6 (a) and (b) show the mean MNR comparison between the proposed algorithm and MPEG AAC for "Herre" and "Messiah" at a typical bit rate of 64 kbits/sec/ch. These figures clearly indicate that the mean MNR value calculated from the reconstructed audio by using our proposed algorithm is better than that of MPEG AAC for most subbands and the mean MNR improvement is more than 1dB per subband for both test audio materials.

Figures 7 (a) and (b) show the average MNR comparison between the proposed algorithm and MPEG AAC for "Herre" and "Messiah" at several different bit rates varying from 16 kbits/sec/ch to 64 kbits/sec/ch. Figure 7 (a) indicates that the proposed algorithm achieves better average MNR values at all test bit rates for test audio "Herre". Figure 7 (b) shows that the proposed algorithm achieves better average MNR values at the bit rate of 16 kbits/sec/ch and 64 kbits/sec/ch, and similar average MNR values at the bit rate of 32 kbits/sec/ch and 48 kbits/sec/ch for test audio "Messiah". Besides, the subjective listening test indicates that, the reconstructed audio generated from the proposed embedded codec has much lower perceptual noise in comparison with reconstructed audio generated from MPEG AAC for both test materials, especially at lower bit rates.

In addition, the bit stream generated by MPEG AAC only achieves an approximate bit rate and is normally a little bit higher than the desired one while our algorithm achieves a much more accurate bit rate in all experiments carried out in this section.

8. CONCLUSION

A fully embedded high-quality multichannel audio coding algorithm was presented in this research. This algorithm utilizes KLT in the pre-processing stage to remove inter-channel redundancy inherent in the original multichannel audio source. Then, rules for channel selection and subband selection were developed and the SAQ process was used to determine the importance of coefficients and their layered information. At the last stage, all information is losslessly compressed by using the context-based QM coder to generate the final multichannel audio bit stream. The distinct advantage of the proposed algorithm over most existing multichannel audio codecs is that it can achieve a precise rate control. That is, audio signals can be coded to meet the desired bit rate. It was shown in our experiments that the proposed algorithm not only achieves the embedded capability, but also outperforms MPEG AAC in both the objective measurement and the subjective listening test at various bit rates. Moreover, the advantage of the embedded multichannel audio codec is more obvious at lower bit rates.

ACKNOWLEDGMENT

This research has been partially funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center with the Cooperative Agreement Number EEC-9529152.

REFERENCES

1. Marina Bosi, "High-Quality Multichannel Audio Coding: Trends and Challenges", Journal of the Audio Engineering Society, Vol. 48, Number 6, June 2000
2. D. Yang, H. Ai, C. Kyriakakis, C.-C. Kuo, "An Inter-channel Redundancy Removal Approach for High-quality Multichannel Audio Compression", presented at the AES 109th convention, Los Angeles, September 2000.
3. D. Yang, H. Ai, C. Kyriakakis, C.-C. Kuo, "An Explorition of Karhunen-Loeve Transform for Multichannel Audio Coding", to appear at Conference of Electronic Cinema, SPIE's International Symposium on Information Technologies, Boston, MA, November 5-8, 2000.
4. J. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", IEEE transaction on signal processing, Vol. 41, No. 12, December 1993.
5. C.C. Jay Kuo et al, "Multithreshold wavelet codec (MTWC)", Doc. N. WG1N665, Novemeber 1997.
6. W. Pennebaker, J. Mitchell, "JPEG Still Image Data Compression Standard".
7. J. Herre, E. Eberlein, H. Schott, and K. Brandenburg, "Advanced Audio Measurement System using Psychoacoustic Properties", AES preprint 3321, prosented at the 92th convention, Vienna, March 1992.
8. K. Brandenburg, "Evaluation of Quality for Audio Encoding at Low Bit Rates", AES preprint 2433, presented at the 82nd convention, London, 1987.
9. ISO/IEC JTC1/SC29/WG11 N2262, "ISO/IEC TR 13818-5, Software Simulation".
10. K. Brandenburg, M. Bosi, "ISO/IEC MPEG-2 Advanced Audio Coding: Overview and Applications", AES preprint 4641, presented at the 103 rd convention, New York, September 1997.
11. M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson and Y. Oikawa, "ISO/IEC MPEG-2 Advanced Audio Coding", AES preprint 4382, presented at the 101 st convention, Los Angeles, November 1996,.
12. ISO/IEC JTC1/SC29/WG11 N1650, "IS 13818-7 (MPEG-2 Advanced Audio Coding, AAC)".
13. Houngjyh Wang and C.-C. Jay Kuo, "High Fidelity Image Compression with Multithreshold Wavelet Coding (MTWC)", Conference on "Application of Digital Image Processing XX", SPIE's Annual Meeting, San Diego, CA, July 27-August 1, 1997.