



Audio Engineering Society Convention Paper 5868

Presented at the 115th Convention
2003 October 10–13 New York, NY, USA

This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Scalable Perceptual and Lossless Audio Coding based on MPEG-4 AAC

Ralf Geiger¹, Gerald Schuller¹, Jürgen Herre², Ralph Sperschneider², Thomas Sporer¹

¹*Fraunhofer IIS AEMT, Ilmenau, Germany*

²*Fraunhofer IIS, Erlangen, Germany*

Correspondence should be addressed to Ralf Geiger (ggr@emt.iis.fhg.de)

ABSTRACT

This paper presents a scalable lossless enhancement of MPEG-4 Advanced Audio Coding (AAC). Scalability is achieved in the frequency domain using the Integer Modified Discrete Cosine Transform (IntMDCT), which is an integer approximation of the MDCT providing perfect reconstruction. With this transform, and only minor extension of the bitstream syntax, the MPEG-4 AAC Scalable codec can be extended to a lossless operation. The system provides bit-exact reconstruction of the input signal independent of the implementation accuracy of the AAC core coder. Furthermore, scalability in sampling rate and reconstruction word length is supported.

1. INTRODUCTION

Traditionally, perceptual and lossless audio coding have been separate worlds. While lossless audio cod-

ing schemes are usually based on prediction, modern perceptual audio coding schemes use subband coding with filter banks, such as the Modified Discrete

Cosine Transform (MDCT) [1] to obtain a blockwise representation of the audio signal in the frequency domain. This paper presents a scalable lossless enhancement of the MPEG-4 AAC perceptual codec [2].

The goal of this codec is to use MPEG-4 AAC as a lossy core coder in such a way that the total bit rate is below the simulcast of a lossy and a lossless coder. The problem is that the MPEG-4 AAC decoder is specified in the float domain, and different architectures may easily lead to slightly different results, and hence to different rounded decoded audio output signals. If the lossless enhancement layer is constructed in the time domain, this means that the entire lossy decoder needs to be specified in such a way that it results in a bit-exact reconstruction. On the other hand, if the lossless enhancement layer is constructed in the frequency domain, only the MDCT and tools connected to it need to be specified such that they become bit-exact. This is because the lossless enhancement in the decoder “bypasses” most of the lossy decoder.

2. ENABLING TECHNOLOGY: INTMDCT

The IntMDCT [3], [4], is an integer approximation of the MDCT maintaining perfect reconstruction. The Lifting scheme [5] or ladder network [6] allows to approximate Givens rotations, which are used to implement the MDCT ([3], [4]), by mapping integers to integers in a reversible way. This process achieves a close approximation of the MDCT by the IntMDCT.

In detail, the basic principle is the following: Every Givens Rotation is decomposed into three so-called lifting steps:

$$\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} = \begin{pmatrix} 1 & \frac{\cos \alpha - 1}{\sin \alpha} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin \alpha & 1 \end{pmatrix} \begin{pmatrix} 1 & \frac{\cos \alpha - 1}{\sin \alpha} \\ 0 & 1 \end{pmatrix}$$

In every lifting step a rounding function can be included to stay in the integer domain. This rounding does not affect the perfect reconstruction property, because every lifting step can be inverted by subtracting the value that has been added.

This approach can also be generalized to a multi-dimensional lifting scheme. The basic building

blocks are block matrices of the form

$$\begin{pmatrix} E_n & 0 \\ A & E_n \end{pmatrix}$$

with an $n \times n$ matrix A and the $n \times n$ unit matrix E_n . After applying the matrix A , n rounding operations are applied to stay in the integer domain. This multi-dimensional lifting step can be inverted by the block matrix

$$\begin{pmatrix} E_n & 0 \\ -A & E_n \end{pmatrix}$$

The MDCT can be partially decomposed into these multi-dimensional lifting steps. This allows to reduce the number of rounding operations necessary for the invertible integer approximation, compared to the previous approach of completely decomposing the MDCT into Givens rotations [3]. A similar approach of multi-dimensional lifting was utilized to derive an integer DCT in [7].

3. GENERAL STRUCTURE OF THE SCALABLE SYSTEM

Based on the perfect reconstruction property and the close approximation of the MDCT, the IntMDCT allows to build a scalable lossless enhancement of MDCT-based perceptual audio coding schemes, as shown in [4]. Figure 1 shows the block structure of this codec.

Encoder: The structure of the encoder is an extension of the well-known general structure of a perceptual audio coding scheme. In addition to the usual MDCT output, the IntMDCT output is calculated. For the lossless enhancement, the difference between the IntMDCT output and the inverse quantized MDCT output is calculated. These difference values are entropy coded and transmitted in the lossless enhancement bitstream.

Decoder: To achieve lossless decoding, the difference values transmitted in the lossless enhancement bitstream are added to the usual MDCT values. In this way the IntMDCT spectral values are reconstructed exactly, and hence the bit-exact audio signal can be obtained.

4. BIT-EXACT RECONSTRUCTION OF LOSSLESS DECODED SIGNAL

When using a scalable coder based on coding of a time domain residual signal, the reconstruction pre-

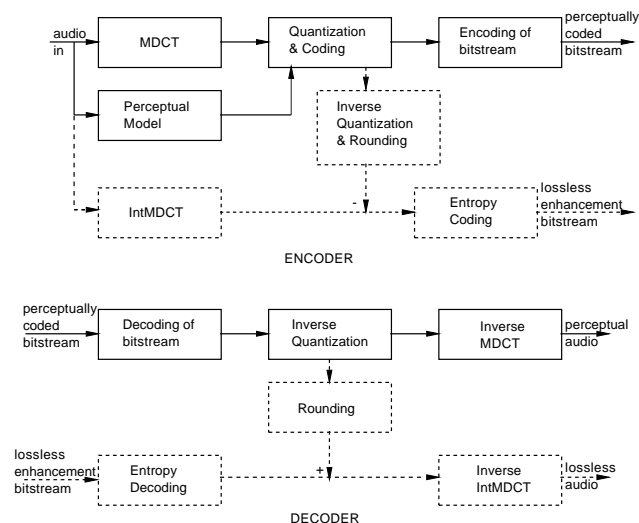


Fig. 1: Perceptual audio coding scheme (solid lines) and scalable lossless enhancement (dashed lines)

cision of the lossy base layer coder becomes critical for achieving an overall lossless reconstruction. Specifically, an implementation with a very high numeric precision (which might exclude cost-effective implementations) is necessary to approximate the “exact” output of the decoding process, which is by definition a floating-point process. Even with a very high numerical precision in the base layer decoder, however, an occasional bit-difference between different decoder implementations cannot be avoided by definition. In order to design a truly lossless system, the IntMDCT-based solution circumvents this problem in the following way: It accepts the transmitted base layer bitstream (which contains quantized numbers, i.e. integers) and stays in the integer domain for further calculations up to the reconstruction of the time signal by means of an inverse IntMDCT. Consequently, a lossless reconstruction can be achieved without having to rely on floating-point precision.

5. SCALABLE SYSTEM BASED ON AAC

The coding scheme presented in this paper contains several AAC-specific refinements. Figures 2 and 3 show the encoder and decoder block structure of this codec. To get an efficient lossless extension of AAC, especially the coding tools Mid/Side-Coding (M/S) and Temporal Noise Shaping (TNS) are considered and implemented in a lossless integer fashion.

The codec is designed to build upon the MPEG-4 AAC Scalable codec [2] and re-uses most of the tools and the structure of this codec. Only a minor extension of the bitstream syntax is needed to allow for this lossless extension.

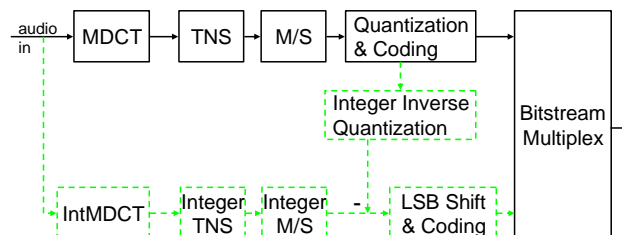


Fig. 2: Encoder for scalable lossless enhancement of AAC

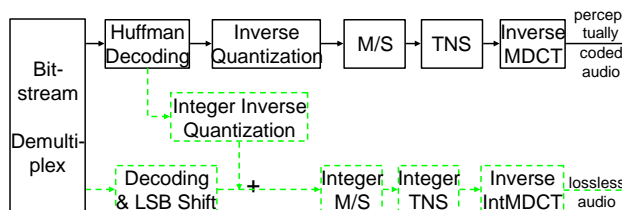


Fig. 3: Decoder for scalable lossless enhancement of AAC

The important blocks of this codec are described in detail subsequently:

5.1. Inverse Quantization to integer values

One crucial point for the lossless operation of the system is the inverse quantization and rounding process in the decoder. According to [2] the inverse quantized and rescaled values depend on the quantized value x_{quant} and the scale factor sf , and they are calculated by:

$$x_{invquant} = \text{sign}(x_{quant}) * |x_{quant}|^{4/3}$$

$$\text{gain} = 2^{0.25*(sf - SF_OFFSET)}$$

$$x_{rescal} = x_{invquant} * \text{gain}$$

These values x_{rescal} are rounded to integer to build the difference values for the lossless enhancement layer. To make these operations reliable and independent of the specific implementation, the codec uses fixed integer look-up tables. According to [2] the maximum allowed absolute amplitude for x_{quant}

is 8191. Thus for one scalefactor an integer table with 8192 values is needed. Such a table is, however, not needed for all possible scale factor values. When increasing the scale factor by 4 the corresponding gain is scaled by a factor of 2. Thus tables are needed for only four specific scale factor values, i.e. sf_0 , $sf_0 + 1$, $sf_0 + 2$, $sf_0 + 3$. For all other scale factor values, the corresponding gain is a multiple of an integer power of 2 of the gain of one of these four scale factors. So with these four tables all the necessary rounded rescaled inverse quantized integer values can be derived by applying bit shift operations.

5.2. Encoding of lossless extension layer values with LSB shift

The coding of the extension layer values can be achieved by completely reusing the bitstream syntax of the AAC Scalable codec [2]. Only one additional element is introduced to enable efficient coding of large spectral values. To achieve an appropriate compression for the integer difference values of the lossless enhancement layer, an entropy coding scheme has to be applied. For this task the noiseless coding tool of AAC can be reused with only a small modification. Compared to the coarsely quantized spectral values of AAC, the integer values for the lossless enhancement layer comprise a much larger range. To apply the AAC noiseless coding tool appropriately, the absolute values of the integer difference values are divided into LSB and MSB values by applying a certain number of bit shift operations. The MSB values are coded using the AAC noiseless coding tool, and the LSB values are coded as PCM values according to the number of bit shifts. The number of bit shifts can be chosen for each scale factor band individually. To encode this value for each scale factor band, the scale factor coding mechanism of AAC is reused. In the decoder the integer difference values can be reconstructed based on the MSB and LSB values. This technique of dividing the integer spectral values into LSB and MSB values for each scale factor band individually and coding the MSB values with the AAC noiseless coding tool allows for a flexible adaptation to the statistical characteristics of the integer values and therefore achieves an efficient entropy coding without the need for additional larger Huffman codebooks. For the bitstream syntax of the lossless enhancement layer, the bitstream

syntax of AAC can be reused by amending the PCM coding of the LSB values.

5.3. Compliance with AAC coding tools

The lossless enhancement is compliant with the AAC coding tools Block Switching and Window Shape Adaptation, TNS, MS and PNS. Thus, no restrictions apply to the usage of these tools in the lossy core. Furthermore, they can even be applied advantageously to the lossless enhancement layer. In detail, the lossless enhancement deals with these tools in the following way:

5.3.1. Block Switching and Window Shape

The IntMDCT in the lossless enhancement layer can use the same window shape and window sequence as the AAC coder, so that the lossless enhancement can simply follow the block switching and window shape decisions of the lossy core coder.

5.3.2. Temporal Noise Shaping (TNS)

The TNS tool in AAC modifies the MDCT spectrum by applying linear prediction filters before applying quantization. Consequently, the difference between the IntMDCT values of the lossless enhancement layer and the quantized MDCT values of the AAC core layer would increase. The TNS tool can, however, also be applied to the IntMDCT values in a lossless way by using the same prediction filter and including a rounding to integer values after each prediction step. In the decoder, the original IntMDCT spectrum is reconstructed by using the inverse filter and the same rounding. This lossless version of the TNS filter works as a closed loop prediction and provides a redundancy reduction for transient signals, as stated in [8].

5.3.3. Mid/Side Coding (M/S)

The M/S coding tool in AAC modifies some scale factor bands of the MDCT spectrum by calculating the sum and difference of left and right channel spectral values. The lossless enhancement layer can follow the M/S decisions of the lossy AAC core coder and apply M/S coding in a lossless way. This is done by applying a rounded Givens rotation with an angle of $\pi/4$, based on the lifting scheme. Thereby the energy is preserved and the original IntMDCT values can be reconstructed in the decoder. The alternative of applying sum and difference to the IntMDCT values would be lossless, too, but it would increase the energy by a factor of 2, and hence increase the

bit-rate.

5.3.4. Perceptual Noise Shaping (PNS)

The lossless enhancement is compliant with the PNS tool. This is done in the same way as for the AAC Scalable codec: In the scale factor bands where PNS is switched on, the inverse quantized spectral values are assumed as zero for calculating the difference values for the enhancement layer.

6. LOSSLESS-ONLY MODE

The system can also operate in a lossless-only mode. In this mode no AAC core layer is used and the IntMDCT spectral values are coded within one layer. In this mode the bitstream is identical with an AAC bitstream extended by the additional LSB coding described above. In the lossless-only mode the coding tools Block Switching and Window Shape Adaption, TNS and M/S can be used in a lossless way for the purpose of further redundancy reduction.

7. COMPRESSION RESULTS

The compression performance was evaluated based on the audio material used for the lossless coding activities of the ISO MPEG group [9]. The audio material consists of recordings of the New York Symphonic Ensemble and Jazz recordings. Both types of music were originally recorded at 96 kHz / 24 bit or 192 kHz / 24 bit. Table 1 summarizes the compression results in bits per sample for the AAC-based lossless enhancement, the lossless-only mode, and, as a comparison, the prediction-based lossless coder Monkey's Audio [10].

The AAC codec is operating at 64 kbps/channel for 48 kHz, 80 kbps/channel for 96 kHz, and 96 kbps/channel for 192 kHz.

It can be observed that the enhancement can clearly benefit from the AAC core, since its bit rate is lower than for the lossless-only mode. Hence the bit demand in the AAC-based mode is only slightly higher than in the lossless-only mode. It can also be observed that the compression performance in the lossless-only mode is only slightly worse than for the Monkey's Audio codec. Comparing the bit demand for the lossless enhancement with the bit demand for lossless coding it can be observed that the scalable solution clearly outperforms a simulcast solution, i.e. a simultaneous transmission of an AAC bitstream and a lossless-only coded bitstream.

8. SAMPLING RATE AND WORD LENGTH SCALABILITY

The codec provides additional scalability in sampling rate and word length by dividing the integer difference values of the enhancement layer into several layers. The concept of sampling rate and word length scalability was originally introduced in [11], and a time domain approach was presented. Here, in the context of frequency domain lossless audio coding, the same functionality can be achieved. Figures 4 and 5 show the block structure of the extended encoder and decoder.

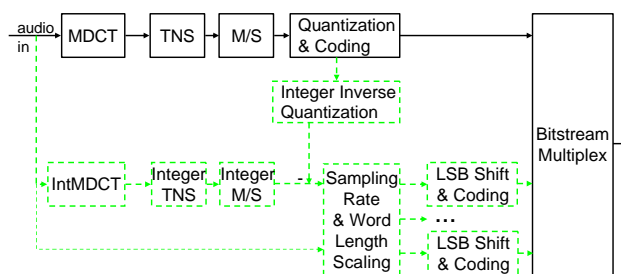


Fig. 4: Encoder for scalable system based on AAC with additional sampling rate and word length scalability

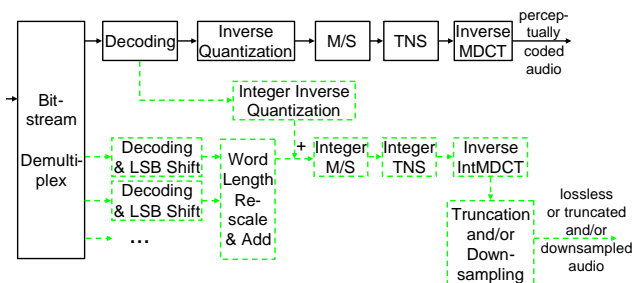


Fig. 5: Decoder for scalable system based on AAC with additional sampling rate and word length scalability

Both directions of format scalability follow the same principle: The scalability is obtained by dividing the integer difference values of the enhancement layer into several layers. In the first enhancement layer, the spectral values represent the IntMDCT values of a lowpass filtered and/or truncated version of the input signal. Further enhancement layers can increase the bandwidth and/or the word length of the repre-

	48 kHz 16 bit	48 kHz 24 bit	96 kHz 24 bit	192 kHz 24 bit
AAC	1.3	1.3	0.8	0.5
Enhancement	6.5	14.4	11.0	9.2
AAC + Enhancement	7.8	15.7	11.8	9.7
Lossless-only	7.5	15.3	11.6	9.5
Monkey's Audio 3.97	7.2	15.2	11.5	9.4
Simulcast (AAC + Monkey's Audio)	8.5	16.5	12.3	9.9

Table 1: Compression results (in bits per sample) for AAC-based lossless enhancement, lossless-only mode, Monkey's Audio, and a simulcast solution

sented signal and finally the full bandwidth and word length is coded in the last layer. The encoder has the flexibility to create the integer spectral values for the intermediate quality layers either from the integer spectral values of the high quality input signal or from the input signal itself. More advanced ways of calculating the intermediate spectral values might be considered in the encoder. One possible way is to calculate the desired intermediate quality signal in time domain and apply an additional IntMDCT to this signal. This increases the complexity of the encoder, but it allows to encode exactly the desired intermediate signal. On the other hand the overall compression ratio achievable with this intermediate signal might not be as high. Thus the system allows a flexible trade-off between overall compression ratio and quality of the intermediate signal. The next enhancement layer simply encodes the difference values between the desired spectral values and spectral values of the layer it builds upon. All the enhancement layers can be coded in the same way as the lossless enhancement layer in the two layer lossy-lossless mode described above. In the decoder all the transmitted enhancement layers are decoded and the integer spectral values are added appropriately. The same inverse IntMDCT is applied for all possible intermediate word lengths and sampling rates. The appropriate truncation and downsampling is applied after the inverse IntMDCT.

In more detail, the different directions of scalability have the following principles:

8.1. Sampling Rate Scalability

The sampling rate scalability is obtained by applying the desired lowpass filter in the encoder, encoding

the filtered signal and applying the downsampling after the inverse transform in the decoder. In this way the process of downsampling is not completely done in the encoder, but it is split between encoder and decoder. This allows for very efficient coding of the lowpass signal and the difference for the enhancement layer with higher bandwidth. A possible way of lowpass filtering is to take the IntMDCT spectrum of the full bandwidth signal and set the higher frequency components to zero. For example, the first enhancement layer encodes the lower half of the spectrum and the second enhancement layer encodes the upper half. The system is, however, not restricted to this way of lowpass filtering. In principle, every lowpass filtered version of the input signal can be used in the encoder by calculating the IntMDCT values of this lowpass filtered signal for the first enhancement layer. The last enhancement simply has to code the difference values to transmit the full spectral information for the original input signal.

8.2. Word Length Scalability

Word length scalability is obtained by dividing every integer difference value into an LSB and an MSB part. For example, the MSB part represents the signal with 16 Bit accuracy, the LSB part represents the difference values for 24 Bit accuracy. In the first enhancement layer only the MSB values are coded, in the next enhancement layer the remaining LSB values are coded additionally. A possible way of calculating the LSB and MSB values is to put the lower bits into the LSB part and the higher bits into the MSB part. The system is, however, not restricted to this way of constructing the intermediate signal with lower accuracy. In principle, every truncated ver-

sion of the input signal can be used to calculate the spectral values necessary for the enhancement layer by calculating the IntMDCT values of the truncated signal. For example, an appropriate dithering could be applied for the intermediate signal. The last enhancement layer simply has to code the difference values to transmit the full spectral information for the original input signal. For decoding the signal based on a layer representing the lower accuracy, the values are scaled to compensate for the missing LSB values (e.g. by 2^8 for 8 LSB Bits) and the inverse IntMDCT is applied. The resulting PCM signal with lower accuracy is then scaled down again to compensate for the previous upscaling. This scaling before and after the inverse IntMDCT is necessary to avoid the additional noise floor introduced by applying the inverse IntMDCT to the modified spectral values.

9. APPLICATION SCENARIOS

The scalable perceptual and lossless codec is expected to be useful in the following scenarios:

9.1. Production

While the need for lossless recording is obvious, during a production there is often the need to include remote sites to evaluate the new material. A standardize format allows all studios to take part. A lossy core is helpful to allow, for example, the remote monitoring of a recording session in realtime over cheaper lower capacity networks.

9.2. Streaming

It is desirable to have a system which maintains compatibility in terms of transmission characteristics to lossy coding schemes, like frame rates or error characteristics in case of packet loss and which can be scaled to the instantaneous network data rate up to full lossless transmission without the need for recoding. This will allow scenarios like pre-listening to a low bit rate lossy version and the upgrade to a higher quality versions after purchasing the item.

9.3. Archiving

For the application of combined archiving and transmission it is desirable to store the original audio signal in a lossless representation. Low bit rate versions can be extracted anytime to allow for e.g. remote data bank browsing or extraction of quality-restricted trial versions

It is desirable that all applications described above can be realized with a single coding architecture, al-

lowing lossy, lossy-lossless and lossless operation. A common frame rate/structure allows the same editing tools to be used.

10. CONCLUSIONS

In this paper a scalable lossless enhancement of MPEG-4 AAC was presented. The enhancement is performed in frequency domain using an IntMDCT approach. This allows for a robust bit-exact reconstruction and a lossless decoding on different platforms. The system can also be used as a stand-alone lossless codec by simply omitting the perceptual codec. Further options for scalability in sampling rate and word length are provided. The codec allows for a good compression performance both in the scalable mode and in the lossless-only mode.

11. REFERENCES

- [1] J. Princen and A. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation," *IEEE Trans. ASSP*, vol. ASSP-34, no. 5, pp. 1153–1161, 1986.
- [2] "Information technology - Coding of audio-visual objects - Part 3: Audio," International Standard 14496-3:2001, ISO/IEC JTC1/SC29/WG11 Moving Pictures Expert Group, 2001.
- [3] R. Geiger, T. Sporer, J. Koller, and K. Brandenburg, "Audio Coding based on Integer Transforms," in *111th AES Convention*, New York, 2001.
- [4] R. Geiger, J. Herre, J. Koller, and K. Brandenburg, "IntMDCT - A link between perceptual and lossless audio coding," in *Proc. ICASSP 2002*, Orlando, 2002.
- [5] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," Tech. Rep., Bell Laboratories, Lucent Technologies, 1996.
- [6] F. Bruekers and A. Enden, "New networks for perfect inversion and perfect reconstruction," *IEEE JSAC*, vol. 10, no. 1, pp. 130–137, Jan. 1992.

- [7] J. Wang, J. Sun, and S. Yu, "1-d and 2-d transforms from integers to integers," in *Proc. ICASSP'03*, Hong Kong, April 2003.
- [8] J. Herre and J. D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)," in *101st AES Convention*, Los Angeles, 1996, preprint 4384.
- [9] ISO/IEC JTC1/SC29/WG11 Moving Pictures Expert Group, "Final Call for Proposals on MPEG-4 Lossless Audio Coding," , no. N5208, October 2002.
- [10] M. T. Ashland, "Monkey's Audio - a fast and powerful lossless audio compressor," <http://www.monkeysaudio.com>.
- [11] T. Moriya, A. Jin, T. Mori, K. Ikeda, and T. Kaneko, "Hierarchical lossless audio coding in terms of sampling rate and amplitude resolution," in *Proc. ICASSP*, 2003.