AN INTRODUCTION TO MPEG-4 AUDIO LOSSLESS CODING

Tilman Liebchen

Technical University of Berlin

ABSTRACT

Lossless coding will become the latest extension of the MPEG-4 audio standard. In response to a call for proposals, many companies have submitted lossless audio codecs for evaluation. The codec of the Technical University of Berlin was chosen as reference model for *MPEG-4 Audio Lossless Coding (ALS)*, attaining working draft status in July 2003. The encoder is based on linear prediction, which enables high compression even with moderate complexity, while the corresponding decoder is straightforward. The paper describes the basic elements of the codec, points out envisaged applications, and gives an outline of the standardization process.

1. INTRODUCTION

Lossless audio coding enables the compression of digital audio data without any loss in quality due to a perfect reconstruction of the original signal. The MPEG audio subgroup is currently working on the standardization of lossless coding techniques for high-definition audio signals. As an extension to MPEG-4 Audio [1], the amendment "ISO/IEC 14496-3:2001/AMD 4" will define methods for lossless coding of signals with resolutions up to 24 bits and sampling rates up to 192 kHz.

In July 2002, MPEG issued a call for proposals [2] to initiate the submission of technology for lossless audio coding. This call basically supported two different approaches, either a hierarchical system consisting of a lossy core codec (e.g. MPEG-AAC [3]) and a lossless enhancement layer, or a lossless-only codec. By December 2002, seven companies submitted one or more codecs which met the basic requirements. In the following, those submissions were evaluated in terms of compression efficiency, complexity and flexibility [4]. In March 2003, the audio subgroup decided to proceed at first with the standardization of a lossless-only codec, while further investigating hierarchical methods as well.

The lossless-only codec of the Technical University of Berlin (TUB), which offered the highest compression among all submissions, was chosen as reference model [5]. In July 2003, the codec attained working draft status. In October 2003, an alternative entropy coding scheme, proposed by RealNetworks, was added to enable even better compression [6].

Below, we describe the basics of MPEG-4 Audio Lossless Coding [7], give some compression results, point out applications and outline the standardization process.

2. MPEG-4 AUDIO LOSSLESS CODING

In most *lossy* MPEG coding standards, only the decoder is specified in detail. However, a *lossless* coding scheme usually requires the specification of some (but not all) encoder portions. Since the encoding process has to be perfectly reversible without loss of information, several parts of both encoder and decoder have to be implemented in a deterministic way.

The MPEG-4 ALS codec uses forward-adaptive *Linear Predictive Coding (LPC)* to reduce bit rates compared to PCM, leaving the optimization entirely to the encoder. Thus, various encoder implementations are possible, offering a certain range in terms of efficiency and complexity. This section gives an overview of the basic encoder and decoder functionality.

2.1. Encoder Overview

The MPEG-4 ALS encoder (Figure 1) typically consists of these main building blocks:

- *Buffer*: Stores one audio frame. A frame is divided into blocks of samples, typically one for each channel.
- *Coefficients Estimation and Quantization*: Estimates (and quantizes) the optimum predictor coefficients for each block.
- *Predictor*: Calculates the prediction residual using the quantized predictor coefficients.
- *Entropy Coding*: Encodes the residual using different entropy codes.
- *Multiplexing*: Combines coded residual, code indices and predictor coefficients to form the compressed bitstream.

For each channel, a prediction residual is calculated using linear prediction with adaptive predictor coefficients and (preferably) adaptive prediction order in each block. The coefficients are quantized prior to filtering and transmitted as side information. The prediction residual is entropy coded using one of several different entropy codes. The indices of the chosen codes have to be transmitted. Finally, a multiplexing unit combines coded residual, code indices, predictor coefficients and other additional information to form the compressed bitstream. The encoder also provides a CRC checksum, which is supplied mainly for the decoder to verify the decoded data. On the encoder side, the CRC can be used to ensure that the compressed data is losslessly decodable.

Additional encoder options comprise block length switching, random access and joint stereo coding (see section 3). The encoder might use these options to offer several compression levels with differing complexities. However, the differences in terms of coding efficiency usually are rather small, so it may be appropriate to abstain from the highest compression in order to reduce the computational effort. Coding results for a variety of audio material will be given in section 4.

2.2. Decoder Overview

The MPEG-4 ALS decoder (Figure 2) is significantly less complex than the encoder. It decodes the entropy coded residual and,

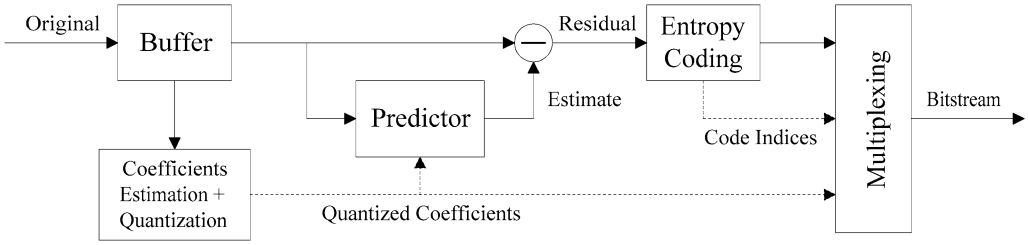

Fig. 1. MPEG-4 ALS encoder

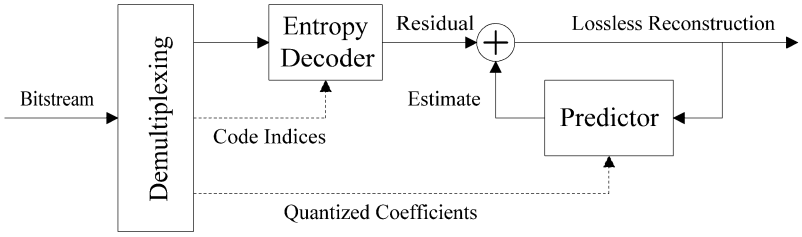

Fig. 2. MPEG-4 ALS decoder

using the predictor coefficients, calculates the lossless reconstruction signal. The computational effort of the decoder mainly depends on the order of the predictor chosen by the encoder. Since the maximum order usually depends on the encoder's compression level, higher compressed files might take slightly longer to decode. Apart from the predictor order, the decoder complexity is nearly independent from the encoder options.

3. ENCODER DESCRIPTION

3.1. Linear Prediction

The current sample of a time-discrete signal $x(n)$ can be approximately predicted from previous samples $x(n-k)$. The estimate is given by

$$\hat{x}(n) = \sum_{k=1}^{K} h_k \cdot x(n-k), \tag{1}$$

where K is the order of the predictor. If the predicted samples are close to the original samples, the residual

$$e(n) = x(n) - \hat{x}(n) \tag{2}$$

has a smaller variance than $x(n)$ itself, hence $e(n)$ can be encoded more efficiently.

In forward linear prediction, the optimal predictor coefficients h_k (in terms of a minimized variance of the residual) are usually estimated for each block by the autocorrelation method or the covariance method [8]. The autocorrelation method, using the Levinson-Durbin algorithm, has additionally the advantage of providing a simple means to iteratively adapt the *order* of the predictor [9].

Increasing the predictor order decreases the variance of the prediction error, leading to a smaller bit rate for the residual. On the other hand, the bit rate for the predictor coefficients will rise with the number of coefficients to be transmitted. Thus, the task is to find the optimal order which minimizes the total bit rate.

The Levinson-Durbin algorithm determines recursively all predictors with increasing order. For each order, a complete set of predictor coefficients is calculated. Moreover, the variance σ_e^2 of the corresponding residual can be calculated, resulting in an estimate of the expected bit rate for the residual. Together with the bit rate for the coefficients, the total bit rate can be determined in each iteration, i.e. for each predictor order. The optimal order is set at the point where the total bit rate no longer decreases.

3.2. Quantization of Predictor Coefficients

Direct quantization of the predictor coefficients h_k is not very efficient for transmission, since even small quantization errors might produce large spectral errors. Although parcor (reflection) coefficients are less sensitive to quantization, they are still too sensitive when their magnitude is close to unity. In order to expand the region near unity, an arcsine function is applied to the parcor coefficients. The behavior of the resulting *arcsine coefficients* is very similar to the more familiar log-area ratio (LAR) coefficients.

For a predictor filter of order K, a set of parcor coefficients $\gamma_k, k = 1 \ldots K$, can be estimated using the Levinson-Durbin recursion. Those coefficients are converted to arcsine coefficients using

$$\alpha_k = \arcsin(\gamma_k). \tag{3}$$

The value of each α_k is restricted to $[-\pi/2, +\pi/2]$. A linear 8-bit quantization is applied to the arcsine coefficients, which is equivalent to a non-linear quantization of the corresponding parcor coefficients. Only the 8-bit indices of the quantized arcsine coefficients are finally transmitted.

However, the direct form predictor filter uses predictor coefficients h_k according to (1). In order to employ identical coefficients in the encoder and the decoder, the h_k values have to be derived from the quantized arcsine values in both cases.

While it is up to the encoder how to determine a set of suitable arcsine coefficients, the conversion of those values α_k back to predictor coefficients h_k has to be exactly the same in both encoder and decoder.

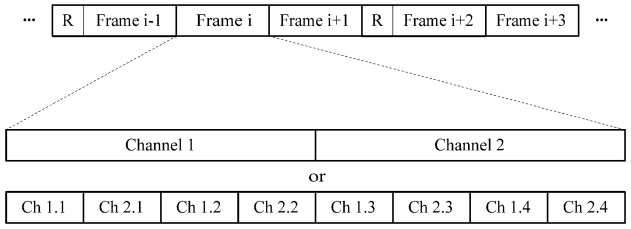

Fig. 3. Bitstream structure (R: random access info)

3.3. Block Length Switching

The basic version of the encoder uses one sample block per channel in each frame, where the frame length can initially be adjusted to the sampling rate of the input signal, e.g. 2048 for 48 kHz or 4096 for 96 kHz (approximately 43 ms in each case).

While the frame length is constant for one input file, optional *block length switching* enables a subdivision into four shorter sub-blocks to adapt to transient segments of the audio signal. Thus, either one long block or four short sub-blocks are used in each frame, e.g. 1×4096 or 4×1024 samples (Figure 3).

3.4. Random Access

Random access enables fast access to any part of the encoded audio signal without costly decoding of previous parts. The encoder optionally generates bitstream information allowing random access at intervals of several frames by inserting frames that can be decoded without decoding previous frames. In those *random access frames*, no samples from previous frames are used for prediction. Each random access frame starts with an info field (Figure 3) that specifies the distance in bytes to the next random access frame, thus enabling a fast search inside the compressed file.

3.5. Joint Stereo Coding

Joint stereo coding can be used to exploit dependencies between the two channels of a stereo signal. It is straightforward to process the two channels $x_1(n)$ (left) and $x_2(n)$ (right) independently. A simple way to exploit dependencies between the channels is to encode the difference signal

$$d(n) = x_2(n) - x_1(n) \tag{4}$$

instead of $x_1(n)$ or $x_2(n)$. Switching between $x_1(n)$, $x_2(n)$ and $d(n)$ in particular frames depends on which two signals can be coded most efficiently. Such prediction with switchable difference coding is beneficial in cases where both channels are very similar.

3.6. Entropy Coding of the Residual

In default mode, the residual values $e(n)$ are entropy coded using Rice codes. For each block, either all values can be encoded using the same Rice code, or the block can be further divided into four parts, each encoded with a different Rice code. The indices of the applied codes have to be transmitted, as shown in Figure 1. Since there are different ways to determine the optimal Rice code for a given set of data, it is up to the encoder to select suitable codes depending on the statistics of the residual.

Alternatively, the encoder can use a more complex and efficient coding scheme called BGMC (Block Gilbert-Moore Codes), proposed by RealNetworks [10]. In BGMC mode, the encoding of residuals is accomplished by splitting them in two categories: Residuals that belong to a central region of the distribution, $|e(n)| < e_{\max}$, and ones that belong to its tails. The residuals in tails are simply re-centered (i.e. for $e(n) > e_{\max}$ we have $e_t(n) = e(n) - e_{\max}$) and encoded using Rice codes as described earlier. However, to encode residuals in the center of the distribution, the BGMC encoder splits them into LSB and MSB components first, then it encodes MSBs using block Gilbert-Moore (arithmetic) codes, and finally it transmits LSBs using direct fixed-lengths codes. Both parameters $e_{\max}$ and the number of directly transmitted LSBs are selected such that they only slightly affect the coding efficiency of this scheme, while making it significantly less complex.

4. COMPRESSION RESULTS

The MPEG-4 ALS encoder was compared with Monkey's Audio Codec (MAC) [11], version 3.97, using maximum compression. The results for the ALS encoder were determined for both Rice and BGMC mode, with all other options set to maximum compression. The test material was taken from the standard audio sequences for MPEG-4 Lossless Coding. It comprises almost 1 GB of stereo waveform data with sampling rates of 48, 96, and 192 kHz, and resolutions of 16 and 24 bits.

4.1. Compression Ratio

In the following, the compression ratio is defined as

$$C = \frac{CompressedFileSize}{OriginalFileSize} \cdot 100\%, \tag{5}$$

where smaller values mean better compression. The results for the examined audio formats are shown in Table 1.

Format	ALS-R	ALS-B	MAC
48 kHz / 16-bit	46.5	46.0	45.3
48 kHz / 24-bit	64.0	63.6	63.2
96 kHz / 16-bit	31.1	30.4	30.9
96 kHz / 24-bit	47.1	46.7	48.1
192 kHz / 16-bit	21.9	21.1	22.2
192 kHz / 24-bit	38.2	37.8	39.1
Total	**41.1**	**40.6**	**41.3**

Table 1. Compression ratio (percentage) for different audio formats (grand total over all tracks). ALS results for Rice (-R) and BGMC (-B) mode.

The compression ratios of the MPEG-4 ALS encoder are either comparable or better than those of Monkey's Audio. On average, a slightly improved compression is achieved. However, particularly for high-definition material (i.e. 96 kHz / 24-bit and above), MPEG-4 ALS performs clearly better. The use of BGMC even further improves compression, at the expense of a slightly increased encoder and decoder complexity.

4.2. Complexity

The MPEG-4 ALS encoder, at maximum compression, has approximately twice the complexity of Monkey's Audio, but one should take into account that the tested ALS codec binary was not yet optimized for speed. However, while encoder and decoder complexity of Monkey's Audio are nearly equal (symmetric codec), the ALS decoder complexity is significantly lower than that of Monkey's Audio [5].

The ALS encoder is designed to offer different compression levels. While the *maximum* level achieves the highest compression at the expense of slowest encoding speed, the faster *medium* level, compared to the results in Table 1, only leads to less than 0.5% degradation [12].

Table 2 shows the encoding and decoding speed for both levels. The tests were conducted on a 1.2 GHz Pentium III-M, with 512 MB of memory, using the same audio material as in the compression ratio section.

	Medium		Maximum	
Format	**Enc**	**Dec**	**Enc**	**Dec**
48 kHz / 16-bit	19.3	34.4	5.3	23.2
48 kHz / 24-bit	14.3	24.6	4.5	18.2
96 kHz / 16-bit	11.6	21.2	2.8	13.2
96 kHz / 24-bit	7.9	13.7	2.3	10.0
192 kHz / 16-bit	5.3	10.2	1.6	8.0
192 kHz / 24-bit	3.2	6.3	1.2	5.1

Table 2. Encoding and decoding speed factor for stereo signals, compared to real-time processing.

For example, a 96 kHz / 24-bit stereo signal can be encoded nearly eight times faster than real-time at the medium compression level. Decoding is typically 2-5 times faster than encoding. The results show that the coding scheme is capable of processing even multichannel data in real-time.

5. APPLICATIONS

Applications for lossless audio coding exist in different areas, including archival systems, studio operations, and file transfer for collaborative working or music distribution over the internet (download services). In general, lossless coding is required wherever audio data is designated to be stored, transmitted, or processed without introducing any coding artifacts - even if they would be imperceptible.

A global MPEG standard for lossless audio coding will facilitate interoperability between different hardware and software platforms, thus promoting long-lasting multivendor support.

6. STANDARDIZATION PROCESS

In July 2003, the reference model software and documentation was supplied to MPEG, and a first working draft document was issued. In October 2003, the BGMC algorithm was added to the current working draft [6]. Subsequently, further improvements and extensions of the ALS coding scheme will have to be considered by the audio subgroup.

Prospective extensions of the codec will cover even higher resolutions and sampling frequencies, support for other input formats such as floating point [13] [14], coding of multichannel material [15] [16], and the improvement of coding efficiency.

MPEG-4 Audio Lossless Coding (ALS) is expected to be an international standard by the end of 2004.

7. REFERENCES

[1] ISO/IEC 14496-3:2001, "Information technology - Coding of audio-visual objects - Part 3: Audio," *International Standard*, 2001.

[2] ISO/IEC JTC1/SC29/WG11 N5040, "Call for Proposals on MPEG-4 Lossless Audio Coding," *61st MPEG Meeting, Klagenfurt, Austria*, July 2002.

[3] M. Bosi et al., "ISO/IEC MPEG-2 Advanced Audio Coding," *J. Audio Eng. Soc.*, vol. 45, no. 10, October 1997.

[4] ISO/IEC JTC1/SC29/WG11 N5383, "Report on Responses to Call for Proposals for Lossless Audio Coding," *63rd MPEG Meeting, Awaji Island, Japan*, October 2002.

[5] ISO/IEC JTC1/SC29/WG11 N5576, "Analysis of Audio Lossless Coding Performance, Complexity and Architectures," *64th MPEG Meeting, Pattaya, Thailand*, March 2003.

[6] ISO/IEC JTC1/SC29/WG11 N6012, "WD 1 of ISO/IEC 14496-3:2001/AMD 4, Audio Lossless Coding (ALS)," *65th MPEG Meeting, Brisbane, Australia*, October 2003.

[7] T. Liebchen, "MPEG-4 Lossless Coding for High-Definition Audio," *115th AES Convention*, 2003.

[8] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[9] T. Robinson, "SHORTEN: Simple lossless and near-lossless wavform compression," *Technical report CUED/F-INFENG/TR.156, Cambridge University Engineering Department*, 1994.

[10] ISO/IEC JTC1/SC29/WG11 M9893, Y. Reznik, "Proposed Core Experiment for Improving Coding of Prediction Residual in MPEG-4 Lossless Audio RM0," *65th MPEG Meeting, Trondheim, Norway*, July 2003.

[11] "Monkey's Audio," *www.monkeysaudio.com*.

[12] ISO/IEC JTC1/SC29/WG11 M9314, T. Liebchen, "Technology for MPEG-4 Lossless Audio Coding," *63rd MPEG Meeting, Awaji Island, Japan*, October 2002.

[13] D. Yang and T. Moriya, "Lossless Compression for Audio Data in the IEEE Floating-Point Format," *115th AES Convention*, 2003.

[14] ISO/IEC JTC1/SC29/WG11 M10055, T. Moriya and D. Yang, "Proposal of Core Experiment for Lossless Coding of Audio in Floating-point Format," *65th MPEG Meeting, Brisbane, Australia*, October 2003.

[15] T. Liebchen, "Lossless Audio Coding Using Adaptive Multichannel Prediction," *113th AES Convention*, 2002.

[16] ISO/IEC JTC1/SC29/WG11 M10080, T. Liebchen, "Proposed Core Experiment for Supporting Multichannel, 32-bit PCM, and Arbitrary Input File Formats in MPEG-4 ALS," *65th MPEG Meeting, Brisbane, Australia*, October 2003.