Audio Engineering Society
Convention Paper 6449
Presented at the 118th Convention
2005 May 28–31 Barcelona, Spain

Improved Forward-Adaptive Prediction for MPEG-4 Audio Lossless Coding

Tilman Liebchen[1], and Yuriy A. Reznik[2]

[1] *Technical University of Berlin, Berlin, Germany*

[2] *RealNetworks, Inc., Seattle, WA, USA*

Correspondence should be addressed to Tilman Liebchen (liebchen@nue.tu-berlin.de)

ABSTRACT

MPEG-4 Audio Lossless Coding (ALS) is a new addition to the suite of MPEG-4 audio coding standards. The ALS codec is based on forward-adaptive linear prediction, which offers remarkable compression even with low predictor orders. Nevertheless, performance can be significantly improved by using higher predictor orders, more efficient quantization and encoding of the predictor coefficients, and adaptive block length switching. The paper describes the basic elements of the ALS codec with a focus on these recent improvements. It also presents the latest developments in the standardization process and describes several important applications of this new lossless audio format in practice.

1. INTRODUCTION

Lossless audio coding permits 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 addition to the MPEG-4 audio standard [1], Audio Lossless Coding (ALS) will define methods for lossless coding of audio signals with arbitrary sampling rates, resolutions of up to 32 bit, and up to 256 channels [2]. In July 2003, the lossless codec from Technical University of Berlin was chosen as the first working draft. Since then, further improvements and extensions have been integrated. MPEG-4 ALS is expected to become an international standard by the end of 2005.

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.

Although remarkable compression is achieved even

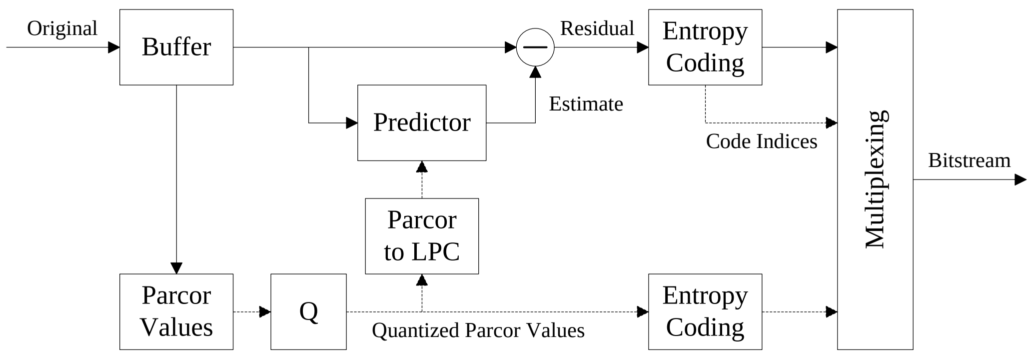

Fig. 1: MPEG-4 ALS encoder

for low predictor orders, still better compression becomes possible using high-order prediction. In this case, more efficient coding of the predictor coefficients is necessary in order to limit the amount of side information. This is achieved by applying a non-linear compander to the most important coefficients, followed by linear quantization and entropy coding of the quantized values. In addition, adaptive block length switching is used to account for changing signal statistics. As a result, compression ratios are comparable to the best high-order backward-adaptive prediction schemes, but with a significantly less complex decoder, and maintaining full random access to arbitrary parts of the encoded signal.

The paper constitutes an update of previous publications on MPEG-4 ALS [3][4][5]. The following chapters will provide a more detailed description of the codec. Following an overview of the codec structure in section 2, section 3 puts the main focus on linear prediction together with block switching and random access. Section 4 illustrates methods for joint channel coding, and section 5 describes the entropy coding scheme for the prediction residual. Coding results for a variety of audio material (including high-resolution and multichannel) are given in section 6, while section 7 provides a discussion of application scenarios for lossless audio coding in general and MPEG-4 ALS in particular.

2. STRUCTURE OF THE CODEC

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.

This section gives an overview of the basic encoder and decoder functionality.

2.1. Encoder

The MPEG-4 ALS encoder consists of several basic elements. Figure 1 shows the typical processing for one input channel of audio data.

A buffer stores one block of input samples, and an optimum set of parcor coefficients is calculated for each block. The number of coefficients, i.e. the order of the predictor, can be adaptively chosen as well. The quantized parcor values are entropy coded for transmission, and converted to LPC coefficients for the prediction filter which calculates the prediction residual. The residual is entropy coded using different entropy codes. The indices of the chosen codes have to be transmitted as side information. 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 (section 3.4), random access (section 3.5) and joint channel coding (section 4). The encoder may use these options to offer several compression

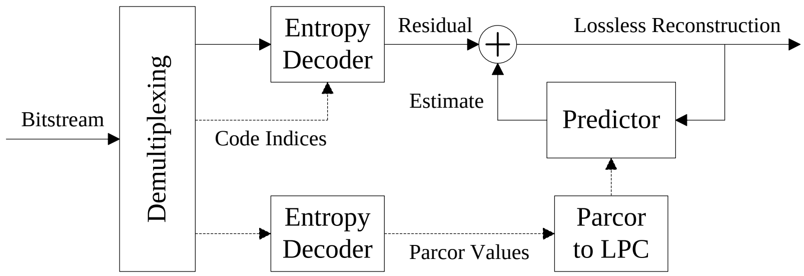

Fig. 2: MPEG-4 ALS decoder

levels with different complexities. The basic version of the encoder uses a fixed block length. Optionally, the encoder can switch between different block lengths to adapt to stationary regions as well as to transient segments of the audio signal. The codec allows random access in defined intervals down to some milliseconds, depending on the block length. Furthermore, joint channel coding is used to exploit dependencies between channels of stereo or multichannel signals. This can be achieved by coding the difference between two channels in those segments where this difference can be coded more efficiently than one of the original channels.

The entropy coding part of the prediction residual (section 5) provides two alternative coding techniques with different complexities. Besides low-complexity yet efficient Golomb-Rice coding, the BGMC arithmetic coding scheme offers even better compression at the expense of a slightly increased complexity.

Furthermore, MPEG-4 ALS will also offer efficient compression of floating-point audio data in the 32-bit IEEE format [6]. This codec extension employs an algorithm that basically splits the floating-point signal into a truncated integer signal and a difference signal which contains the remaining fractional part. The integer signal is then compressed using the normal encoding scheme for PCM signals, while the difference signal is coded separately. A detailed description of the floating-point extension can be found in [7].

2.2. Decoder

The MPEG-4 ALS decoder (Figure 2) is significantly less complex than the encoder, since no adaptation has to be carried out. The decoder merely decodes the entropy coded residual and the parcor values, converts them into LPC coefficients, and applies the inverse prediction filter to calculate the lossless reconstruction signal.

The computational effort of the decoder mainly depends on the predictor orders chosen by the encoder. Since the *average* order is typically well below the *maximum* order, prediction with greater maximum orders does not necessarily lead to a significant increase of decoder complexity. In most cases, real-time decoding is possible even on low-end systems.

3. LINEAR PREDICTION

Linear prediction is used in many applications for speech and audio signal processing. In the following, only FIR predictors are considered.

3.1. Prediction with FIR Filters

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

The procedure of estimating the predictor coefficients from a segment of input samples, prior to filtering that segment, is referred to as forward adaptation. In that case, the coefficients have to be transmitted. If the coefficients are estimated from previously processed segments or samples, e.g. from the residual, we speak of backward adaptation. This procedure has the advantage that no transmission of the coefficients is needed, since the data required to estimate the coefficients is available to the decoder as well [8].

Forward-adaptive prediction with orders around 10 is widely used in speech coding, and can be employed for lossless audio coding as well [9][10]. The maximum order of most forward-adaptive lossless prediction schemes is still rather small, e.g. $K = 32$ [11]. An exception is the special 1-bit lossless codec for the Super Audio CD, which uses predictor orders of up to 128 [12].

On the other hand, backward-adaptive FIR filters with some hundred coefficients are commonly used in many areas, e.g. channel equalization and echo cancellation [13]. Most systems are based on the LMS algorithm or a variation thereof, which has also been proposed for lossless audio coding [14][15][16]. Such LMS-based coding schemes with high orders are applicable since the predictor coefficients do not have to be transmitted as side information, thus their number does not contribute to the data rate. However, backward-adaptive codecs have the drawback that the adaptation has to be carried out both in the encoder and the decoder, making the decoder significantly more complex than in the forward-adaptive case.

3.2. Forward-Adaptive Prediction

In forward-adaptive 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 [17]. The autocorrelation method, using the Levinson-Durbin algorithm, has the additional advantage of providing a simple means to iteratively adapt the *order* of the predictor [9]. Furthermore, the algorithm inherently calculates the corresponding parcor coefficients as well.

Another crucial point in forward-adaptive prediction is to determine a suitable predictor order. Increasing the order decreases the variance of the prediction error, which leads to a smaller bit rate R_e for the residual. On the other hand, the bit rate R_c for the predictor coefficients will rise with the number of coefficients to be transmitted. Thus, the task is to find the optimum order which minimizes the total bit rate. This can be expressed by minimizing

$$R_{total}(K) = R_e(K) + R_c(K) \tag{3}$$

with respect to the prediction order K. As the prediction gain rises monotonically with higher orders, R_e decreases with K. On the other hand R_c rises monotonically with K, since an increasing number of coefficients have to be transmitted.

The search for the optimum order can be carried out efficiently by the Levinson-Durbin algorithm, which 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 derived, 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 optimum order is found at the point where the total bit rate no longer decreases.

While it is obvious from Eq. (3) that the coefficient bit rate has a direct effect on the total bit rate, a slower increase of R_c also allows to shift the minimum of R_{total} to higher orders (where R_e is smaller as well), which would lead to better compression. Hence, efficient though accurate quantization of the predictor coefficients plays an important role in achieving maximum compression.

3.3. Quantization of Predictor Coefficients

Direct quantization of the predictor coefficients h_k is not very efficient for transmission, since even small quantization errors may result in large deviations from the desired spectral characteristics of the optimum prediction filter [8]. For this reason, the quantization of predictor coefficients in MPEG-4 ALS is based on the parcor (reflection) coefficients r_k, which can be calculated by means of the Levinson-Durbin algorithm. In that case, the resulting values are restricted to the interval $[-1, 1]$. Although parcor coefficients are less sensitive to quantization, they are still too sensitive when their magnitude is close to unity. The first two parcor coefficients r_1 and r_2 are typically very close to -1 and $+1$, respectively, while the remaining coefficients $r_k, k > 2$, usually have smaller magnitudes. The distributions of the first coefficients are very different, but high-order coefficents tend to converge to a zero-mean gaussian-like distribution (Figure 3).

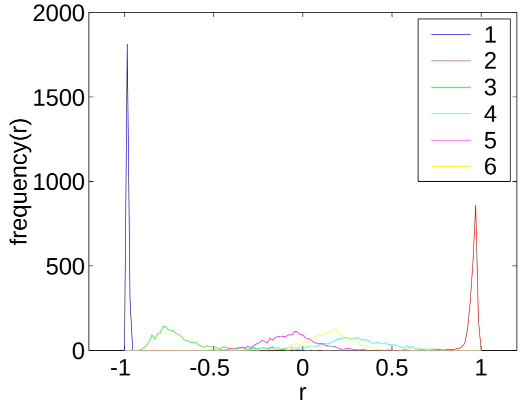

Fig. 3: Measured distributions of parcor coefficients $r_1 \ldots r_6$, for 48 kHz, 16-bit audio material.

Therefore, only the first two coefficients are companded based on the following function:

$$C(r) = -1 + \sqrt{2}\sqrt{r+1} \tag{4}$$

This compander results in a significantly finer resolution at $r_1 \to -1$, whereas $-C(-r_2)$ can be used to provide a finer resolution at $r_2 \to +1$ (see Figure 4).

However, in order to simplify computation, $+C(-r_2)$ is actually used for the second coefficient, leading to an opposite sign of the companded value. The two companded coefficients are then quantized using a simple 7-bit uniform quantizer. This results in the following values:

$$a_1 = \left\lfloor 64 \left(-1 + \sqrt{2}\sqrt{r_1+1} \right) \right\rfloor \tag{5}$$

$$a_2 = \left\lfloor 64 \left(-1 + \sqrt{2}\sqrt{-r_2+1} \right) \right\rfloor \tag{6}$$

The remaining coefficients $r_k, k > 2$ are not companded but simply quantized using a 7-bit uniform quantizer again:

$$a_k = \lfloor 64 r_k \rfloor \tag{7}$$

In all cases the resulting quantized values a_k are restricted to the range $[-64, +63]$. These quantized

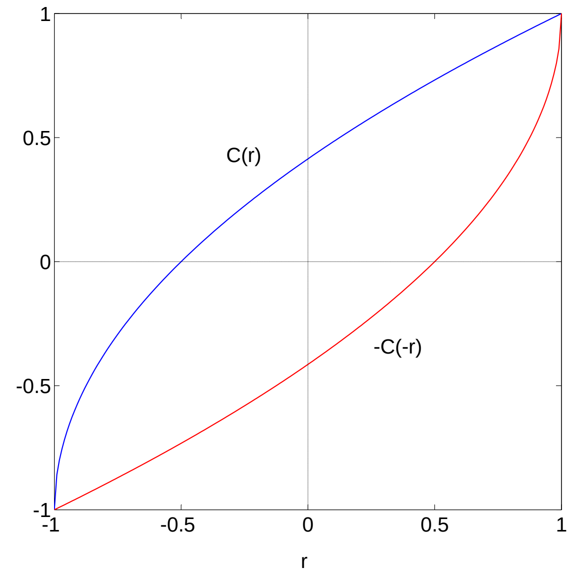

Fig. 4: Compander functions $C(r)$ and $-C(-r)$.

coefficients are re-centered around their most probable values, and then encoded using Golomb-Rice codes. As a result, the average bit rate of the encoded parcor coefficients can be reduced to approximately 4 bits/coefficient, without noticeable degration of the spectral characteristics. Thus, it is possible to employ very high orders up to $K = 1023$, preferably in conjunction with large block lengths (see section 3.4).

However, the direct form predictor filter uses predictor coefficients h_k according to Eq. (1). In order to employ identical coefficients in the encoder and the decoder, these h_k values have to be derived from the quantized a_k values in both cases (see Figures 1 and 2). While it is up to the encoder how to determine a set of suitable parcor coefficients, MPEG-4 ALS specifies an integer-arithmetic function for conversion between quantized values a_k and direct predictor coefficients h_k which ensures their identical reconstruction in both encoder and decoder.

3.4. Block Length Switching

The basic version of the encoder uses one sample block per channel in each frame. 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 of a frame into shorter blocks in order to adapt to transient segments of the audio signal. Previous versions of the MPEG-4 ALS codec [5] already included a simple block switching mechanism which allowed to encode each frame of N samples using either one full length block ($N_B = N$) or four blocks of length $N_B = N/4$. Meanwhile, an improved and more flexible block switching scheme was designed, and each frame of length N can be hierarchically subdivided into up to 32 blocks. Arbitrary combinations of blocks with $N_B = N$, $N/2$, $N/4$, $N/8$, $N/16$, and $N/32$ are possible within a frame, as long as each block results from a subdivision of a superordinate block of double length. Therefore, a partition into $N/4 + N/4 + N/2$ is possible, whereas a partition into $N/4 + N/2 + N/4$ is not (Figure 5).

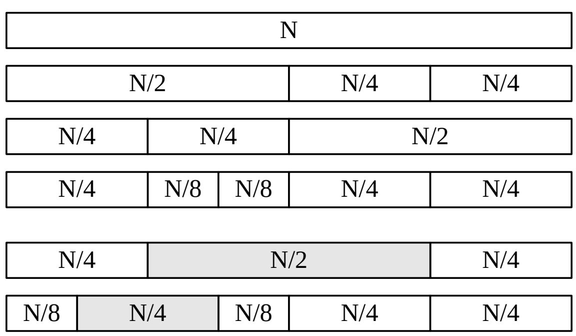

Fig. 5: Block switching examples. The last two partitions are not allowed due to the positions of the shaded blocks.

Block length switching allows the use of both very short and very long blocks within the same audio signal. For stationary segments, long blocks with high predictor orders may be chosen, while for transient segments short blocks with lower orders are more convenient. As the maximum block length is bounded by the frame length, the latter has to be chosen such that a reasonable range of block lengths is covered. For instance, a frame length of $N = 8192$ enables blocks with lengths $N_B = 8192$, 4096, 2048, 1024, 512, and 256.

The choice of a suitable block partition is entirely left to the encoder, and thus not further specified by MPEG. Possible methods may range from evaluating of the signal statistics to exhaustive search algorithms. The actual partition has to be transmitted as side information, which takes at most 32 bits per frame. Since the decoder still has to process the same number of samples per frame, block switching enables significantly improved compression without increasing the decoder complexity.

3.5. Random Access

Random access stands for fast access to any part of the encoded audio signal without costly decoding of previous parts. It is an important feature for applications that employ seeking, editing, or streaming of the compressed data.

In order to enable random access, the encoder has to insert frames that can be decoded without decoding previous frames. In those *random access frames*, no samples from previous frames may be used for prediction. The distance between random access frames can be chosen from 255 to one frame. Depending on frame length and sampling rate, random access down to some milliseconds is possible.

However, prediction at the beginning of random access frames still constitutes a problem. A conventional K-th order predictor would normally need K samples from the previous frame in order the predict the current frame's first sample. Since samples from previous frames may not be used, the encoder has either to assume zeros, or to transmit the first K original samples directly, starting the predition at position $K + 1$.

As a result, compression at the beginning of random access frames would be poor. In order to minimize this problem, the MPEG-4 ALS codec uses *progressive prediction* [18], which makes use of as many available samples as possible. While it is of course not feasible to predict the first sample of a random access frame, we can use first-order prediction for the second sample, second-order prediction for the third sample, and so forth, until the samples from position $K + 1$ on are predicted using the full K-th order predictor. Since the predictor coefficients h_k are calculated recursively from the quantized parcor coefficients a_k anyway, it is possible to calculate each coefficient set from orders 1 to K without additional costs.

In the case of 500 ms random access intervals, this scheme produces an absolute overhead of only

0.01-0.02% compared to continuous prediction without random access.

4. JOINT CHANNEL CODING

Joint channel coding can be used to exploit dependencies between the two channels of a stereo signal, or between any two channels of a multi-channel signal. While it is straightforward to process two channels $x_1(n)$ and $x_2(n)$ independently, a simple way to exploit dependencies between these channels is to encode the difference signal

$$d(n) = x_2(n) - x_1(n) \tag{8}$$

instead of $x_1(n)$ or $x_2(n)$. Switching between $x_1(n)$, $x_2(n)$ and $d(n)$ in each block can be carried out by comparison of the individual signals, depending on which two signals can be coded most efficiently (see Figure 6). Such prediction with switched difference coding is beneficial in cases where two channels are very similar. In the case of multi-channel material, the channels can be rearranged by the encoder in order to assign suitable channel pairs.

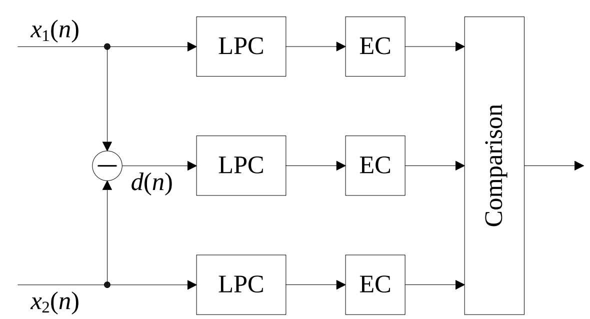

Fig. 6: Switched difference coding (LPC - prediction, EC - entropy coding).

Besides simple difference coding, MPEG-4 ALS also supports a more complex scheme for exploiting inter-channel redundancy between arbitrary channels of multichannel signals [2].

5. ENTROPY CODING OF THE RESIDUAL

In simple 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). In BGMC mode, the encoding of residuals is accomplished by splitting the distribution in two categories (Figure 7): 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.

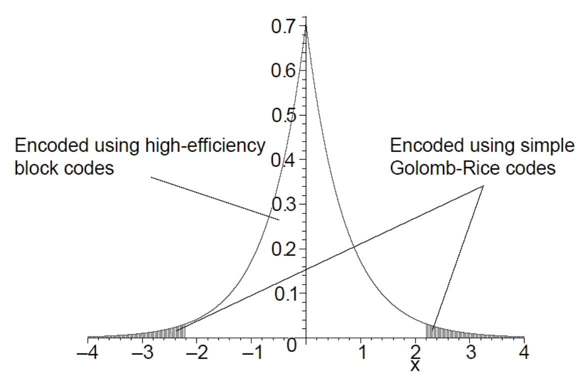

Fig. 7: Partition of the residual distribution.

A more detailed description of the entropy coding schemes used in MPEG-4 ALS is given in [3] and [19].

6. COMPRESSION RESULTS

In the following, the MPEG-4 ALS codec [20] is compared with two of the most popular programs for lossless audio compression: The open-source codec

FLAC [11], which uses forward-adaptive prediction as well, and Monkey's Audio (MAC 3.97) [16], a backward-adaptive codec identified by MPEG as the current state-of-the-art algorithm in terms of compression. Both codecs were run with options providing maximum compression (`flac -8` and `mac -c4000`). The results for the ALS encoder were determined for a medium compression level (with the prediction order restricted to $K \leq 60$) and a maximum compression level ($K \leq 1023$), both with random access of 500 ms.

Apart from the bit stream syntax, MPEG does not specify how to realize some encoder features such as predictor adaptation or block length switching. Even though we used the best ALS encoder implementation so far, future improvements in terms of compression, speed, and trade-off between those two are still possible.

The tests were conducted on a 1.7 GHz Pentium-M system, with 1024 MB of memory. The test material was taken from the standard set of audio sequences for MPEG-4 Lossless Coding. It comprises nearly 1 GB of stereo waveform data with sampling rates of 48, 96, and 192 kHz, and resolutions of 16 and 24 bits.

6.1. Compression Ratio

In the following, the compression ratio is defined as

$$C = \frac{CompressedFileSize}{OriginalFileSize} \cdot 100\%, \tag{9}$$

where smaller values mean better compression. The results for the examined audio formats are shown in Table 1 (192 kHz material is not supported by the FLAC codec).

Format	FLAC	MAC	ALS medium	ALS maximum
48/16	48.6	45.3	45.5	44.7
48/24	68.4	63.2	63.3	62.7
96/24	56.7	48.1	46.5	46.2
192/24	–	39.1	37.7	37.6
Total	–	**48.9**	**48.3**	**47.8**

Table 1: Comparison of average compression ratios for different audio formats (kHz/bits).

The results show that ALS at maximum level outperforms both FLAC and Monkey's Audio for all formats, but particularly for high-definition material (i.e. 96 kHz / 24-bit and above). Even at medium level ALS delivers the best overall compression.

6.2. Complexity

The complexity of different codecs strongly depends on the actual implementation, particularly that of the encoder. As mentioned earlier, the ALS encoder is just a snapshot of an ongoing development. Thus, we restrict our analysis to the ALS reference decoder [20], a simple C code implementation with no further optimizations. The compressed data was generated by the currently best encoder implementation.

The average CPU load for real-time decoding of various audio formats, encoded at different complexity levels, is shown in Table 2. Even for maximum complexity, the CPU load of the MPEG-4 ALS reference decoder is only around 20-25%, which in return means that file based decoding is at least 4-5 times faster than real-time.

Format	ALS low	ALS medium	ALS maximum
48/16	1.6	4.9	18.7
48/24	1.8	5.8	19.6
96/24	3.6	12.0	23.8
192/24	6.7	22.8	26.7

Table 2: Average CPU load (percentage on a 1.7 GHz Pentium-M), depending on audio format (kHz/bits) and ALS encoder complexity.

The MPEG-4 ALS codec is designed to offer a large range of complexity levels. While the maximum level achieves the highest compression at the expense of slowest encoding and decoding speed, the faster medium level only slightly degrades compression, but decoding is significantly less complex than for the maximum level (around 5% CPU load for 48 kHz material). Using a low-complexity level ($K \leq 15$, Rice coding) degrades compression by only 1-1.5% compared to the medium level, but the decoder complexity is further reduced by a factor of three (less than 2% CPU load for 48 kHz material). Thus, MPEG-4 ALS data can be decoded even on hardware with very low computing power.

7. APPLICATIONS

MPEG-4 ALS defines efficient and fast lossless audio compression techniques for both professional and consumer applications. It offers many features not included in other lossless compression schemes:

- General support for virtually any uncompressed digital audio format.
- Support for PCM resolutions of up to 32-bit at arbitrary sampling rates.
- Multi-channel / multi-track support for up to 256 channels (including 5.1 surround).
- Support for 32-bit floating-point audio data.
- Fast random access to the encoded data.
- Optional storage in MP4 file format (allows multiplex with video).

Examples for the use of lossless audio coding in general and MPEG-4 ALS in particular include both professional and consumer applications:

- Archival systems (broadcasting, studios, record labels, libraries)
- Studio operations (storage, collaborative working, digital transfer)
- High-resolution disc formats
- Internet distribution of audio files
- Online music stores (download)
- Portable music players

In the case online music stores, downloads of the latest CD releases will no longer be restricted to lossy formats such as MP3 or AAC. Instead, the consumer can purchase all tracks in full quality of the original CD, but still receive the corresponding files at reduced data rates.

Furthermore, MPEG-4 ALS is not restricted to audio signals, since it can also be used to compress many other types of signals, such as medical (ECG, EEG) or seismic data.

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

8. CONCLUSION

MPEG-4 Audio Lossless Coding (ALS) is a highly efficient and fast lossless audio compression scheme for both professional and consumer applications which offers many innovative features. It is based on a codec developed by Technical University of Berlin. Further improvements and extensions were contributed by RealNetworks and NTT.

Maximum compression can be achieved by means of high prediction orders together with efficient quantization of the predictor coefficients and adaptive block length switching. Using low and medium complexity modes, real-time encoding and decoding is possible even on low-end devices.

In the course of standardization, MPEG-4 ALS has reached the FPDAM (Final Proposed Draft Amendment) stage [2] in January 2005. It is therefore expected to become an international standard by the end of 2005.

9. 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 N7016, "Text of 14496-3:2001/FPDAM 4, Audio Lossless Coding (ALS), new audio profiles and BSAC extensions," *71st MPEG Meeting, Hong Kong, China*, January 2005.

[3] T. Liebchen and Y. Reznik, "MPEG-4 ALS: An Emerging Standard for Lossless Audio Coding," *Data Compression Conference, Snowbird, USA*, 2004.

[4] T. Liebchen, Y. Reznik, T. Moriya, and D. Yang, "MPEG-4 Audio Lossless Coding," *116th AES Convention*, 2004.

[5] T. Liebchen, "An Introduction to MPEG-4 Audio Lossless Coding," *Proc. IEEE ICASSP*, 2004.

[6] ANSI/IEEE Standard 754-1985, "IEEE Standard for Binary Floating-Point Arithmetic," 1985.

[7] D. Yang, T. Moriya, and T. Liebchen, "A Lossless Audio Compression Scheme with Random Access Property," *Proc. IEEE ICASSP*, 2004.

[8] W. B. Kleijn and K. K. Paliwal, *Speech Coding and Synthesis*, Elsevier, Amsterdam, 1995.

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

[10] A. A. M. L. Bruekers, A. W. J. Oomen, R. J. van der Vleuten, and L. M. van de Kerkhof, "Lossless Coding for DVD Audio," *101st AES Convention*, 1996.

[11] "FLAC - Free Lossless Audio Codec," *http://flac.sourceforge.net*.

[12] E. Janssen and D. Reefman, "Super Audio CD: An Introduction," *IEEE Signal Processing Magazine*, July 2003.

[13] G.-O. Glentis, K. Berberidis, and S. Theodoridis, "Efficient Least Squares Adaptive Algorithms For FIR Filtering," *IEEE Signal Processing Magazine*, July 1999.

[14] G. Schuller, B. Yu, and D. Huang, "Lossless Coding of Audio Signals Using Cascaded Prediction," *Proc. ICASSP 2001, Salt Lake City*, 2001.

[15] R. Yu and C. C. Ko, "Lossless Compression of Digital Audio Using Cascaded RLS-LMS Prediction," *IEEE Trans. Speech and Audio Processing*, July 2004.

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

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

[18] T. Moriya, D. Yang, and T. Liebchen, "A Design of Lossless Compression for High-Quality Audio Signals," *International Conference on Acoustics, Kyoto, Japan*, 2004.

[19] Y. Reznik, "Coding of Prediction Residual in MPEG-4 Standard for Lossless Audio Coding (MPEG-4 ALS)," *Proc. IEEE ICASSP*, 2004.

[20] "MPEG-4 ALS Reference Software," *ftp://ftlabsrv.nue.tu-berlin.de/mp4lossless/*.