A Design of Lossless Compression for High-Quality Audio Signals

Takehiro Moriya *, Dai Tracy Yang ** and Tilman Liebchen ***

* NTT Cyber Space Labs., Tokyo, Japan t.moriya@ieee.org
** University of Southern California, Los Angeles, USA daiyang@alumni.usc.edu
*** Technical University of Berlin, Berlin, Germany liebchen@nue.tu-berlin.de

Abstract

Three extension tools for extending and enhancing the compression performance of prediction-based lossless audio coding are proposed. The first extension aims at supporting floating-point data input in addition to integer PCM data. The second is progressive-order prediction of the starting samples at each random-access frame, where the information on previous frame is not available. The third is inter-channel joint coding. Both predictive coefficients and prediction-error signals are efficiently coded making use of the inter-channel correlation. These new prediction tools will contribute to enhance the forthcoming MPEG-4 Audio Lossless Coding (ALS) scheme, currently being under development as an extension of the ISO/IEC MPEG-4 audio standard.

1. Introduction

For archiving and broadband transmission of music signals, compression schems with lossless reconstruction become more attractive than high-compression perceptual coding schemes such as MP3 or AAC. Although DVD-audio and Super Audio CD [1, 2] include proprietary lossless compression schemes, there is a demand for an open and general compression scheme among content-holders and broadcasters. In response to this demand, a new lossless coding is being defined as an extension to the MPEG-4 Audio standard [3, 4].

In the course of this standardization process, a time-domain compression scheme based on linear predictive coding (LPC) was defined as a reference model. This model was proposed by the Technical University of Berlin [5] and the decoding process is shown in Fig. 1. For every frame, the optimum LPC coefficients are calculated and the associated PARCOR coefficients [6, 7] are quantized in an arcsine-transformed domain. The prediction error signal is derived by the quantized predictive coefficients and coded with a Rice code. For stereo signals, simple inter-channel coding is applied, where either the L-channel or R-channel and the difference between the R- and L-channel are coded.

This paper proposes three extension tools for prediction-based lossless coding. The first is support for floating-point data. The second is progressive-order prediction to improve compression performance of starting samples at each random-access frame. The third is inter-channel joint coding of both predictive coefficients and prediction error signals. In the following sections, all three tools are described and the results of performance evaluation are given.

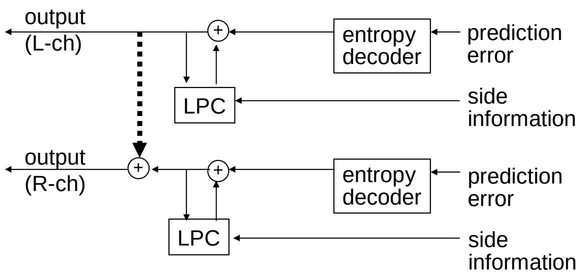

Figure 1: *Decoding process of the reference predictive-coding system with simple inter-channel prediction.*

2. Floating-point input

The IEEE-754 floating point format [8] is widely used as a data type for general computation as well as audio signals because it provides simplicity in editing, mixing, and modification and relieves the designer of having to be concerned about amplitude overflow. Byte-wise compression schemes such as "gzip" are inefficient for this floating-point format because it consists of a sign bit, an 8-bit exponent, and a 23 bit mantissa.

We propose decomposition of the floating-point data into a truncated integer and a signal representing the difference between the original floating-point data and the floating-point data as reconverted from the truncated integer. As a result of this decomposition, we can make use of any efficient prediction tools for integer sequences and use of the relationships between difference signal and the truncated integer signal. We need send neither the sign nor the exponent of the difference, since both are always zero. Furthermore, if M is the 16 bit truncated absolute integer obtained from the floating-point data and n is the necessary bits for representing the difference between the recovered and original mantissas, then n is uniquely determined according to the value of M, as shown in eq. (1).

$$n = \begin{cases} 32 & \text{if } M = 0 \\ 23 - k & \text{if } 2^k <= M < 2^{k+1} \quad k = 0, \ldots, 14 \end{cases} \tag{1}$$

The decoding process is shown in Fig. 2, where information on word boundary (necessary bits) for the difference data of each sample is provided by the reconstructed integer value.

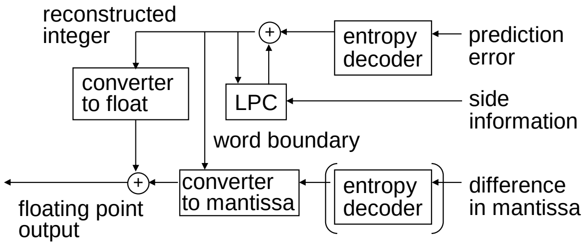

Figure 2: *Process of decoding floating-point data (single channel).*

3. Progressive-order prediction

3.1. Random access

The samples of an audio signal usually show strong correlation in the time domain. Auto-regressive linear prediction is well-known as one of the most powerful and simple tools for reducing the amplitudes of error signals, which in turn enables reductions of bit rate [2]. In the editing and playback of audio signals, however, the ability to start from randomly accessible points is desirable. At these points, we have to be able to reconstruct perfect signals without using any of the previous signal information. This leads to a significant loss of compression performance, since the auto-regressive prediction must be shut off at each such point. In the reference system, the first p samples, where p is the prediction order, are kept unchanged and require separate entropy coding due to their large amplitudes.

3.2. Progressive prediction

Progressive-order prediction is useful for the starting samples of random-access frames, since this technique makes full use of the available samples and thus reduces prediction error by as much as possible. While it is of course impossible to predict the first sample, the second sample is predictable by first-order prediction from the first sample. The prediction error at the $(q+1)$-th sample is in general derivable by q-th-order prediction.

For this progressive-order prediction, PARCOR coefficients can be directly applied in a lattice-form [6, 7] of prediction. Alternatively, we can recursively calculate conventional LPC coefficients from the PARCOR coefficients. Although they need to be calculated for every prediction order q up to p, these calculations are in any case necessary for a typical direct-prediction filter.

Examples of the waveform around a random access point are shown in Fig. 3. Waveforms (a), (b), and (c) respectively represent the original input signal, the conventional prediction error in the p non-predicted samples after the random access point, and the prediction errors produced by progressive-order prediction. The first sample in (c) has a significantly larger amplitude than the later samples and so we need to use special coding for this sample. Since prediction errors for the second and third samples are available through first- and second-order prediction, their amplitudes are lower than that for the first sample, so we only need a Rice code for lower amplitude than the first sample. We found that the prediction errors for the fourth and later samples are also small enough that we are able to use the same Rice code as for normal continuous-prediction errors.

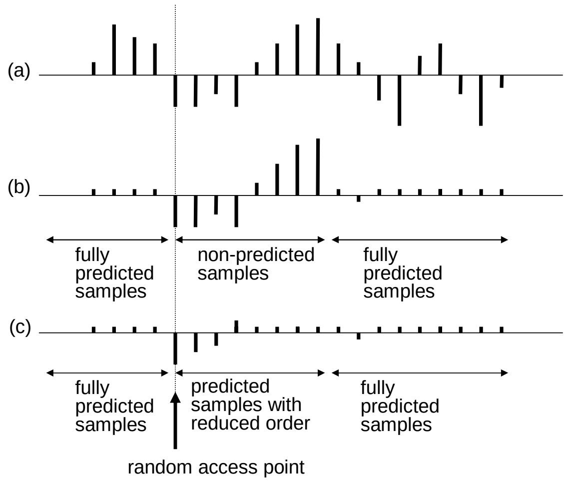

Figure 3: *Examples of waveforms around a random access point: (a) input signal, (b) error signal by non-progressive prediction, and (c) error signal by progressive prediction.*

4. Inter-channel joint coding

4.1. Differential coding of PARCOR coefficients

In the reference system, the PARCOR coefficients are independently quantized for each of the channels. However, there is a strong similarity between the PARCOR coefficients for the two channels of a stereo signal. One way to take advantage of this is to reuse the coefficients of one channel on the other, saving bit rate for the coefficients at the cost of a larger amplitude of the prediction error signals. The other is to use differential coding of the PARCOR coefficients or coding coefficients for the R- and L-channels. We can exploit the lower amplitude of the difference between PARCOR coefficients by using the same Rice code as for the prediction-error sequence. Both configurations are shown below in Fig. 4.

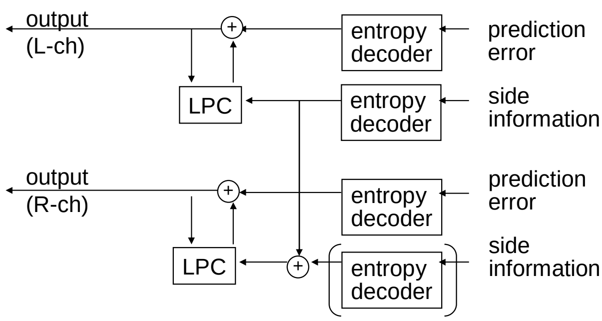

Figure 4: *Decoding process for the inter-channel prediction of PARCOR coefficients: R-channel PARCOR coefficients may be differentially coded or skipped.*

4.2. Inter-channel prediction of prediction error

For some stereo signals, i.e. those where the inter-channel correlation is particularly strong, simple differential coding in the time domain is useful. However, adjacent samples of

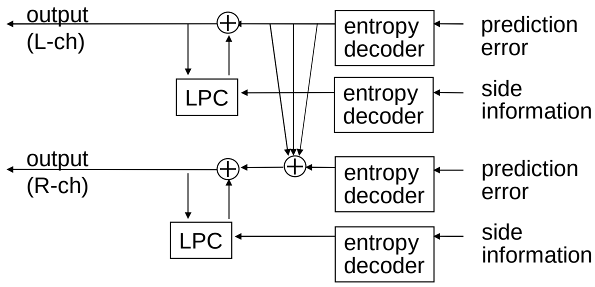

Figure 5: *Decoding process for inter-channel prediction of the prediction error signal when three-tap-based weighting is used.*

most signals are correlated with each other, so it is generally more effective to use time-domain linear prediction. The prediction error signals then still reflect the correlation between channels, and prediction based on the cross-correlation function can further reduce the amplitude of the error. We use $\mathbf{X}(x(0),\ldots,x(N-1))$ to denote the first or L-channel and $\mathbf{Y}(y(0),\ldots,y(N-1))$ to denote the second or R-channel. N is the number of samples and γ represents the optimum coefficient in terms of minimizing the distortion d in eq. (2).

$$d = ||\mathbf{Y} - \gamma\mathbf{X}||^{\mathbf{2}} \tag{2}$$

$$\gamma = \frac{\mathbf{X}_0^T\mathbf{Y}_0}{\mathbf{X}_0^{\mathbf{T}}\mathbf{X}_0}, \tag{3}$$

$$\text{where}\quad \mathbf{X}_j^{\mathbf{T}}\mathbf{Y}_k = \sum_{i=0}^{N-max(j,k)-1} x(i+j)y(i+k). \tag{4}$$

Inter-channel prediction is extensible to multi-tap cases, as in the three-tap case shown in Fig. 5. Multi-tap prediction may compensate for the small phase difference between channels. The optimum coefficients γ are found by solving equation (8), which minimizes the distortion d between the L-channel prediction error and the weighted sum of the R-channel prediction error. γ_1 and γ_{-1} usually have smaller amplitudes than γ_0. γ_1 and γ_{-1} can be quantized with two bits each and γ_0 can be quantized with four bits representing values in the range from 0 to 0.8.

$$d = \sum_{i=1}^{N-2}(y(i) - \sum_{j=-1}^{1}\gamma_j x(i+j))^2 \tag{5}$$

$$R_{j,k} = \mathbf{X}_j^{\mathbf{T}}\mathbf{X}_k \tag{6}$$

$$U_j = \mathbf{X}_j^{\mathbf{T}}\mathbf{Y}_0 \tag{7}$$

$$\begin{bmatrix} \gamma_{-1} \\ \gamma_0 \\ \gamma_1 \end{bmatrix} = \begin{bmatrix} R_{-1,-1} & R_{-1,0} & R_{-1,1} \\ R_{-1,0} & R_{0,0} & R_{1,0} \\ R_{-1,1} & R_{1,0} & R_{1,1} \end{bmatrix}^{-1} \begin{bmatrix} U_{-1} \\ U_0 \\ U_1 \end{bmatrix} \tag{8}$$

5. Performance evaluation

5.1. Floating-point data

Average compression performance of the proposed scheme (right-hand bars) was compared with that of the universal compression tool, "gzip" (left-hand bars), in Fig. 6. The vertical axis shows the compression ratio as percentage defined below.

$$\text{compression\_ratio(\%)} = 100 \times \frac{\text{compressed\_filesize}}{\text{original\_filesize}} \tag{9}$$

We prepared two types of floating-point data as test items. One has interger accuracy which was obtained by direct conversion from 16 bit integer items (6 bars at left), while the other type has full floating-point accuracy which was generated by multiplying 24 bit integer data by 0.97 (6 bars at right). We used integer test items, all of which were 30 s in duration, with 15 sampled at 48 and 96 kHz and 6 at 192 kHz. Note that all samples were provided by Matsushita Corp. specifically for use in standardizing MPEG-4 lossless coding. We used fixed-order (30th-order) linear prediction with no random-access points.

Fig. 6 clearly shows that the compression performance of the proposed scheme is better than that of "gzip" under all test conditions. In the case of the items with 16 bit integer accuracy, the file sizes were almost the same as those of the corresponding compressed 16 bit integer sequences, since all of the difference signals were zero in this case.

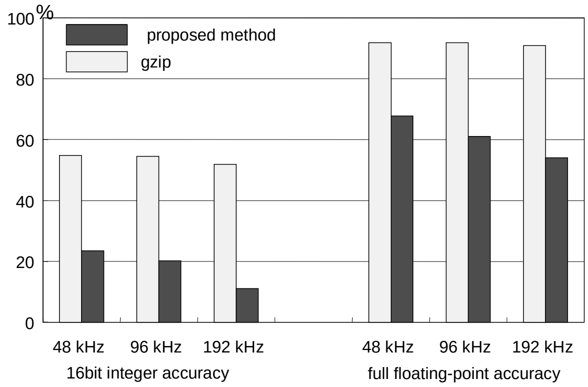

Figure 6: *Compression ratios for floating-point data.*

5.2. Progressive-order prediction

Results on compression performance for 48 kHz, 16 bit items with a 100 ms random access interval are shown in Fig. 7. The vertical axis shows the relative improvement ratios as percentages defined below.

$$\text{ratio(\%)} = 100 \times \frac{\text{filesize\_by\_reference} - \text{filesize\_by\_tested}}{\text{filesize\_by\_reference}} \tag{10}$$

For each item, the left-hand bars shows the improvement ratios for full continuous prediction compared with the reference system, in which the first p samples are not predicted. The right-hand bars show the improvement ratios for the proposed form of progressive-order prediction.

We see that the proposed progressive-order prediction system consistently improves compression to an extent approaching that of a continuous prediction system while also providing random-access capability.

5.3. Inter-channel coding

Performance improvements achieved by combined form of inter-channel coding are shown in Fig. 8, where the vertical

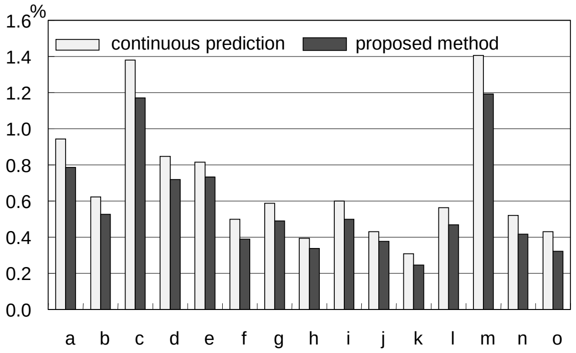

Figure 7: *Improvement ratios relative to reference-system. Fifteen input audio items are sampled at 48 kHz with 16 bit word length and labeled* **a** *through* **o**.

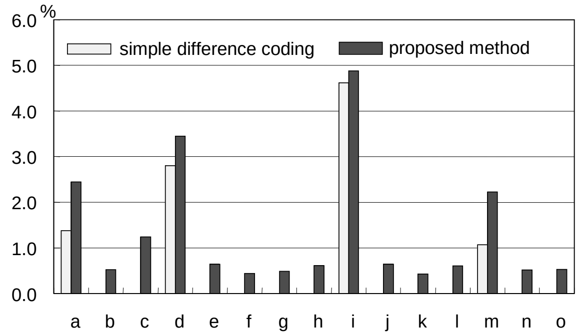

Figure 8: *Improvement ratios relative to the performance of independent channel coding, for stereo data sampled at 48 kHz and 16 bit word length.*

axis shows the relative improvement ratios as percentages defined below.

$$\text{ratio}(\%) = 100 \times \frac{\text{filesize\_by\_independent} - \text{filesize\_by\_tested}}{\text{filesize\_by\_independent}} \quad (11)$$

The combinations are listed in Table 1, in which the two-bit code indicates whether or not inter-channel coding of the coefficients (first bit) and inter-channel coding of the prediction errors (second bit) are in use. The best combination is selected and the corresponding two-bit code is included as side information. To make the selection in this case, we tried prediction twice and the compression of prediction error four times, although some forms of simple estimation may be more practical. The test items and test conditions are identical with those in the previous section. Left-hand bars show the performance improvement for the reference form of inter-channel coding in comparison with independent channel coding. The right-hand bars show the improvement over independent coding gained by the proposed coding tools.

We see that simple inter-channel coding delivers no improvement in compression ratio for some input signals, such as **b**, **c** and **e**. In contrast to this, the proposed tools are shown to be more effective than independent coding and simple inter-channel coding in all test cases.

Table 1: *Choice of combined inter-channel coding tools.*

tools	no prediction of prediction error	prediction of prediction error
substitution of coefficients	00	01
difference of coefficients	10	11

6. Conclusion

We have proposed three new tools for extending and enhancing the compression performance of prediction-based lossless audio coding. In the first, input floating-point data is decomposed into a truncated integer and the error of the mantissa, the code length of which is uniquely determined from the truncated integer. The second is an application of progressive-order prediction for the starting samples in each random-access frame. The third tool is for the differential coding of PARCOR coefficients and the 3-tap prediction of prediction error signals.

Compression testing for each item of the proposed tool has been shown to improve performance in compression over that of a conventional tool or the baseline reference system under all input and test conditions. The tools are expected to contribute to extended and enhanced performance for the ISO/IEC MPEG-4 Audio Lossless Coding (ALS) standard, which is currently under development.

Note that progressive-order coding is also useful for reducing noise due to packet loss in predictive speech-coding schemes. Furthermore, the form of inter-channel coding we have described is extensible to cases where there are more than two channels. The overall coding system is flexible and applicable to various signals, including bio-medical signals and environmental monitoring signals from sensor networks.

7. References

[1] K. Konstantinides, "An introduction to Super Audio CD and DVD-Audio," *IEEE Signal Proc. Mag.*, vol. 20, no. 4, pp. 71–82, 2003.

[2] M. Hans and R. W. Schafer, "Lossless Compression of Digital Audio," *IEEE Signal Proc. Mag.*, vol. 18, no. 4, pp. 21–32, 2001.

[3] ISO/IEC JTC1/SC29/WG11 N5040, *Call for Proposals on MPEG-4 Lossless Audio Coding*, Klagenfurt, AT, July 2002.

[4] ISO/IEC JTC1/SC29/WG11 N5208, *Revised Call for Proposals on MPEG-4 Lossless Audio Coding*, Shanghai, China, October 2002.

[5] T. Liebchen, "MPEG-4 Lossless Coding for High-Definition Audio," in *AES 115th Convention paper 5872*, New York, USA, October 2003.

[6] F. Itakura and S. Saito, "Digital Filtering Techniques for Speech Analysis and Synthesis," in *Proc. Int. Acoust. Cong. 25C-1*, Budapest, Hungary, 1971.

[7] J. D. Markel and Jr. R. M. Gray, *Linear Prediction of Speech*, Springer-Verlag, New York, USA, 1976.

[8] ANSI/IEEE Std. 754-1985 American National Standard, *IEEE Standard for Binary Floating-Point Arithmetic*, New York, 1985.