2005

Embedded lossless audio coding using linear prediction and cascade

Kevin Adistambha
University of Wollongong, ka07@uowmail.edu.au

Christian Ritz
University of Wollongong, critz@uow.edu.au

Jason Lukasiak
jl01@uow.edu.au

Abstract

Embedded lossless audio coding is a technique for embedding a perceptual audio coding bitstream within a lossless audio coding bitstream. This paper provides an investigation into a lossless embedded audio coder based on the AAC coder and utilising both backward Linear Predictive Coding (LPC) and cascade coding. Cascade coding is a technique for entropy coding of large dynamic range integer sequences that has the advantage of simple implementation and low complexity. Results show that employing LPC in an embedded architecture achieves approximately an 8% decrease in the coding rate. The overall compression performance of cascade coding closely follows Rice coding, a popular entropy coding method for lossless audio. It is also shown that performance can be further improved by incorporating a start of the art lossless coder into the proposed embedded coder.

Embedded Lossless Audio Coding Using Linear Prediction and Cascade Coding

Kevin Adistambha, Christian H. Ritz, Jason Lukasiak

ABSTRACT

Embedded lossless audio coding is a technique for embedding a perceptual audio coding bitstream within a lossless audio coding bitstream. This paper provides an investigation into a lossless embedded audio coder based on the AAC coder and utilising both backward Linear Predictive Coding (LPC) and cascade coding. Cascade coding is a technique for entropy coding of large dynamic range integer sequences that has the advantage of simple implementation and low complexity. Results show that employing LPC in an embedded architecture achieves approximately an 8% decrease in the coding rate. The overall compression performance of cascade coding closely follows Rice coding, a popular entropy coding method for lossless audio. It is also shown that performance can be further improved by incorporating a start of the art lossless coder into the proposed embedded coder .

I. INTRODUCTION

Lossless audio coding has received attention recently with MPEG's effort in standardizing MPEG-4 Audio Lossless Coding (MPEG-4 ALS) [1]. However, little attention has focused on researching embedded lossless coding. In this scheme, depicted in Fig. 1, a lossless enhancement layer is appended to an embedded lossy layer, resulting in both a lossy and lossless bitstream. The lossy layer is useful for transmission or reviewing purposes, whereas the full lossless signal would be more suitable for archival or high quality transmission purposes.

In Figure 1, the input signal, s(n) is coded with a perceptual coder to produce a synthesized version, s'(n) and a bit stream, b_p. The residual signal, r(n), is found as:

$$r(n) = s(n) - s'(n) \qquad (1)$$

The resulting residual is first decorrelated to produce a new signal, r'(n) which is then encoded with an entropy coder to produce bit stream $b_e(n)$. In the decoder, the received bitstreams are decoded to produce signals s'(n) and r'(n). The decoded signal, r'(n) is then re-correlated to produce r(n) and the original signal losslessly recovered as described in expression (2).

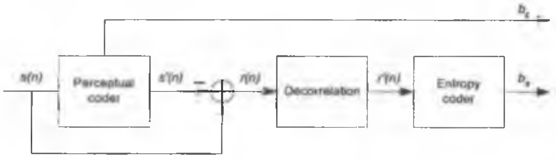

Fig. 1. Diagram of an embedded lossless audio coder.

$$s(n) = s'(n) + r(n) \qquad (2)$$

Existing approaches to embedded lossless coding include using AAC as the lossy layer [2], and using a method based on scalable image coding [3]. Work performed in [4] analyzed the characteristics of employing an AAC coder as a lossy base layer and losslessly coding the difference between the lossy base layer and the original signal (the lossless enhancement layer) using established lossless compression schemes such as gzip (based on Lempel-Ziv compression [5]) and Monkey's Audio [6].

In the field of entropy coding for audio, which is the final step in achieving lossless compression in Fig. 1, Rice coding (which is a special case of Huffman coding) is the de-facto standard [7]. It is used in many pure lossless compression algorithms such as Shorten [8], Free Lossless Audio Coder (FLAC) [9], Monkey's Audio (MAC) [6], and more recently in MPEG-4 ALS [1].

This paper examines the performance of a lossless coder based on the one described in Fig. 1. The paper extends the research described in [4] to include a decorrelation stage (based on Linear Predictive Coding (LPC)) and an entropy coding stage based on cascade coding [10, 11].

Section II will describe the embedded lossy coder adopted in this work, Section III will describe and present results for the decorrelation stage based on LPC and Section IV will provide an overview of the entropy coding stage based on cascade coding. Section V details the resulting overall compression performance of the proposed lossless coder and Section VI provides conclusions and future directions.

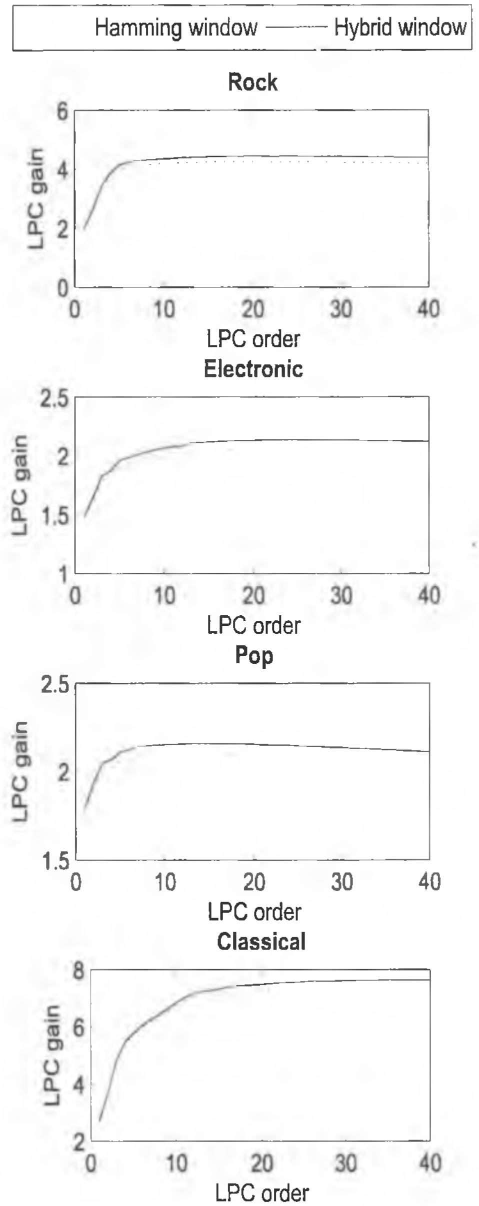

Fig. 2. LPC gain plot for backward LPC of order 1-40.

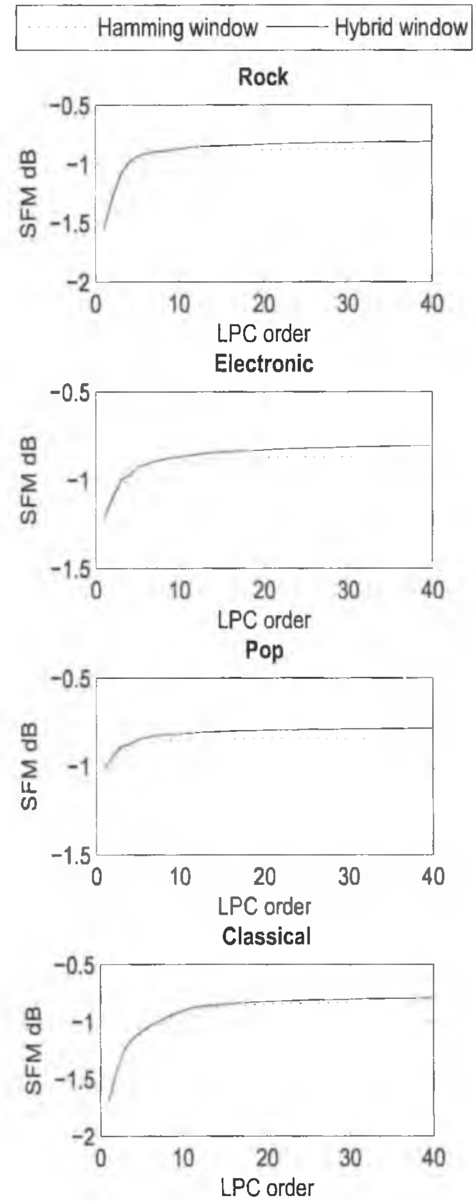

Fig. 3. SFM comparison for backward LPC order 1-40.

II. EMBEDDED LOSSY CODER

For the coder described in Fig. 1, the AAC coder [12] is used as the lossy coder or base layer. The AAC coder provides high quality audio at low bit rates [12]. The difference between the AAC synthesized signal and the original signal is denoted the AAC residual.

To achieve embedded coding, we examined the results from [4] which show the feasibility of AAC to function as the base layer of an embedded coder. Based on these results, it was decided to use AAC at the bitrate where it achieves a good trade-off between perceptual quality and a low entropy for the AAC residual, which is at 96 kbps mono [4].

III. DECORRELATION OF THE AAC RESIDUAL SIGNAL

The AAC scheme aims to produce a perceptually transparent version of the original audio rather than an exact reproduction. As such, there may still be periodic components and hence correlation in the AAC residual signal, which is defined as difference between the original and synthesised audio signals. In order to enhance the performance of the entropy coding stage, further whitening of the AAC residual may be needed. For this purpose, linear prediction coding (LPC) was chosen.

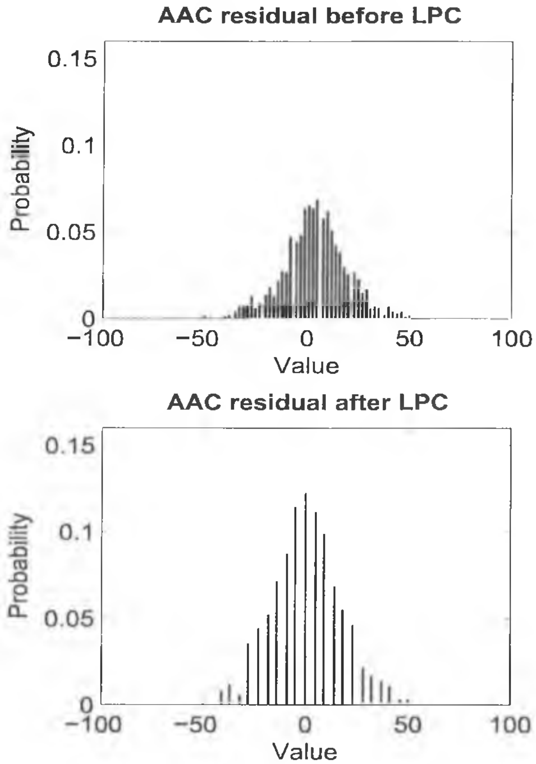

Fig. 4. One frame of an AAC residual histogram before and after LPC order 15.

A. Decorrelation via Backward LPC

Whereas pure lossless coders such as MPEG-4 ALS [1] (which is the state of the art in lossless coding) employ forward adaptive LPC due to its performance benefits, this work concentrates on the use of backward adaptive LPC in the embedded coding structure. In addition, the LPC filtering is conducted on the AAC residual signal rather than the original audio signal.

Backward adaptive LPC estimates the LPC coefficients using the previous frame and uses these to predict the current frame. The most obvious disadvantage of backward adaptive LPC is the mismatching prediction model for the current frame, making the prediction gain sub optimal. However, an advantage is not having to quantize and transmit the LPC coefficients, thus reducing bitrate overhead. This allows one to increase the prediction order without increasing the bitrate overhead [13].

The LPC process uses floating point operations, which creates rounding errors and upon resynthesis to 16 bit audio may not enable perfect reconstruction of the original signal. In order to eliminate this problem, a rounded backward adaptive LPC process was used:

$$\text{Analysis: } E = S - fix(\dot{S}) \tag{3}$$

$$\text{Synthesis: } S = E + fix(\dot{S}) \tag{4}$$

$$\dot{S}(n) = \sum_{i=1}^{p} a_i S_{n-i} \tag{5}$$

Where S is the original signal, E is the LPC residual and *fix(·)* is rounding to the nearest integer toward zero. *fix(·)* operation is used to restrict the LPC residual to integer values.

The frame length used for LPC analysis is 1024 samples, which was primarily chosen to coincide with the frame length used by the AAC coder. The LPC window used is the hybrid window described in [14], which consists of half of a Hamming window and a quarter of a cosine window where the cosine section of the window comprises 5% of the overall window length, e.g. in a window of length 1280 samples, the Hamming part is 1216 samples long and the cosine part is 64 samples long. As this window function is more heavily weighted toward the most recent samples in the window, it is far more suitable to backward LPC analysis than a symmetric window function (such as a Hamming window) as typically used in forward LPC [13]. To minimize the impact of mismatching prediction models, backward LPC analysis is performed with a ¼ frame overlap, or 256 samples in the case of a frame size of 1024 samples, and four updates of the LPC coefficients per frame.

B. Decorrelation Results

The test signals used in this work are extracted from the Q-Music database [15]. Thirty second excerpts of 20 files, consisting of 5 files each from the electronic, pop, rock and classical music genres, were downmixed to mono by taking the left channel and discarding the right channel. Audio signals in this database are sampled at 44.1 kHz and are quantized to 16 bits. The signals are then encoded using the Nero AAC encoder [16] and decoded back to PCM audio. The difference between the original signal and the AAC decoded signal is then called the AAC residual signal. This AAC residual signal is then processed with lossless backward LPC processing.

A thorough testing of the performance of LPC analysis on the AAC residual for the test set was conducted. For comparative purposes, the performance of LPC was measured using two metrics; prediction gain and spectral flatness. The calculation of these metrics is described by expressions (6) and (7) respectively.

$$G = \frac{\sum_{n=1}^{N} S_n^2}{\sum_{n=1}^{N} E_n^2} \tag{6}$$

$$SFM = 10\log_{10}\left(\frac{G_m}{A_m}\right) \tag{7}$$

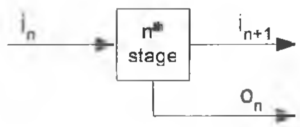

Fig. 5. Operation of cascade coding at the n^{th} stage

In expressions (6) and (7), G is the prediction gain, S is the original signal, E is the LPC residual, SFM is the Spectral Flatness Measure, G_m and A_m are the geometric mean and arithmetic mean of the LPC residual power spectrum. An SFM value close to zero indicates that the signal is noiselike. A summary of the LPC analysis results are shown in Figs 2 and 3.

Fig. 2 shows that the performance of LPC on rock, electronic and pop signals has a knee point at LPC order 5, whereas the LPC gain on classical signals has a knee point of around order 15-20. The spectral flatness curves of Fig. 3 show similar knee points to those of Fig. 2. Hence, based on these results, an LPC order of 15 is a good choice to ensure that all signals in the test database are adequately decorrelated.

C. Statistical Characteristics of the LPC Residual

As outlined in many papers such as [8] and [17], the residual of an LPC process roughly follows a Laplacian distribution, where it has zero mean and has a larger probability of a small magnitudes compared to large magnitudes. A Laplacian distribution is desirable for lossless coding purposes because it reduces the dynamic range of the signal and centres the distribution of samples around 0. Fig. 4 shows typical histograms of AAC residual frames before and after 15th order LPC processing.

In Fig. 4, the AAC residual before LPC has a mean value of 0.332 whereas the AAC residual after 15^{th} order LPC processing has a mean value of 0.108. In addition, the probability of a zero magnitude has increased. Hence, it is shown that the LPC process moves more values in the residual toward zero.

IV. ENTROPY CODING USING CASCADE CODING

A. Entropy Coding Theory

Equation (6) is the first order approximation of entropy as defined by Shannon [5] and provides the absolute minimum bits per sample required to code an independently and identically distributed (iid) stream of data into its binary representation.

$$H = -\sum_x p_x \log_2 p_x \tag{8}$$

In (8), H is the entropy of a signal in bits per sample and $p(x)$ is the probability of a symbol occurring in a stream, which can be estimated from a histogram

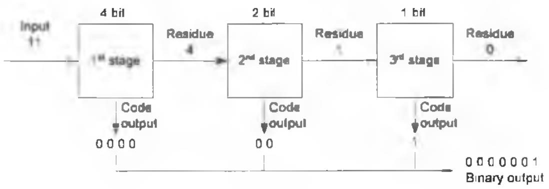

Fig. 6. Example of cascade coding of a three stages cascade coder and an input of 11

of the signal. Coding a given stream of symbols into binary taking into account the probability of each symbol occurring is typically called *entropy coding*. The performance of a given entropy coder can be measured by comparing the resulting compression rate with the entropy of the signal.

B. Cascade Coding

Cascade coding processes an input integer with a series of cascading stages, with each stage using a previously allocated number of bits to code a portion of the input integer. If a given input integer overflows the current stage, the difference between the maximum integer magnitude possible to be coded in the current stage and the input integer is then passed on (cascaded) to the next stage. This difference is called the cascade residual. The binary output of the current stage is then set to all zeros, denoting that the input overflows the current stage. This process then repeats itself for subsequent stages, until the cascade residual can be fully coded in a stage without overflow.

Fig. 5 shows the operation of one stage of a cascade coder, with the equations representing the variables represented in (9) and (10) [10].

$$i_{n+1} = i_n - \left(2^{b_n} - 1\right) \tag{9}$$

$$o_n = \begin{cases} 0, & i_n > 2^{b_n} - 1 \\ i_n, & i_n \le 2^{n_n} - 1 \end{cases} \tag{10}$$

Where i_n is the input of the n^{th} stage, o_n is the binary code output of the stage and b_n is the bit allocation for the stage. If i_{n-1}>0 then it is cascaded into the next stage.

Decoding works in reverse, by observing if the current stage has an output of all zeros. If that is the case, then the maximum value of the current stage is then added to the value of the next stage. The decoding process stops when a stage with a non zero output is encountered, with the output as the cumulative sum of the values of all stages used.

Fig. 6 is an example of a 3 stage cascade coder with bit allocation of [4 2 1] and 11 as the input. The output of the first two stages are zeros, and the output of the third stage is 1, thus the final output bit-

Compression Rate (bits/sample)

Genre	No LPC			LPC order 15			MAC on
	Cacsade	Rice	Entropy	Cascade	Rice	Entropy	residual
Classical	6.08	5.93	5.63	4.97	5.00	4.66	4.81
Electronic	8.50	7.94	7.42	8.13	7.58	7.12	7.46
Pop	8.17	7.66	7.27	7.75	7.24	6.87	7.11
Rock	8.76	8.17	7.68	7.99	7.42	7.04	7.31

Table 1: Comparison of cascade coding and Rice coding performance and the entropy of the signal with the compression performance of Monkey's Audio (MAC) operated on the AAC residual signal.

stream of the coder is 0000001. As another example, for the input of -8, the output bitstream will be 100001 with the third stage unused.

A binary output of all zeros for all stages means that the input integer is the maximum magnitude possible to be coded for a given set of bit allocations. Therefore, a binary output of all zeros for all stages does not denote that the input integer is zero. In order to decode the bitstream correctly when zero is the input, we need to intentionally overflow the input integer so that integer zero denotes the maximum magnitude:

$$|s_n| = |s_n| + 1 \quad , s_n \geq 0 \tag{11}$$

For encoding, and:

$$|s_n| = |s_n| - 1 \quad , s_n \geq 0 \tag{12}$$

For decoding, where *s(n)* is the input to the cascade encoder or decoder.

In this work, cascade coding is used to code the 15^{th} order LPC prediction residuals described in Section III. The performance of cascade coding is highly efficient for integer sequences which have a probability distribution function with zero mean with steeply descending probabilities of larger values. Such a distribution is similar to a Laplacian distribution, which is known to be a good approximation of the distribution of LPC residual sequences [8].

C. *Frame Adaptive Cascade Coding*

To minimise the coding range, finding the optimal bit allocation for each of the stages of the cascade coder is of utmost importance. It was proposed in [10] to use a curve-fitting recursive equation to find an allocation that is close to optimal. In [10] it is suggested that calculating a new set of cascade parameters for each file (or sequence) to be coded, results in good compression performance. This adaptation is suitable for stationary sequences, however, due to the time varying nature of audio, adapting the cascade parameters across an entire audio file may not yield the most efficient results. Therefore, in order to maximize efficiency, we propose a frame adaptive cascade coding technique that adapts the cascade parameters for each audio frame. As a comparison, for one signal of the classical genre, the average compression rate using frame adaptive cascade coding is 5.52 bits/sample, whereas not adapting the cascade coder allocation for each frame results in 9.00 bits/sample, a significant performance increase.

D. *Codebook-based cascade coding*

Adapting the cascade coder on a frame basis requires transmission of the coder bit allocation for each audio frame. However, sending the exact bit allocation for each stage results in a significant bitrate overhead as the decoder must be notified of both the length of each stage and the number of stages. To reduce this excessive overhead, we propose a codebook based approach to transmit the bit allocation to the decoder.

Reference [10] details optimal bit allocations for integer sequences of various magnitudes compiled by coding a Gaussian random sequence and determining the best allocation. We propose, as a preliminary study on how cascade coding performs with AAC residual signals, using the allocations described in [10] as a 15 entry codebook. Thus our implementation of the cascade coder transmits the coder configuration, via a 4 bit codebook index for each coded audio frame.

The codebook consists of entries with a description of the bit allocation and the maximum magnitude of integers that can be coded with the given allocation. The codebook entry to be used is determined by examining the maximum magnitude of the integer sequence to be coded and choosing the corresponding entry in the codebook.

V. Lossless Coding Results

For comparison, we performed experiments of entropy coding of the decorrelated AAC residual (described in Section III) using the aforementioned cascade coding structure and Rice coding. Rice coding was chosen as it is known as the de-facto standard for entropy coding of a Laplacian distribution of values with high efficiency and simple implementation [7]. The results of these experiments are shown in Table 1.

The results in Table 1 indicate that LPC processing appears to offer an overall benefit. The entropy of the signals to be coded reduces by approximately 8 % on average, with a similar reduction in the average compression rates required for either coder when LPC processing is employed.

From Table 1, for specific signals such as classical, the performance of cascade coding surpasses Rice coding by 0.03 bits per sample. However, for other genres, Rice coding results in an average saving of approximately 0.5-0.6 bits per sample compared with cascade coding. Overall, the compression rate of cascade coding lagged 0.36 bits per sample behind that of Rice coding.

An explanation of the poorer performance of cascade coding when applied to genres other than classical could be explained by referring to the entropy of the resulting signals. For classical signals, the entropy is much lower than for the other genres. In [10], the performance (in terms of compression rate) of the designed cascade coders (which are used here) significantly decrease as the entropy of the samples increase. Hence, the designed cascade coders are suboptimal for higher entropy signals.

It is also important to note that the codebook entries described in [10] and implemented in this paper are not the optimal set of allocations for an LPC residual signal, since the entries were designed using a Gaussian distribution instead of a Laplacian distribution. From the performance of Rice coding, which is designed specifically for coding a Laplacian distribution, we can conclude that the LPC residual of an AAC residual signal has a Laplacian like distribution. This observation is also confirmed in Fig. 4.

It should be noted that the side information required to decode the transmitted signal is not shown in Table 1. However for both coders this overhead is only 4 bits per frame or approximately 0.004 additional bits/sample.

From Table 1, the bits/sample rate of Monkey's Audio (MAC) is the lowest of all the methods tested, although the entropy of the decorrelated AAC residual signal is below the Monkey's Audio compression rate. Without using backward LPC however, the entropy of the AAC residual is higher than the bits/sample result of Monkey's Audio. This suggests that backward LPC of order 15 removes the majority of sample-to-sample correlation.

VI. CONCLUSIONS & FUTURE WORK

This paper has proposed an embedded lossless coding structure based upon an AAC lossy layer appended with a lossless layer consisting of an LPC stage and an entropy coding stage. Results presented indicate that performing LPC processing is beneficial in terms of lowering the entropy values of the AAC residual signal, which translates directly to bitrate savings of approximately 8% in the subsequent entropy coding stage. Furthermore, it is concluded that using backward adaptive LPC processing of order 15 results in the best overall performance for all music genres tested.

Frame adaptive cascade coding using a codebook approach as an entropy coding technique that also shows promising results for coding the LPC residual signal, as shown in Section V. However, since the cascade coding codebook used in this paper was not primarily designed to code a high magnitude integer sequence such as an LPC residual signal, the overall performance can potentially be improved by designing a new codebook specifically for target signals described in this paper. In particular, this will require an investigation into appropriate optimization techniques for the design of the cascade coder for these signals.

REFERENCES

[1] T. Liebchen, Y. A. Reznik, T. Moriya, and D. T. Yang, "MPEG-4 audio lossless coding," presented at 116th AES Convention, Berlin, Germany, 2004.

[2] R. Yu, X. Lin, S. Rahardja, and C. C. Ko, "A scalable lossy to lossless audio coder for MPEG-4 lossless audio coding," presented at Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, 2004.

[3] M. Raad, *Scalable and perceptual audio compression*, PhD thesis, University of Wollongong,2002

[4] K. Adistambha, C. H. Ritz, J. Lukasiak, and I. S. Burnett, "An Investigation Into Embedded Audio Coding Using an AAC Perceptually Lossless Base Layer," presented at Speech Science and Technology Conference 2004, Sydney, 2004.

[5] K. Sayood, *Introduction to data compression*, 2nd ed: Morgan Kaufmann Publishers, 2000.

[6] Monkey's Audio: A fast and powerful lossless audio compressor, available: http://www.monkeysaudio.com.

[7] M. Hans and R. W. Schafer, "Lossless compression of digital audio," *Signal Processing Magazine, IEEE*, vol. 18, pp. 21-32, 2001.

[8] T. Robinson, "SHORTEN: Simple Lossless and Near-lossless Waveform Compression," Cambridge University Engineering Department CUED/F-INFENG/TR.156, 1994.

[9] Free Lossless Audio Codec, available: http://flac.sourceforge.net.

[10] L. Wu, A. Zielinski, and J. S. Bird, "A cascade coding technique for compressing large-dynamic-range integer sequences," in *Digital Signal Processing*, vol. 14: Elsevier, 2004, pp. 54-71.

[11] L. Wu, A. Zielinski, and J. S. Bird, "Lossless compression of hydroacoustic image data," *Oceanic Engineering, IEEE Journal of*, vol. 22, pp. 93-101, 1997.

[12] J. Herre and H. Purnhagen, "General Audio Coding," in *The MPEG-4 Book*, F. Pereira and T. Ebrahimi, Eds.: IMSC Press, 2003, pp. 487-544.

[13] J. H. Chen, "Low-Delay Coding of Speech," in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds.: Elsevier, 1995, pp. 209-256.

[14] J.-L. Garcia, P. Gournay, and R. Lefebvre, "Backward Linear Prediction for Lossless Coding of Stereo Audio," presented at 116th AES Convention, Berlin, Germany, 2004.

[15] Q-Music royalty free music, available: http://www.q-music.co.uk.

[16] Nero AG Homepage, available: http://www.nero.com.

[17] Y. A. Reznik, "Coding of Prediction Residual in MPEG-4 Standard for Lossless Audio Coding (MPEG-4 ALS)," presented at Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). 2004 IEEE International Conference on, 2004.