Low Delay Perceptually Lossless Coding of Audio Signals

Sean Dorward, Dawei Huang, Serap A. Savari, Gerald Schuller
Bell Labs, Lucent Technologies

Bin Yu
Dept. of Statistics, University of California, Berkeley, CA

Abstract

A novel predictive lossless coding scheme is proposed. The prediction is based on a new weighted cascaded least mean squared (WCLMS) method. To obtain both a high compression ratio and a very low encoding and decoding delay, the residuals from the prediction are encoded using either a variant of adaptive Huffman coding or a version of adaptive arithmetic coding. WCLMS is especially designed for music/speech signals. It can be used either in combination with psycho-acoustically pre-filtered signals (an idea presented in [1]) to obtain *perceptually* lossless coding, or as a stand-alone lossless coder. Experiments on a database of moderate size and a variety of pre-filtered mono-signals show that the proposed lossless coder (which needs about 2 bit/sample for pre-filtered signals) outperforms competing lossless coders, such as ppmz, bzip2, Shorten, and LPAC, in terms of compression ratios. The combination of WCLMS with either of the adaptive coding schemes is also shown to achieve better compression ratios and lower delay than an earlier scheme combining WCLMS with Huffman coding over blocks of 4096 samples.

I. Introduction

In [1], [2] a new scheme for perceptually lossless coding of audio signals was proposed. The *irrelevance* of an audio signal are distortions that cannot be detected by the human ear. The schemes in [1], [2] use the combination of a pre-filter and a quantizer to remove the irrelevance on an input signal. This stage is followed by a lossless coder to reduce the *redundancy* of the signal. This separation of the coder into two distinct stages has several advantages. For example, each stage can be optimized independently of the other. This two-part coder does particularly well on speech signals when compared with other audio coders such as [3]. Therefore, this coding scheme is better suited to communications applications than earlier schemes.

To be more precise, the irrelevance reduction stage contains a "psycho-acoustic" model, which computes the signal dependent threshold of hearing over time and frequency. This psycho-acoustic model tunes the pre-filter so that its frequency response is the inverse to the threshold of hearing. The pre-filter is followed by a constant step-size uniform quantizer which introduces noise. This is illustrated in Fig. 1. The post-filter in the decoder is the inverse to the pre-filter, and hence has a frequency response similar to the threshold of hearing. The output of the post-filter is a noisy version of the original signal. However, the noise cannot be detected by the ear, because it is right at or just below the threshold of hearing. Since the pre-filter coefficients are changing, they need to be transmitted as side information to the decoder. Furthermore, we need to transmit the quantized pre-filter output, which is the primary information, to the decoder. A lossless compression scheme is applied after quantization in the encoder to remove the redundancy in the integer-valued,

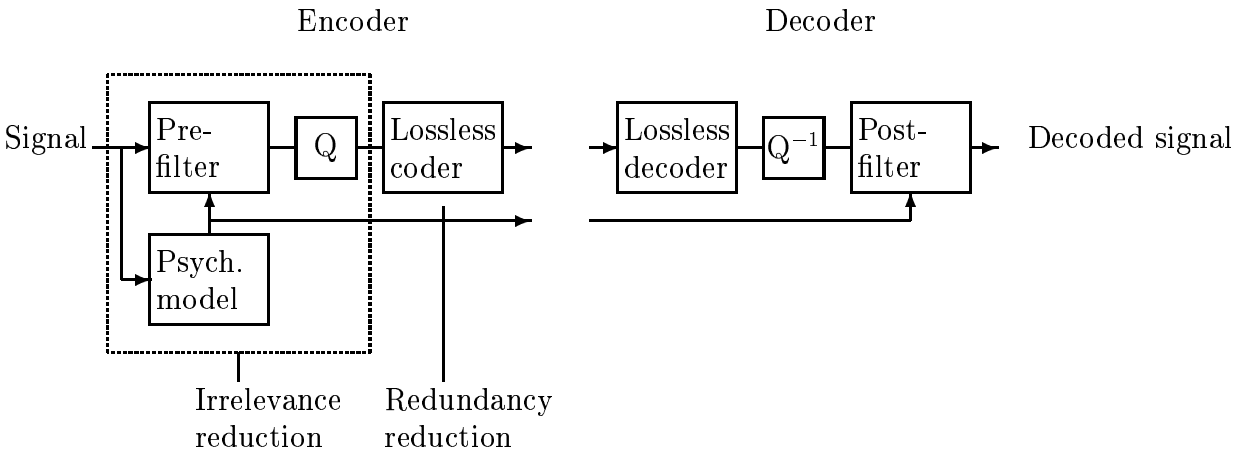

Fig. 1. The audio coding scheme with separated irrelevance and redundancy reduction, using a psycho-acoustic Pre- and Post-filter and lossless compression.

quantized, and pre-filtered signal. This is the lossless coding stage.

In the present paper we will concentrate on the lossless coding stage. To design a lossless coder that is suitable for communication applications, the objectives are to simultaneously maximize compression and minimize encoding and decoding delays. An earlier scheme [4] [5] related to the ones we will propose focused primarily on compression performance. Alternate lossless coders are usually based on blockwise prediction or transforms, which inherently require a substantial encoding/decoding delay. Furthermore, for complexity reasons, they artificially limit the prediction length and this limits the compression ratio.

We believe that for future communication applications the compression ratio and encoding and decoding delays will become increasingly critical and that additional complexity will be tolerable. These considerations motivate an examination of backward adaptive prediction schemes as opposed to blockwise prediction. To obtain high compression ratios we use a cascaded prediction scheme. We present our lossless coder in the context of pre-filtered audio signals. However, we found that it is also effective as a stand-alone lossless coder for audio signals.

II. Prediction based on WCLMS

The new prediction method Weighted Cascaded LMS (WCLMS), which was first introduced in [4], [5], has three components which we will describe in greater detail:

1. a normalized LMS (Least Mean Square) prediction scheme,
2. a cascade of three normalized LMS predictors, and
3. a Predictive Minimum Description Length (or PMDL) weighting of the cascade predictors.

A. Normalized LMS Prediction

LMS is a well-known and fast stochastic gradient algorithm to minimize adaptively the least squared prediction error or residual. Its complexity is linear in the order of the predictor. LMS is extensively applied in a number of applications, including on-line automatic control, signal processing, and acoustic echo cancellation (cf. [6]).

Let $x(n)$ be the signal at time n, and $\mathbf{x}^T(n)$ be defined by $\mathbf{x}^T(n) := [x(n-L+1), ..., x(n)]$. L is then called the order of the prediction. An L'th-order predictor is of the form

$$P(\mathbf{x}(n-1)) = \mathbf{x}^T(n-1) \cdot \mathbf{h}(n), \tag{1}$$

where $\mathbf{h}(n)$ is the L-dimensional vector of predictor coefficients at time n.

We initialize the algorithm at time 0 with $\mathbf{x}^T(0) = [0, 0, ..., 0]$, $\mathbf{h}^T(0) = [1/L, ..., 1/L]$. Let $\tilde{e}(n)$ denote the prediction error associated with the n'th prediction. For $n \geq 1$, we calculate $\tilde{e}(n)$ and $\mathbf{h}(n)$ as follows:

$$\tilde{e}(n) = x(n) - P(\mathbf{x}(n-1)) \tag{2}$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\tilde{e}(n)}{1 + \lambda ||\mathbf{x}(n-1)||^2} \mathbf{x}(n-1). \tag{3}$$

(3) is a special case of the normalized LMS procedure presented in [6, pp. 432-447]; i.e. we used only one tuning parameter λ instead of two.

Our experience shows that this prediction scheme works well for λ in the range $15 \leq \lambda \leq 25$ and across a variety of pre-filtered sound signals. For the results that we present later in the paper we use $\lambda = 20$. We observed that these signals usually take on values between -20 to 20, and these limits are determined by the pre-filter's psycho-acoustic model.

B. Cascade of the Predictors

Cascaded adaptive predictors have been used and described before, e.g., in [7]. In a cascade of predictors, the prediction error from one predictor is used as the input to the next predictor. These cascades are known to be advantageous in terms of adaptation speed, prediction accuracy, and numerical stability. However, previous cascading schemes used only the output of the final stage as the "end result" for further processing. Our predictor combines different order predictors from the cascade. The motivation for taking advantage of the extra information from intermediate stages of the cascade is that speech and audio signals have varied orders of correlations. For example, there are very nonstationary signals like the sound from castanets that need a rapidly adapting or short predictor. Some music or audio signals such as the sound from flutes are much more stationary and require higher prediction orders to accurately model the signal with all its spectral details. In our predictive coding application, we apply normalized LMS prediction three times, leading to the predictors P_1, P_2 and P_3 which we describe below. A pictorial overview of the WCLMS predictor appears in Figure 2.

After having conducted extensive experiments, we concluded that we would achieve the best compression performance by cascading three normalized LMS predictors with different orders. In our implementation of WCLMS, we choose the predictor orders to be $L_1 = 200, L_2 = 80, L_3 = 40$. This combination works well for different signals

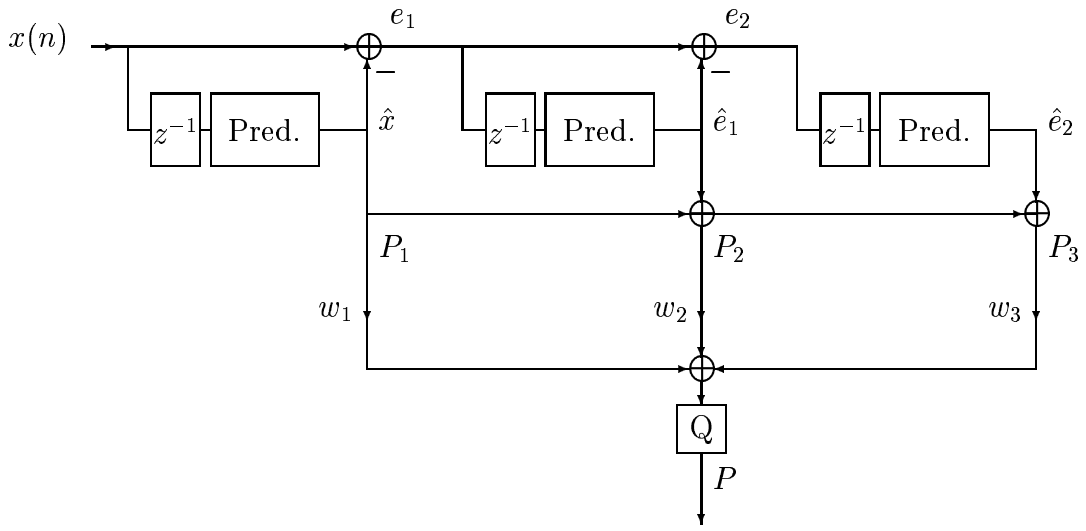

Fig. 2. The WCLMS predictor. Input $x(n)$, output $P(\mathbf{x}(n-1))$.

and at different sampling rates. To gain intuition into this phenomenon, we make the following observations. For most if not all audio and music signals, there is a dominant pattern which is close to stationary and hence requires a high order to capture. The first stage LMS predictor of order 200 finds this pattern. The remaining LMS predictors are designed to uncover short term and nonstationary behavior in the signal, and thus they use smaller orders. We next provide more details about the cascade of predictors.

Since the error terms from each normalized LMS predictor are not integers but floating point numbers, they cannot be reproduced and stored in finite precision without losing accuracy. The encoder and decoder must use the same arithmetic throughout the prediction process. One option is to use a standard arithmetic package such as the one sponsored by IEEE. For the results we discuss, we limit the precision of the residuals by using 8 bit precision after the fractional point. More generally, for any real number x, let $[x]$ denote the closest integer to x and define $[x]_A$ by

$$[x]_A \doteq A^{-1}[Ax]. \tag{4}$$

Using 8 bit precision is equivalent to choosing $A = 256$. The first predictor P_1 of $x(n)$ is a quantized version of (1):

$$P_1(\mathbf{x}(n-1)) = [\mathbf{x}^T(n-1) \cdot \mathbf{h}(n)]_A. \tag{5}$$

The quantized residual $e_1(n) = x(n) - P_1(\mathbf{x}(n-1))$ of the first predictor serves as the input to a second predictor, which is a normalized LMS predictor of order L_2. We let $\hat{e}_1(n)$ denote the quantized estimate of $e_1(n)$ which is the output of the second LMS predictor. We choose the second predictor P_2 of $x(n)$ to be given by

$$P_2(\mathbf{x}(n-1)) = P_1(\mathbf{x}(n-1)) + \hat{e}_1(n). \tag{6}$$

We denote the quantized error term associated with the second LMS predictor by

$$e_2(n) = e_1(n) - \hat{e}_1(n). \tag{7}$$

For the third prediction stage, the quantized residual $e_2(n)$ of the second predictor serves as the input to a third predictor, which is a normalized LMS predictor of order L_3. We let $\hat{e}_2(n)$ symbolize the quantized estimate of $e_2(n)$ which is the output of the third LMS predictor. We choose the second predictor P_3 of $x(n)$ to be given by

$$P_3(\mathbf{x}(n-1)) = P_2(\mathbf{x}(n-1)) + \hat{e}_2(n). \tag{8}$$

We denote the quantized error term associated with the third LMS predictor by

$$e_3(n) = e_2(n) - \hat{e}_2(n). \tag{9}$$

We will next describe how to combine the three predictors P_1, P_2, P_3 of $x(n)$ into an overall predictor P.

C. Predictive Minimum Description Length Weighting

Bayesian statistics (cf. [8]) motivates the use of a composite predictor P which is a weighted average of the predictors P_1, P_2, P_3. In particular, the three predictors can be combined into a predictor P which takes on integer values:

$$P = [\sum_i w_i P_i], \quad w_i \geq 0, \ \sum_i w_i = 1. \tag{10}$$

The relative weight w_i of predictor P_i reflects an estimate of how well P_i will predict the signal $x(n)$ given its performance to date. The relative weights are updated every time a prediction is made.

Our choice for the relative weights w_i is based on the predictive Minimum Description Length (MDL) principle (see, e.g., [9]). For this application, the basic idea is that the probability density function (abbreviated pdf) of the prediction error $e_i(n) = x(n) - P_i(\mathbf{x}(n-1))$ of predictor P_i can be modelled well for the data we have studied by a Laplacian distribution:

$$\text{pdf}(e_i(n)) \approx \frac{ce^{-c|e_i(n)|}}{2},$$

for some positive parameter c. Since signals are nonstationary, the relative weights for our composite predictor are chosen to emphasize prediction accuracy in the recent past:

$$w_i(n) \propto e^{-c(1-\mu)\sum_{i=1}^{n} |e_i(n-i)| \cdot \mu^{i-1}}. \tag{11}$$

We used $c = 2$ and $\mu = 0.9$ in our implementation of WCLMS.

III. Low Delay Lossless Coding of Residuals

The output of the prediction scheme is an integer-valued prediction P. Let

$$e(n) = x(n) - P(\mathbf{x}(n-1)).$$

Since the input signal is also integer-valued, the error terms $e(n)$ are also integer-valued. Assuming that the decoder has a copy of the WCLMS predictor used by the encoder, the decoder can perfectly recover the input signal sequence $x(0), x(1), x(2), \ldots$

from the sequence $x(0), e(1), e(2), \ldots$. Therefore, to complete our description of the perceptually lossless coder, we will discuss a few options for encoding the error terms $e(n)$.

The prediction scheme works well on all of the data we examined and the vast majority of errors have small absolute values. For most of the signals we considered, more than half of the residuals are zeroes. The original implementation of WCLMS [5] used a variant of semi-static Huffman coding to encode the error terms. A superletter was introduced to represent a pair of consecutive zeroes. Statistics on each block of 4096 samples were collected and a Huffman code over the original error alphabet plus the superletter was used to encode the block. For the version of semi-static Huffman coding that we report in the first column of Table I, we used twenty-five superletters instead of one; i.e., we group all pairs of $\{-2, -1, 0, 1, 2\}$, and this minor change generally leads to an improvement in compression ratio of up to a few percent. Our attempts to find semi-static codes which capture additional higher order dependencies did not lead to codes with better compression performance because the overhead of transmitting a larger model to the decoder dominates any potential compression improvement from the model.

Since we are concerned with delay as well as with compression ratio, we investigated numerous adaptive coding schemes. We first considered algorithms that have been very successful for lossless data compression applications. We looked at compression schemes like gzip [10], which are based on the Lempel-Ziv codes. Although they have large delay, we also examined variations of the Burrows-Wheeler coder such as bzip2 [11]. All of these achieved uniformly worse compression performance than the semi-static Huffman code. The best general purpose lossless compression algorithm is currently ppmz [12], which is in the family of Prediction by Partial Matching algorithms. Applying ppmz to the stream of residuals leads to compression results that are comparable to those obtained by semi-static Huffman coding, but ppmz is significantly slower and more complex than semi-static Huffman coding.

We next considered a few versions and variations of adaptive Huffman coding and adaptive arithmetic coding algorithms. The best adaptive Huffman codes were over the original error alphabet combined with the twenty-five superletters. The implementation borrowed ideas from [13], [14], and [15]. The best adaptive arithmetic codes were over only the original error alphabet. Our version combines techniques from [16], [17], [18], and [19]. The compression performance of these two algorithms is given in the second and third columns of Table I.

We were somewhat surprised to find that the adaptive Huffman coding algorithm almost always achieves slightly better compression than the semi-static Huffman code while reducing the delay down to about 17 samples compared to the original delay of 4096 samples. This suggests that the statistics from consecutive blocks of 4096 samples do not change rapidly and that the overhead used to transmit the semi-static Huffman coding models does not compensate for any potential gains obtained by knowing the precise semi-static frequencies.

The adaptive arithmetic coding algorithm always has the best compression amongst all the algorithms we consider and achieves about 2% better compression than the semi-static Huffman code. The delay associated with our implementation is about 100 samples. This is considerably smaller than the delay associated with semi-static Huff-

man coding but is larger than the delay for the adaptive Huffman code. The choice between using an adaptive arithmetic code and an adaptive Huffman code should be based on the relative importance of compression versus delay for the particular application.

These experiments suggest that it is very unlikely that there are statistical dependencies in the error signal that can be exploited to further improve the compression ratio of the combined algorithm.

IV. Application to Pre-filtered Signals

This section assesses the performance of the WCLMS coder when applied to mono-signals processed by the psycho-acoustic pre-filter described in [1], [2]. The results reported here do not contain the side-information for the coefficients of the post-filter, because it is the same for all schemes; this side-information generally requires between 0.03 and 0.17 bits per sample. We first consider how the combination of a WCLMS predictor with an entropy coder from the previous section performs on the output from the pre-filter. The results of these experiments form the first three columns of Table I.

We next look at the performance of the best general purpose lossless compression algorithms on the output from the psycho-acoustic pre-filter. Two of the best ones to date are ppmz [12], which uses prediction by partial matching, and bzip2 [11], which is a block sorting compressor. The results of these experiments are shown in the fourth and fifth columns of Table I. It is clear from Table I that any of the three compression schemes using a WCLMS coder is considerably better than ppmz or bzip2. The reason for this is that neither of the latter two schemes takes advantage of the structure of the output from the pre-filter, while WCLMS has been designed to do so.

Finally, we examine how the benchmark lossless audio coding schemes perform on the output from the psycho-acoustic pre-filter. We specifically consider LTAC [20], which is a Transform based lossless coder, LPAC [21], which is based on block prediction, Shorten [22], which is based on polynomial block prediction, and Wavezip [23]. LTAC has a coding part closest to traditional audio coders, because it uses a transform for compression. Meridian Lossless Packing is a lossless, prediction based coder which was recently adopted for use on DVD audio [24]. However, since it is more intended for higher sampling rates and we had no evaluation copy available, we did not include it in our comparison. The results of these experiments compose the last four columns of Table I.

A representative database is chosen to assess WCLMS and the benchmark coders performance in terms of bit rates. The database contains music, speech and mixed music/speech with sampling rates 8, 16, and 32 kHz. In Table I, chart is pop music, 16cj is classical jazz, mixed is speech with background music, spot2 is a commercial containing speech. These signals represent difficult or "critical" signals in terms of perceptually lossless compression.

Clearly, our WCLMS coder gives the best coding rate for every signal in the table: roughly, a 10% improvement over the second best LPAC, a 20% improvement over LTAC, a 25% improvement over Shorten, and a 35% improvement over WaveZip which is the common PC sound compression software. Similar results hold for other

	Huff.	ad. H.	arith.		ppmz	bzip2		LPAC	LTAC	Sho.	WZ.
32kHz											
chart	1.94	1.93	1.90		2.16	2.44		2.23	2.36	2.51	3.22
16cj	1.96	1.95	1.92		2.38	2.61		2.47	2.42	2.67	3.35
mixed	2.16	2.16	2.12		2.26	2.54		2.34	2.59	2.58	3.19
spot2	1.94	1.93	1.91		2.05	2.34		2.12	2.42	2.47	3.09
16kHz											
chart	2.01	2.00	1.97		2.45	2.69		2.49	2.55	2.68	3.42
16cj	2.02	2.02	1.97		2.64	2.88		2.64	2.56	2.85	3.48
mixed	2.27	2.27	2.23		2.41	2.68		2.50	2.80	2.67	3.23
spot2	2.18	2.17	2.14		2.32	2.61		2.38	2.75	2.63	3.27
8kHz											
chart	2.02	2.01	1.98		2.67	2.84		2.58	3.10	2.89	3.67
16cj	1.95	1.95	1.91		2.83	3.05		2.33	3.04	3.11	3.77
mixed	2.29	2.29	2.25		2.47	2.72		2.56	3.36	2.78	3.46
spot2	2.28	2.27	2.24		2.39	2.67		2.53	3.38	2.76	3.46

TABLE I

COMPARISON OF SEVERAL LOSSLESS COMPRESSION SCHEMES ON PRE-FILTERED SIGNALS. THE ALGORITHMS USED IN THE FIRST THREE COLUMNS COMBINE WCLMS (200,80,40) COMPRESSION WITH AN ENTROPY CODER FROM SECTION III; HUFF.: SEMI-STATIC HUFFMAN CODING, AD. H.: ADAPTIVE HUFFMAN CODING, ARITH.: ADAPTIVE ARITHMETIC CODING. THE ALGORITHMS USED FOR THE NEXT TWO COLUMNS ARE TWO STANDARD LOSSLESS COMPRESSION SCHEMES. THE ALGORITHMS USED FOR THE LAST FOUR COLUMNS ARE BENCHMARK LOSSLESS AUDIO COMPRESSION SCHEMES; SHO.: SHORTEN, WZ.: WAVEZIP.

samples in our database.

Thus far, we have only discussed the compression of the WCLMS coders averaged over the entire signal. Another performance metric is the peak bit rate needed among blocks of 4096 consecutive samples. This is an important consideration when designing buffers for constant bit rate channels. For the signals in the database, the peak rate of any of the WCLMS coders is not much higher than the average value over the entire signal.

V. CONCLUSIONS

We presented new perceptually lossless compression schemes for music and audio signals which are motivated by least mean square prediction. Although these schemes have a higher complexity than other lossless coders, they achieve better compression ratios than other existing schemes and have low encoding/decoding delays. We designed the combination of WCLMS prediction with variants of adaptive Huffman coding and adaptive arithmetic coding in order to reduce delays. To our surprise, they also improved compression performance over the original WCLMS coder, which uses semi-static Huffman coding. We believe that these perceptually lossless compression algorithms would be well-suited for many communication applications requiring high quality coding.

REFERENCES

[1] B. Edler and G. Schuller, "Audio Coding Using a Psychoacoustic Pre- and Post-Filter," ICASSP 2000, II, 881-884, Istanbul, Turkey, 2000.

[2] B. Edler, C. Faller, G. Schuller, "Perceptual Audio Coding Using a Time-Varying Linear Pre- and Post-filter," AES Symposium, Los Angeles, CA, Sept. 2000.
[3] V. Madisetti, D. B. Williams, eds., "The Digital Signal Processing Handbook," Chapter 42, D. Sinha et al., "The Perceptual Audio Coder (PAC)," CRC Press, Boca Raton, Fl., 1997.
[4] D. Huang, G. Schuller, B. Yu, "A lossless coder based on weighted cascade LMS prediction," submitted to International Symposium on Information Theory, Washington, DC, June 2001.
[5] G. Schuller, B. Yu, D. Huang, "Lossless coding of audio signals using cascaded prediction," submitted to ICASSP 2001.
[6] S. S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey 1999.
[7] P. Prandoni and M. Vetterli, "An FIR cascade structure for adaptive linear prediction," *IEEE Trans. Sig. Proc.*, **46**, 2566-2671, Sept. 1998.
[8] A. Gelman, H. Stein, and D. Rubin, *Bayesian data analysis*, Chapman & Hall, New York, 1995.
[9] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE. Trans. Inform. Th.* (Special Commemorative Issue: Information Theory: 1948-1998), **44**, 2743-2760, 1998.
[10] P. Deutsch., "GZIP file format specification version 4.3.", rfc1952, May 1996.
[11] J. Seward, bzip2 version 1.0.1. Source code and executable dowloadable from http://sourceware.cygnus.com/bzip2/index.html.
[12] C. Bloom, "Solving the Problems of Context Modeling," March, 1998. Available at http://www.cbloom.com/papers/ppmz.zip. See also http://www.cbloom.com/src/ppmz.html.
[13] G. V. Cormack and R. N. Horspool, "Algorithms for adaptive Huffman codes," *Inf. Proc. Let.*, **18**, 159-165, March 1984.
[14] D. E. Knuth, "Dynamic Huffman coding," *J. Alg.*, **6**, 163-180, 1985.
[15] J. S. Vitter, "Design and Analysis of Dynamic Huffman Codes", *J. ACM*, **34**(4), 825-845, Oct. 1987.
[16] A. Moffat, R. Neal, and I. H. Witten, "Arithmetic coding revisited," DCC 1995, 202-211, March 1995.
[17] L. Stuiver and A. Moffat, "Piecewise integer mapping for arithmetic coding," DCC 1998, 3-12, March 1998.
[18] A. Moffat, "An improved data structure for cumulative probability tables," *Softw. Pract. Exper.*, **29**(7), 647-659, 1999.
[19] G. G. Langdon, "An introduction to arithmetic coding," *I.B.M. J. Res. Develop.* **28**, 135-149, 1984.
[20] T. Liebchen, LTAC, version 1.71, blocksize 4096. http://www-ft.ee.tu-berlin.de/~liebchen/ltac.html
[21] T. Liebchen, LPAC, version 0.99h, setting "Extra High Compression," TU Berlin, Germany, http://www-ft.ee.tu-berlin.de/~liebchen/lpac.html
[22] Softsound, Great Britain, Shorten, version 1.03, default setting (polynomial prediction), http://www.softsound.com/Shorten.html
[23] WaveZip, version 2.00 uses MUSICompress of Soundspace, Sunnyvale, CA. http://www.gadgetlabs.com/wavezip.htm
[24] M.A. Gerzon et al.,"The MLP Lossless Compression System," AES 17th Int. Conf., Florence, Italy, 61-75. Sept. 1999.