ADAPTIVE CONTEXT BASED SEQUENTIAL PREDICTION FOR LOSSLESS AUDIO COMPRESSION

Ciprian Doru Giurcăneanu , Ioan Tăbuş and Jaakko Astola
Signal Processing Lab., Tampere University of Technology,
P.O. BOX 553, Tampere 33101, FINLAND
Tel: +358 3 3653911; fax: +358 3 3653857
e-mail: {cipriand,tabus,jta}@cs.tut.fi

ABSTRACT

In this paper we propose the use of adaptive-context-based prediction in a sequential mode for lossless audio compression. We show that lossless compression algorithms with sequential context based prediction can achieve better compression results than with forward-frame-based linear prediction. Two distinct algorithms are proposed and evaluated for audio signal sampled at 48 kHz with 16 bits/sample. The context quantization and prediction in both algorithms are similar to those used in an algorithm previously proposed for image compression[7] but new solutions are provided for modelling of errors and collecting the coding statistics. The first algorithm uses histogram bucketing in a small number of contexts in conjunction with an arithmetic coder. The second algorithm uses parametric modelling of errors in a large number of contexts in conjunction with Golomb-Rice encoding.

1 INTRODUCTION

Lossless compression of high quality audio signals has become a significant topic of research, being considered a desirable feature which must be implemented (at least as one option) in the new high quality audio applications, e.g. digital versatile disks [1]. Lossless audio compression may appear even more necessary in audio and speech archiving [4] or whenever the audio signal undergoes multiple encoding-decoding operations.

We consider here the lossless compression of high quality audio signal, sampled at 48kHz with 16 bits/sample. The application of classical universal coding methods as Lempel-Ziv related algorithms[11], or context algorithm [3][10] in their original forms does not provide a good solution for lossless compression of audio signal sampled at 16 bits/sample and it is even less effective for 20 or 24 bits/sample. Although the universal algorithms are proven to be asymptotically optimal for stationary sources, the complexity of their underlying models will add an extra term to the best achievable coding rate. This additional term is asymptotically decreasing to zero, but it will still be important for usual sizes of files (in the range of Mbytes).

One solution to the above problem is to constrain the underlying model of Context algorithm according to the apriori information about the nature and characteristics of the signals to be compressed. In this way, the universality of the coding scheme will be limited, but for the particular constrained class of signals the coding performance will be improved. Properties such as scaling, extrapolation capability, finite dynamic range of the output signal, must be apriori enforced during the modelling stage of audio signal.

The combination of linear prediction models (largely used in audio and speech modelling for compression application) with the powerful context algorithm (proven to encode at optimal rates Markov, or even more generally, tree sources) has not yet been considered for lossless audio compression. Several studies have been performed for lossless image compression [6] [7] [9], but there the set of the input signal was usually restricted to $\{0,\ldots,255\}$, while now the interesting alphabet is at least $\{0,\ldots,65535\}$. Two major problems arise when dealing with such large sizes of the alphabet: the selection of quantized contexts for coding, and the mechanism to be used in collecting the statistics for error coding.

In this paper we examine several solutions for the application of Context algorithm to audio compression: the prediction model operates in sequential mode (no parameters need to be transmitted); each context has a specific linear predictor, updated by recursive Least Squares (RLS) algorithm; the intercept in the model is estimated based on some auxiliary contexts, selected according to the order statistics of previous samples; histogram-bucketing is used for modelling the error pdf when arithmetic encoding is used to encode the errors, according to their estimated pdf; or, alternatively, parametric modelling of errors conditioned on more refined contexts makes possible the use of Golomb-Rice encoding.

2 CONTEXT BASED PREDICTION

Different solutions were considered for using the Context algorithm for compression of graylevel images, where the

most common uncompressed format is 8 bits per pixel. Several alternative techniques are available for context quantization and error modelling[6],[7],[9]. We use here context tree modelling similar to the one proposed in [7]. For each sample x_n of the audio signal we select the contextual information from a context mask containing the most recent N past samples $x_{n-N},...,x_{n-1}$. The values in the context mask are increasingly ordered resulting in $X_{(1)},...,X_{(N)}$. A primary conditioning context is selected by a decision tree of depth 2 where at each node one decision variable is compared with some fixed thresholds. The first decision variable is the absolute value of the prediction error at the preceding position $\varepsilon(n-1)$; the second one is the range of the samples inside the prediction mask, defined as $\delta = X_{(N)}\text{-}X_{(1)}$. The primary conditioning context is used for labeling the parameters of the RLS predictors and in the case of the first algorithm, also for labeling the histogram of prediction error. The thresholds in the decision tree are scaled according to the number of bits per sample in the original signal. A secondary context is associated to each primary context, in order to take into account the ranking of sample magnitudes inside the context window. A Hasse cube forms the state transition diagram of the finite state machine (FSM) selecting the secondary context[7]. In our experiments only the nodes in the middle layer of the Hasse diagram are used to specify the secondary context and therefore used for labeling the fine tuning parameters (intercepts) needed in the adaptive prediction and, in the second algorithm, for labeling the additional parameters for the Golomb-Rice encoding. We note that fully adaptive context selection, using all nodes in the Hasse diagram as presented in [6], has the potential of improving the compression rate with several percents, but unfortunately with the cost of a twofold increase in the overall complexity of the algorithm.

The context tree previously used in the *image* compression algorithm described in [7] was surprisingly found to have an optimal behaviour also for the present *audio* compression application: the best achievable performance was obtained by keeping the context structure and parameters at their optimal values found in the experiments for image compression (these parameters represent quantization thresholds, forgetting factors). This fact expresses once more the universality of Context algorithm, where, due to adaptivity, most important features in the signal (be it image or sound) can be learned on the fly, during encoding.

A recursive least squares algorithm with forgetting factor is used for updating the parameters of the predictors in various primary contexts, which was shown in [7] to be a particular case of FSM-L predictors. For each context we track the power of prediction residuals (using an exponential forgetting accumulator) and whenever it is below a threshold θ the updating step is omitted. We obtained a significant speeding up of the algorithm experimenting with different thresholds, and it appears that a feedback from the observed number of updates per processed sample can be used to adapt the threshold θ.

3 CONTEXT BASED MODELLING AND CODING OF PREDICTION ERRORS

The large alphabet size makes it very difficult to handle nonparametric distributions of prediction errors, but we present in the next subsection a solution by combining histogram tracking with histogram bucketing, which provides all necessary information needed to efficiently use arithmetic coding (algorithm FSM-L-HMB). The more direct solution of using a parametric distribution leads in the second subsection to the use of Golomb-Rice coding (algorithm FSM-L-PD).

3.1 Histogram modelling and bucketing (HMB)

Histogram tracking is widely used in signal compression to adaptively track the time varying distribution of prediction residuals. For audio signals with 16 or more bits per sample, it is not anymore possible to directly use the adaptive histogram tracking method as presented in [6] and hence we have combined adaptive histogram tracking with the histogram bucketing method[5].

The value of the error ε is first invertibly mapped to positive integers (negative errors into odd numbers and positive errors to even numbers).

$$\varepsilon' = \begin{cases} 2\varepsilon & \text{if } \varepsilon \geq 0 \\ 2|\varepsilon| - 1 & \text{otherwise} \end{cases} \quad (1)$$

We split the value ε' into two parts, and transmit them in two different ways. The M most significant bits of ε' define the integer $\varepsilon_q = \lfloor \frac{\varepsilon'}{2^M} \rfloor$. This integer will be first encoded, by means of arithmetic coding based on the observed histogram. There is one histogram corresponding to each primary context. We choose to adapt the size of the alphabet for each histogram, according to the number symbols actually observed over a time period in the corresponding context. The symbols which are not occurring frequently are not modeled by the histogram, and therefore have to be transmitted using ESC sequences.

The statistics of the $N - M$ less significant bits of ε' (the second part to be encoded) in various contexts are bucketed, by conditioning only with respect to the value of the most significant bits (the context is not taken into account at all). The quantized contexts will condition the probability distribution function of the quantized value, ε_q, while the quantization error $r_{\varepsilon_q} = \varepsilon' - M\varepsilon_q$ is conditioned on ε_q, which reflects the bucketing principle in [5]. Now for each input sample we will send two symbols, ε_q and r_{ε_q}; the first one will be encoded with the arithmetic encoder, to take full advantage of the adaptivity to symbol statistics. The gain of using an

arithmetic encoder for transmitting the second type of symbols, $\varepsilon_{\varepsilon_q}$, was found to be less than a percent from the total bitrate, when compared to the use of a Huffman coding with a fixed coding table. The high memory requirement of histogram modelling is the main limiting factor in establishing the number of primary contexts.

3.2 Context based, parametric distribution modelling of prediction residuals

The use of parametric distribution for modelling the distribution of prediction residuals solves all difficulties induced by the large size of the alphabet, but raises the question of how much is lost by using a conditioning distribution less flexible than the histogram. Fortunately, with parametric modelling of residuals, the number of contexts can be increased significantly and the method will still be practical. There will be a clear loss compared to the use of histogram tracking for the same number of contexts, but the latter requires more memory which makes it nonpractical.

Selecting as parametric distribution the one-sided geometric distribution (OSG) makes the encoding very fast, since the optimal code for this distribution is known[2] to be the Golomb-Rice code (which is a special class of Huffman codes). We note that a similar encoding technique was used in [4], but without context modelling and with a different prediction technique. Many variations of Golomb-Rice codes have been introduced in a series of papers, associated with the selection of the LOCO-I [8] algorithm as the new JPEG-LS standard for lossless image compression.

We have used a invertible mapping for transforming the residuals in each context into new residuals, ε', which have distributions close to the one sided geometrical distribution. If the set of the original samples is $\{0, \alpha - 1\}$, the dynamic range for prediction residuals is $\{-(\alpha - 1), \alpha - 1\}$, but can be reduced to $\{-(\frac{\alpha}{2}), \frac{\alpha}{2} - 1\}$ [8] by a first invertible mapping. Finally, for converting the remapped errors ε^1 to non-negative integers ε' (having distribution close to OSG) we apply the mapping (1) which interleaves negative values and positive values in the sequence $0, -1, 1, -2, 2, ...$

Golomb-Rice coding of ε' is very fast, consisting in sending the last k bits of ε' followed by the unary representation of $\lfloor \frac{\varepsilon'}{2^k} \rfloor$. The unary representation is terminated with a 0 bit to allow uniquely decoding resulting in a total number of bits $\lfloor \frac{\varepsilon'}{2^k} \rfloor + k + 1$.

The value of the parameter k can be computed in each context given the sufficient statistic A_{con} (the sum of absolute values of all previous errors in the respective context) using $k_{con} = \lfloor \log_2 \frac{A_{con}}{N_{con}} + 0.47 \rfloor$[8] and N_{con} represents how many times the context *con* was visited.

The parameters to be stored in each context are: N_{con}, A_{con} (previously defined), B_{con} (the sum of all ε^1, needed for prediction bias correction) and C_{con} (an integer correction variable, incremented when $\frac{B_{con}}{N_{con}} \geq 0.5$ and decremented when $\frac{B_{con}}{N_{con}} \leq -0.5$)[8].

4 EXPERIMENTAL RESULTS

In the following the proposed compression algorithms are referred to as FSM-L-HMB (FSM context, L-predictor, Histogram Modelling and Bucketing) and FSM-L-PD (FSM context, L-predictor, Parametric Distribution Estimation). Both algorithms have been implemented in C and all coding rates reported in the following are ratios of actual length of compressed file per number of samples in the file.

Six audio files (sampled at 48 kHz and A/D converted at 16 bits/sample) of different lengths (between 1.6 and 2.7 Mbytes) are considered in the experiments, the content of each file being suggested in the first column of Table 1.

First the effect of changing the predictor order was analyzed for both FSM-L HMB and FSM-L-PD algorithms (see Table 1). The results obtained with $N = 20$ are superior to the results obtained with $N = 10$ for all files, indicating that even larger predictor order may still improve the performance, and there is no need to use a order selection procedure (as used in [4]) since at $N = 20$ our linear prediction models are not yet overfitting. Therefore, mainly the complexity of implementation will dictate the order of the linear prediction.

The results of the proposed algorithm are compared with the results of three public domain data compression algorithms: the first two are standard UNIX programs, *compress* and *pack*, based on universal compression algorithms. The third compression procedure is SHORTEN algorithm[4], which is especially designed for speech and audio lossless compression and is used for the distributions of speech databases on CD-ROM. The method combines forward coding (of predictor coefficients, estimated and encoded at the beginning af a whole block of new data) and sequential coding, by sending sequentially the prediction residuals, modeled using a parametric distribution and encoded using Huffman coding. There are two options for the prediction stage, and we experimented with both: the first is to use a "polynomial predictor" computing the high order differences of the input samples, while the second option is the use of a linear predictor, whose N coefficients are sent along with the encoded error. The results of using Compress, Pack, SHORTEN with linear prediction and for "polynomial predictor" and the best of our algorithms are listed in Table 2.

The best of our new algorithms outperformed all other tested methods for the given audio files. Even if the experiments were limited, the nature of the audio material was quite diversified and therefore we expect the results of ranking with respect to other methods to be typical situations for audio and speech compression.

References

[1] P. Craven and M. Gerzon. Lossless coding for audio discs. *J. Audio. Eng. Soc.*, 44:706–720, Sept. 1996.

	FSM-L-HMB $M=12$ $N=10$	FSM-L-HMB $M=12$ $N=20$	FSM-L-HMB $M=14$ $N=20$	FSM-L-PD $M=16$ $N=10$	FSM-L PD $M=16$ $N=20$
arpeggio	7.915	7.819	7.484	7.224	**6.990**
castanets	8.143	7.936	8.646	7.640	**7.040**
male speech	7.547	7.539	7.128	6.872	**6.701**
bagpipe	9.028	8.850	8.551	8.835	**8.529**
xylophone	7.214	6.669	6.618	5.151	**4.776**
harmonica	7.613	7.502	7.273	7.447	**7.231**
AVERAGE	7.910	7.7192	7.6167	7.1948	**6.8778**

Table 1: Comparison of compression rates (bits/sample) for different parameters of the two proposed algorithms.

	FSM-L-PD $N_{con}=2500$ N=20	SHORTEN "Polynomial prediction"	SHORTEN LP $N=10$	Unix Compress	Unix Pack
arpeggio	**6.990**	7.465	7.649	13.045	13.078
castanets	**7.040**	8.145	7.773	14.051	12.654
male speech	**6.701**	7.078	7.566	14.620	13.269
bagpipe	**8.529**	10.118	9.520	15.591	14.013
xylophone	**4.776**	7.100	7.515	15.259	13.848
harmonica	**7.231**	8.198	8.094	13.719	13.163
AVERAGE	**6.8778**	8.017	8.020	14.381	13.338

Table 2: Comparison of compression rates (bits/sample) of the best method in Table 1 with other lossless compression methods (best and second best are framed)

[2] R.G. Gallager and D.C. Van Voorhis. Optimal source codes for geometrically distributed integer alphabets. *IEEE Transactions on Information Theory*, IT-21:228–230, Mar. 1975.

[3] J. Rissanen. A universal data compression system. *IEEE Transactions on Information Theory*, IT-29:656–664, Sept. 1983.

[4] T. Robinson. SHORTEN : Simple lossless and near-lossless waveform compression. Cambridge University Engineering Department, Dec. 1994. http://www.softsound.com/shortendownload.html

[5] S. Todd, G.G. Langdon, and J. Rissanen. Parameter reduction and context selection for compressing of grey-scale images. *IBM J. Res. Develop*, 29:188–193, Mar. 1985.

[6] I. Tăbuş and J. Astola. Adaptive Boolean predictive modelling with application to lossless image coding. In *SPIE - Statistical and Stochastic Methods for Image Processing II*, pages 234–245, San Diego, California, Jul. 1997.

[7] I. Tăbuş, J. Rissanen, and J. Astola. Adaptive L-predictors based on finite state machine context selection. In *Proc. ICIP'97 International Conference on Image Processing*, pages 401–404, Santa Barbara, California, Oct. 1997.

[8] M. Weinberg, G. Seroussi, and G. Sapiro. LOCO-I a low complexity, context-based, lossless image compression algorithm. In *Proc. DCC'96 Data compression conference*, pages 140–149, Snowbird, Utah, Mar. 1996.

[9] M. Weinberger, J. Rissanen, and R. Arps. Applications of universal context modeling to lossless compression of gray-scale images. *IEEE Transactions on Image Processing*, IP-5:575–586, Apr. 1996.

[10] M. Weinberger, J. Rissanen, and M. Feder. A universal finite memory source. *IEEE Transactions on Information Theory*, IT-3:643–652, May 1995.

[11] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23:337–343, 1977.