West Pomeranian University of Technology in Szczecin
www.zut.edu.pl

A high efficient cascade coder with predictor blending method for lossless audio compression

Grzegorz Ulacha, Cezary Wernik
gulacha@wi.zut.edu.pl, cwernik@wi.zut.edu.pl

Faculty Of Computer Science And Information Technology
Żołnierska 49, Szczecin, 71-210, Poland, www.wi.zut.edu.pl

Introduction

The compression for the audio and video signal is getting increasing attention as the transmission bandwidth is getting wider, storage media is getting cheaper and a requirement for better quality is getting higher. This has been proven by the dynamic work of the MPEG Group operating from 1988 to today and its impressive portfolio [3]. Among the developed standards and technologies, the lossless compression algorithms deserve special attention, which enables the conversion of the raw data into a compressed form and reconversion without losing any information.

Our approach

Based on the idea of RLS-LMS cascade of predictors as in MPEG-4 lossless audio coder [1] we developed our own improved cascade of predictors with backward adaptation and new predictor blending method (see Fig. 1). In carried out experiment comparing the results with other known lossless audio coders, our method obtained the best efficiency (see Tab. 1) [2].

In lossless audio compression approach based on *r*-order linear prediction, to the output file the prediction errors being stored instead of the signal samples. In our solution the final prediction error is calculated using the new predictor blending method (1), where OLS+ and second stage of NLMS use inter-channel dependencies to calculate prediction coefficients. The final formula for prediction error calculation is as follows:

$$e(n) = x(n) - \left\lfloor \sum_{j=1}^{4} \alpha_j \cdot y_j(n) + C_{\mathrm{mix}} + 0.5 \right\rfloor = x(n) - \left\lfloor \frac{1}{\Psi} \sum_{j=1}^{4} c_j^{\mathrm{blend}} \cdot \varphi^{\tilde{e}_j} \cdot y_j(n) + C_{\mathrm{mix}} + 0.5 \right\rfloor, \quad (1)$$

where $\varphi = 0.96$, $c_j^{\mathrm{blend}} = \{1,1,1,4\}$, $\Psi = \sum_{j=1}^{4} c_j^{\mathrm{blend}} \cdot \varphi^{\tilde{e}_j}$ and $\tilde{e} = \frac{1}{n-1} \sum_{i=1}^{n-1} \left| e_j(i) \right|$.

In the next stages an own implementation of the effective context-dependent constant component removal method (CDCCR) is used. The last two blocks marked in Fig. 1 as Golomb Code and improved adaptive arithmetic coder (CABAC) are used to efficiently code prediction errors $e(n)$ into the resulting binary data stream [2].

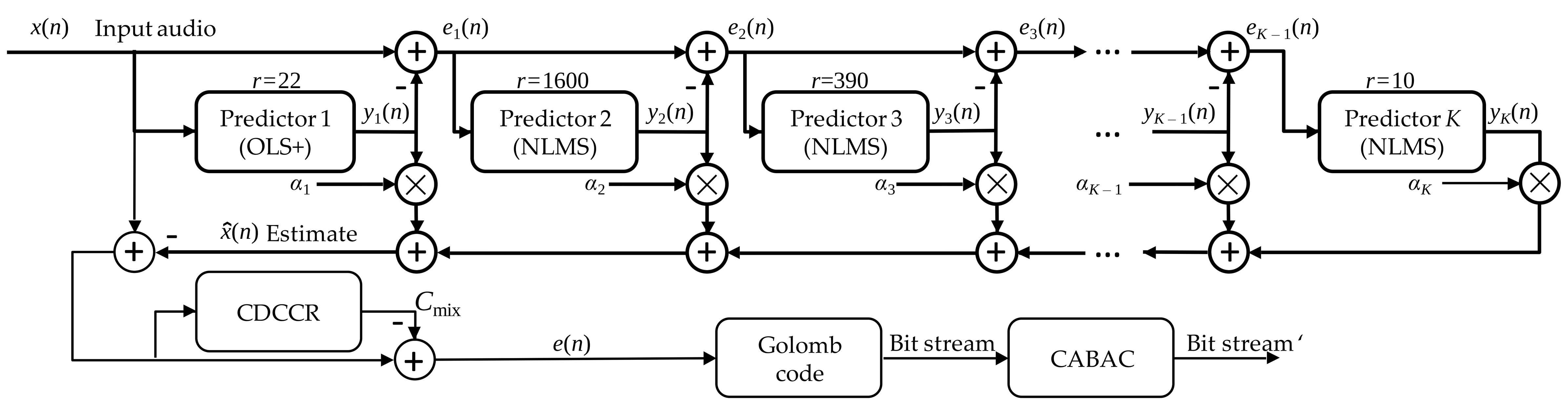

Figure 1: Schema of our approach for lossless audio coding.

Our approach is an extended version of the cascading audio data encoder. In classic solutions of lossless audio coding, adaptive versions of Rice and arithmetic encoders are used interchangeably. The proposed solution uses an adaptive Golomb code which is a generalised form of the Rice code with a potentially higher compression efficiency because of a better adaptation of the code to the current probability distribution of the currently encoding data. The Golomb codec output is additionally compressed using a context-dependent adaptive binary arithmetic encoder. Our proposition adapts better to the sequence of prediction errors characterised by the geometric distribution. A higher degree of compression was obtained due to a better algorithm for selection of context numbers, and also due to the omission a multi-valued arithmetic coder in favour a binary variant. This solution is more flexible also in comparison to classic solutions using the adaptive Rice code.

In the proposed solution, the increase in efficiency of compression compared to the most efficient option MP4-ALS-RM23 (working in backward adaptation mode) was possible due to introducing a more efficient OLS+ block in place of RLS, by improving block NLMS adding an additional CDCCR block in the cascaded predictive model and introducing an efficient CABAC based on initial compression using the adaptive Golomb code. The introduced improvements entail a disproportionately large increase in the implementation complexity relative to the reference version, for which was adopted MP4-ALS-RM23 in the best mode. Similar conclusions can also be drawn by comparing the reference version (the best mode) with the MP4-ALS-RM23 in default mode. Therefore, one should be aware that shortening the length of the result files for each subsequent percentage is paid by the increasing costs of the implementation complexity.

Table 1. The summary bit average of encoded test files [4] using different audio coders.

File	MP4 [1]	FLAC [2]	Monkey's Audio [3]	MP4 [4]	OLS-NLMS	Our proposition
ATrain	7.862	7.933	7.441	7.232	7.199	7.102
BeautySlept	10.049	10.005	8.826	8.305	8.491	8.213
Chanchan	10.160	10.127	9.938	9.886	9.746	9.704
death2	6.496	6.284	5.930	6.660	5.873	5.698
experiencia	11.377	11.371	11.029	10.992	10.911	10.859
female_spech	5.242	5.329	5.085	4.710	4.500	4.467
FloorEssence	10.202	10.174	9.750	9.509	9.355	9.228
ItCouldBeSweet	9.016	9.004	8.577	8.396	8.255	8.246
Layla	10.377	10.344	9.885	9.691	9.633	9.542
LifeShatters	11.182	11.146	10.874	10.836	10.828	10.779
Macabre	9.965	9.895	9.275	9.076	9.166	9.059
MaleSpeech	5.590	5.649	5.221	4.812	4.629	4.564
SinceAlways	11.164	11.211	10.539	10.473	10.394	10.358
thear1	11.742	11.746	11.504	11.425	11.435	11.395
TomsDiner	8.534	8.404	7.423	7.268	7.116	7.074
Velvet	10.672	10.679	10.508	10.212	10.029	9.973
Bit average	**9.352**	**9.331**	**8.863**	**8.718**	**8.597**	**8.516**

[1] MP4-ALS-RM23 in default mode, [2] FLAC in version 1.3.2, [3] Monkey's Audio in version 4.33, [4] MP4-ALS-RM23 in the best mode

References

[1] H. Huang et al., "Cascaded RLS-LMS prediction in MPEG-4 lossless audio coding," IEEE Trans. on Audio, Speech and Language Processing, vol. 16, No. 3, pp. 554-562, March 2008

[2] G. Ulacha, C. Wernik, "A High Efficiency Multistage Coder for Lossless Audio Compression using OLS+ and CDCCR Method," Applied Science, vol. 9, No. 23:5218, November 2019

[3] The Moving Picture Experts Group homepage. Available online: https://mpeg.chiariglione.org/ (accessed on 28 October 2019)

[4] Test database. Available online: http://www.rarewares.org/test_samples/ (accessed on 28 October 2019)