

# Entropy Coder for Audio Signals

Grzegorz Ulacha and Ryszard Stasinski

**Abstract**—In the paper an effective entropy coder designed for coding of prediction errors of audio signals is presented. The coder is implemented inside a greater structure, which signal modeling part is a lossless coding backward adaptation algorithm consisting of cascaded Ordinary Least Squares (OLS), three Normalized Least Mean Square (NLMS), and prediction error bias correction sections. The technique performance is compared to that of four other lossless codecs, including MPEG-4 Audio Lossless (ALS) one, and it is shown that indeed, on the average the new method is the best. The entropy coder is an advanced context adaptive Golomb one followed by two context adaptive arithmetic coders.

**Keywords**—lossless audio coding, entropy coding, OLS, NLMS, prediction error bias, Golomb code, arithmetic code

## I. INTRODUCTION

LOSSLESS audio coding is designed for applications in which preserving of original sound is important: for archiving of recordings, or when recording is intended for post-production (advertisements, radio or TV programs, videography, etc), as lossy decompression-compression cycles lead to its deterioration. It is also welcomed by audiophiles purchasing highest quality music on DVDs, Blu-Rays, or by Internet. The call for proposals for MPEG-4 Audio Lossless Coding (ALS) standard prompted intensive studies in this area, they took place in years 2002-2006 [1]. Apart of ALS they are also other lossless coding systems developed in that time, e.g. OptimFrog [2], and Monkey's Audio [3].

When introducing a new coder usually more attention is directed toward data modelling stage, it usually consists of predictors [4], [5], [6], but exceptions to that exist, too (see e.g. MPEG-4 Audio Scalable Lossless coding, SLS [7]). There are forward and backward predictor adaptation techniques, in both cases Minimum Mean Square Error (MMSE) is the optimization criterion. For improving results cascaded systems are constructed: a two stage forward adaptation algorithm is described in [8], in [1] a five stage backward adaptation one is presented. Another way leading to improvements in sound coding consists in exploiting inter-channel dependencies. For stereo systems coding gain depends on sound type and ranges from 0.5% up to 5% [9], on the average 3% gain can be expected [10].

Entropy coders in lossless audio systems are usually based on Golomb, Rice, or arithmetic coding schemes, and are rather simple. The described in this paper one is much more

sophisticated and forms the main contribution of the work, data modelling stage is the same as in [11]. Complication implies higher computational complexity, nevertheless, even this entropy coder has relatively small impact on the whole algorithm execution time. It can be then argued that the presented here technique has better balanced data modelling and entropy coding stages.

The entropy coder is constructed around an adaptive 9-class Golomb one, its output is then enhanced by two context adaptive arithmetic coders. The following section provides an outline of theoretical background, section II, the coder is introduced in section III. Description of signal modeling part of our algorithm is provided in section IV, it is a 5-stage structure, the first stage being the Ordinary Least Squares (OLS) predictor, then follows three-stage Normalized Least Mean Square (NLMS) one, finally the predictor error is tuned by removing the prediction error bias. As noted above the structure has been described in [11], the whole algorithm is summarized in Figure 1. Experimental results have been obtained by measuring bit rates necessary for coding 16 benchmark sound sequences [12] for the new coder and four other ones, section V. It can be seen that indeed, the new coder is the best, Table I.

## II. THEORETICAL BACKGROUND

In advanced multimedia compression techniques entropy coder is preceded by data modeling stage. The stage is intended for minimization of mutual information between input signal samples, as entropy coder efficiency is optimal when the mutual information is completely removed, in which case output coder bit rate reaches the entropy limit [13]. Data modeling algorithms are usually based on predictors, a linear predictor of rank  $r$  estimates sample  $x(n)$ :

$$\hat{x}(n) = \sum_{j=1}^r b_j \cdot x(n-j) \quad (1)$$

where  $x(n-j)$  are previous samples, and  $b_j$  are prediction coefficients [13]. Then prediction error is calculated (rounded up in lossless coding):

$$e(n) = x(n) - \hat{x}(n) \quad (2)$$

The simplest codecs use predictors with constant coefficients, nevertheless, much better results are obtained when predictors are optimized with respect to the coded signal. There are two classes of optimization algorithms: forward and backward ones. Powerful algorithms exist for both approaches, currently backward adaptation methods seem to be somewhat more efficient [1].

This work was partly supported by the Polish Ministry of Science and Higher Education in 2015.

Grzegorz Ulacha is with Department of Computer Science, West Pomeranian University of Technology, Zolnierska 49, 71-210 Szczecin, Poland, (e-mail: gulacha@wi.ps.pl).

Ryszard Stasinski is with Faculty of Electronics and Communications, Poznań University of Technology, Polanka 3, 60-965 Poznań, Poland, (e-mail: rstasins@et.put.poznan.pl).

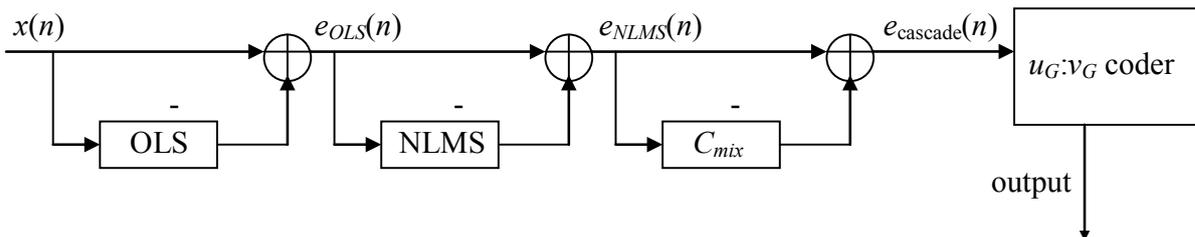


Fig. 1. Analyzed system: cascaded predictor of the OLS-NLMS algorithm followed by entropy codr. Note that NLMS predictor is in fact three-stage one.

TABLE I  
BIT RATE PER SAMPLE FOR 16 AUDIO SEQUENCES AND 5 LOSSLESS AUDIO CODECS

Sequences	RAR 5.0	Shorten 3.6.1 [5]	MP4-ALS-RM23 [16]	LA04b [27]	OLS-NLMS
ATrain	9.552	8.637	7.232	7.204	<b>7.199</b>
BeautySlept	10.771	10.724	<b>8.305</b>	8.318	8.491
chanchan	11.067	10.863	9.886	9.782	<b>9.746</b>
death2	8.825	7.152	6.660	5.907	<b>5.873</b>
experiencia	12.331	12.290	10.992	<b>10.908</b>	10.911
female_spech	8.731	7.539	4.710	5.301	<b>4.500</b>
FloorEssence	11.560	11.464	9.509	9.362	<b>9.355</b>
ItCouldBeSweet	11.510	11.587	8.396	8.591	<b>8.255</b>
Layla	11.207	10.871	9.691	<b>9.586</b>	9.633
LifeShatters	11.823	12.177	10.836	<b>10.777</b>	10.828
macabre	10.695	10.564	9.076	<b>9.096</b>	9.166
MaleSpeech	8.735	7.532	4.812	5.233	<b>4.629</b>
SinceAlways	12.265	12.192	10.473	10.404	<b>10.394</b>
thearl	12.285	12.574	11.425	<b>11.398</b>	11.435
TomsDiner	10.112	9.709	7.268	7.153	<b>7.116</b>
velvet	11.643	11.067	10.212	10.248	<b>10.029</b>
<b>Mean</b>	10.820	10.434	8.718	8.704	<b>8.597</b>

In both approaches mean-square error (MSE) is minimized, which implies that solution of Wiener-Hopf equations is estimated [13]:

$$\mathbf{w} = \mathbf{R}^{-1} \cdot \mathbf{p} \quad (3)$$

where  $\mathbf{R}$  and  $\mathbf{p}$  are signal autocorrelation matrix and vector, and  $\mathbf{w}$  is the vector of optimal predictor coefficients  $b_j$  (1). There are two main approaches to optimization of these coefficients: gradient, in practice some form of Least Mean Square (LMS) algorithm is implemented, and least squares (LS) minimization, the latter usually is based on recursive LS methods (RLS).

In MPEG-4 Audio Lossless (ALS) coder maximum frame length is  $N = 8192$ . This means that for high predictor ranks header information forms an important part of data stream. Note that in LMS and RLS algorithms predictor coefficients are not computed in advance, but from past data which are available both on the coder and decoder sides, hence, there is

no need for including them in coded data. That is why in this paper backward adaptation approach is implemented.

Prediction error tend to have Laplacian probability distribution, which means that it is natural to encode it using Golomb [14], or Rice entropy code [15]. Even better results are obtained when arithmetic coder is used, like the Gilbert-Moore one [16], or an advanced approach like that described in this paper.

### III. THE ENTROPY CODER

As stated above, prediction error of acoustic signal is usually coded using some extended version of Golomb, Rice, or arithmetic coder [16]. The described in this section entropy coder is constructed around an adaptive Golomb one. In Golomb coder a non-negative error sample is coded using two numbers:

$$u_G = \left\lfloor \frac{|e(n)|}{m} \right\rfloor \quad (4)$$

$$v_G = |e(n)| - u_G \cdot m$$

where optimal value of  $m$  depends on the variance of Laplacian probability distribution of coded error. Number  $u_G$  is unary, i.e. its value is equal to the number of zeros in a sequence ended with single one,  $v_G$  is usually coded using phase-in binary coder [17], in this paper a more advanced approach is implemented.

To further reduce coding redundancy in our algorithm numbers  $u_G$  and  $v_G$  are processed by separate adaptive arithmetic coders, more details on this can be found in consecutive sections. The proposed in this paper entropy coding scheme is more sophisticated than that presented in [14], nevertheless, total computational cost of the introduced in our paper system is only moderately affected by this fact. In contrast, reduction in final bit rate is clearly visible, if compared to more conservative approaches based on Rice, or single arithmetic coder [16].

### A. Estimation of Signal Probability Distribution

The use of Golomb code implies that the described below entropy coder is intended for coding of prediction error absolute values, their signs are processed separately. It is a well known fact that prediction error distribution is usually similar to Laplacian one, of course, here it is a one-sided distribution. Prediction error distribution is evaluated by error histogram, histogram slots are represented by entries of vector  $\mathbf{n}_e$ . At the algorithm start all histogram slots are set to the same small value. Then, when an error sample arrives, and its absolute prediction value is equal to  $\bar{e}(n)$ , vector position  $\mathbf{n}_e[\bar{e}(n)]$  is increased by one.

With growing numbers of counted in the histogram errors temporary variations of prediction error distribution have decreasing influence on obtained in this way distribution estimate. In this way tracking of signal non-stationarities is weak-end. To keep error counts not too high forgetting mechanism is implemented. Namely, a counter of coded error samples is introduced, every time its value reaches  $2^s$ , histogram entries are halved:

$$\mathbf{n}_e(i) := \left\lfloor \frac{\mathbf{n}_e(i)}{2} \right\rfloor + 1, \text{ for all } i \text{ from } 0 \text{ to } \bar{e}_{\max} \quad (5)$$

The new value of error samples counter is computed as the sum of all  $\mathbf{n}_e$  entries.

### B. Contexts and Classes

Best signal prediction is obtained when local signal characteristics are taken into consideration, hence, a good idea is to measure local signal parameters, then to have several predictors, and to use always the predictor designed for the current parameters values. This is the idea of context signal prediction. Namely, measured parameters space is divided into zones, and contexts are zone indices [18].

The same reasoning can be applied to entropy coding. Here predictors are replaced by error histograms. To keep track of local error properties some error parameters are measured, and coding is done on the basis of histogram pointed out by current parameter values. We are talking that the histogram is linked

with some error class. The number of classes should be chosen carefully: the greater the number, the better tracking of local error properties, but at the same time the slower convergence of at least some histograms, hence, longer period of suboptimal coding. In image coding they are examples of systems with 8 [18], 16, and even 20 classes [19], we decided to introduce here 9 ones.

The parameter chosen for defining class number is based on a similar idea to that in [20]. Namely, average of absolute values of few previous errors is considered:

$$\omega = \frac{5}{4z} \sum_{i=1}^z \frac{|e(n-i)|}{\sqrt{i}} \quad (6)$$

where value  $z = 17$  has been found experimentally. The parameter  $\omega$  is quantized using 8 thresholds:  $\mathbf{t}_h = \{4, 10, 30, 50, 80, 180, 500, 1100\}$ , the class number is the index of range defined by thresholds.

### C. Refinement of Golomb coder

Golomb code is optimal when signal probability distribution is geometric (Laplacian). In this paper 38 predefined variants of such probability distributions are used for each error class, section III.B. The laplacian distributions are given by the following formula:

$$G_j(i) = (1 - p_{(j)})p_{(j)}^i \quad (7)$$

$j = 0, 1, \dots, 37$ . Each probability  $p_{(j)}$  is linked with appropriate parameter  $m$  in (4), approximate formula linking  $m_{(j)}$  with  $p_{(j)}$  is:

$$p_{(j)} = 2^{-\left(\frac{1}{m_{(j)}}\right)}, \quad (8)$$

$m_{(j)}$  are from the experimentally found set  $\{1, 2, 4, 8, 12, 16, 20, 24, 32, 40, 48, 64, 80, 96, 128, 160, 192, 256, 320, 384, 448, 512, 576, 640, 768, 896, 1024, 1152, 1280, 1536, 1792, 2048, 2560, 3072, 4096, 5120, 6144, 8192\}$ .

The current  $m_{(j)}$  in (4) is chosen to minimize average cost of signal coding. The cost for  $(n+1)$ -st error sample is obtained from formula:

$$L_{cost(j)}(n+1) = f_{Golomb} \cdot L_{cost(j)}(n) + length_{(j)}(\bar{e}(n)) \quad (9)$$

where  $length_{(j)}(\bar{e}(n))$  is the amount of bits needed for coding absolute value of current error  $e(n)$ , and  $f_{Golomb}$  is a forgetting factor, its experimentally found value is 0.986. Its name is linked with the fact that when formula (9) is expanded, current cost appears to be the sum of previous ones multiplied by increasing powers of  $f_{Golomb}$ . The cost is computed for all 38 probability distributions of the currently used class, section III.B, the lowest one determines the choice of  $m_{(j)}$ . Note that they are 9 classes, hence, in total  $9 \cdot 38 = 342$  cost counters should be implemented.

The approach proposed in [16] (MPEG-4 ALS) is much simpler, where only one counter is used, and optimal Rice code parameter is obtained as a result of a simple function. Results provided in Table I show that this coder is inferior to that described in this paper.

#### D. Arithmetic Coders

There are two outputs from the Golomb code: unary number  $u_G$ , and binary one  $v_G$  (4). Then, if we want to improve the Golomb code, the numbers should be treated in different ways, as they are represented in different systems. In the paper the numbers are coded by separate context adaptive arithmetic coders. Note that described in the paper Golomb code provides additional information about prediction error: its class, section III.B, and what predefined error distribution was used for its coding, section III.C.

Number  $u_G$  is a unary number, hence, the first coder is a binary one. The number of contexts for this coder is  $38 \cdot 128 = 4864$ , which is the result of the following calculation: Each context is linked with one of 38 error probability distributions, section III.C. There are 9 error classes, section III.B, but here classes 7 and 8 are merged, so the next multiplier is 8, and not 9. In this way we are obtaining  $38 \cdot 8 = 304$  combinations. Further parameters determining number of contexts are as follows:

Definition of parameter  $k_{high}$  is:

$$k_{high} = \max\{|e(n-1)|, |e(n-2)|, |e(n-3)|, 0.6 \cdot |e(n-4)|\} \quad (10)$$

The parameter is quantized using one threshold having value 1500. In this way the number of combinations increases to 608.

The next parameter  $k_{one}$  is just the value represented by two last bits used in coding of previous  $u_G$ .

Then, the last parameter  $k_{last}$  is set to one if any of the following previous bits used for coding of  $u_G$  is one: -3, -4, -5, or -6.

Summarizing, the context number is computed from the following formula:

$$ctx_u = k_{best} \cdot (k_{class} \cdot 2^4 + k_{high} \cdot 2^3 + k_{one} \cdot 2^2 + k_{last}) \quad (11)$$

where  $k_{best}$  is the index of predefined error geometric distribution (7) determining the choice of  $m_{(j)}$  (8) that minimizes the cost function (9), and  $k_{class}$  is the index of class, section III.B, except that classes 7 and 8 are merged.

At start all 4864 binary histograms are set to 100 both for slots used for counting zeros and ones. The described in section III.A forgetting mechanism is activated when the number of counts reaches  $2^s = 2^{17}$  (similarly as in (5) histogram values are halved), value  $s = 17$  has been determined experimentally.

Contexts for coding of  $v_G$  are linked with 37 values of  $m_{(j)}$  enumerated in section III.C (note that for  $m_{(0)} = 1$   $v_G$  has only one value – zero, and need not be coded). Possible values of  $v_G$ , hence, slot indices of their histograms extend from 0 to  $m_{(j)} - 1$ ,  $j = 1, 2, \dots, 37$ . Initially histogram slots are set to integer number closest to:

$$p_{(j)}(i) = \frac{2^{-\frac{i}{m_{(j)}}}}{\sum_{i=0}^{m_{(j)}-1} 2^{-\frac{i}{m_{(j)}}}} \quad (12)$$

times  $2^s$ , where value  $s = 17$  has been found empirically, and index  $i$  has the same meaning as in (7). Note that (12) provides

precise relation between probability powers in (7) and (8). Also here the histogram forgetting mechanism described in section III.A is used.

#### IV. IMPLEMENTED SYSTEM

Tested in this paper lossless audio compression system is depicted in Figure 1. The proposed signal modeling part of the coder is a 5-stage structure, it was described for the first time in [11]. The idea is similar to that from [21], the enhancements in the data modelling stage are as follows: use of Ordinary Least Squares (OLS) predictor instead of Recursive Least Squares (RLS) one, there is no Differential Pulse Code Modulation (DPCM) at the beginning (OLS works equally well on quantized samples), and addition of prediction error bias correction. Similarly as in [21] Normalized Least Mean Square (NLMS) part consists of three cascaded NLMS predictors, the feature is not shown in Figure 1 for clarity sake. Ranks of NLMS predictors are  $\{1000, 25, 10\}$ . And, of course, there is the described in the previous section completely new entropy coder at the end.

Detailed description of signal modeling substructures can be found in the following subsections.

##### A. Inter-Channel Prediction

Such prediction can be done for multichannel audio systems, as dependencies between channels can be exploited for reducing coder bit rate. For used in experiments stereo signals it can be done as follows:

$$\hat{x}_L(n) = \sum_{i=1}^{r_L} a_i \cdot x_L(n-i) + \sum_{j=1}^{r_R} b_j \cdot x_R(n-j),$$

$$\hat{x}_R(n) = \sum_{j=1}^{r_L} b_j \cdot x_R(n-j) + \sum_{i=0}^{r_R-1} a_i \cdot x_L(n-i) \quad (13)$$

In the above formula  $r_R$  denotes rank of predictor for channel coded second rather than for right channel. Equally well this can be the left one, as  $L$  indexes in fact channel which is coded first. It was found experimentally that optimal proportion of predictor ranks  $r_L/r_R$  was often close to 2:1 [22].

##### B. OLS Method

The name OLS, or Ordinary Least-Squares is used for direct LS method, i.e. that in which predictor coefficients  $\mathbf{w}(n)$  are not computed recursively, compare (3):

$$\mathbf{w}(n) = \mathbf{R}^{-1}(n) \cdot \mathbf{p}(n) \quad (14)$$

where  $\mathbf{R}(n)$  is an estimator of signal autocorrelation matrix and  $\mathbf{p}(n)$  is an estimator of autocorrelation vector. Their elements are computed as follows:

$$R_{(j,i)}(n+1) = f_f \cdot R_{(j,i)}(n) + x(n-i) \cdot x(n-j) \quad (15)$$

$$p_{(j)}(n+1) = f_f \cdot p_{(j)}(n) + x(n) \cdot x(n-j)$$

and  $f_f$  is a forgetting factor (gradually diminishes older estimators in (15)), its value has been found experimentally, and is equal to 0.998. Initially matrix  $\mathbf{R}(n)$  and vector  $\mathbf{p}(n)$  are set to zero, hence, inversion of  $\mathbf{R}(n)$  should start for  $n$  much higher than the predictor rank  $r$  (matrix should be well-defined). In our case it is the 100-th iteration, as the optimal

rank of OLS predictor appears to be  $r = 20$  ( $r_L = 10$ ,  $r_R = 10$ ). Previously, the method was successfully applied to lossless coding of images [23], [24].

Computational complexity of OLS is clearly higher than that of any RLS algorithm, nevertheless, the used here predictor ranks are much smaller than those of the following NLMS filters, so this is not of great importance for the complexity of the whole method. The advantage of using OLS consists not only in minimization of problems with numerical stability plaguing RLS approach, OLS output usually has somewhat smaller entropy than that of RLS, hence, better data compression is obtained.

### C. NLMS Adaptive Filter

In the stochastic gradient-descent algorithm filter coefficients are updated as follows:

$$\mathbf{w}_{NLMS}(n+1) = \mathbf{w}_{NLMS}(n) + \mu(n) \cdot e(n) \cdot \mathbf{x}(n) \quad (16)$$

where  $\mathbf{w}(n)$  is the vector of filter coefficients,  $\mathbf{x}(n)$  is data vector containing last  $r$  signal samples, properties of the filter (stability, convergence rate) are determined by learning coefficient  $\mu(n)$ , and  $e(n)$  is filter prediction error. In the case of Normalized Least Mean Square (NLMS) algorithm the learning coefficient is controlled by temporary averaged signal power:

$$\mu(n) = \frac{\mu}{1 + \sum_{i=0}^{r-1} x^2(n-i)} \quad (17)$$

$\mu$  is a constant and  $x(n-i)$  are elements of  $\mathbf{x}(n)$ . We use a somewhat more complicated formula on updating filter coefficients, with decreasing weights for older signal samples (here input data vector is  $\mathbf{e}_{OLS}(n)$ ):

$$\mathbf{w}_{NLMS}(n+1) = \mathbf{w}_{NLMS}(n) + \mu(n) \cdot e(n) \cdot \mathbf{C} \cdot \mathbf{e}_{OLS}(n) \quad (18)$$

where  $\mathbf{C}$  is a diagonal matrix having elements  $\text{diag}\{0.995^1, 0.995^2, \dots, 0.995^r\}$  [25]. Also formula on learning coefficient  $\mu(n)$  is here more complicated than in classical NLMS, compare (17):

$$\mu(n) = \frac{\mu \cdot \sum_{i=0}^{r-1} \frac{1}{(i+1)^{0.8}}}{1 + \sum_{i=0}^{r-1} \frac{e_{OLS}^2(n-i)}{(i+1)^{0.8}}} \quad (19)$$

note forgetting  $1/\text{factor } (i+1)^{0.8}$  which diminishes older error samples at the output of predictor from the previous stage. The scaling factor  $1/(i+1)^{0.8}$ , multiplier 0.995 in matrix  $\mathbf{C}$ , and constant  $\mu = 2^{-8}$  (appropriate for 16-bit samples) have been found empirically.

### D. Prediction Error Bias Correction

Some prediction methods tend to accumulate DC prediction error component (prediction error bias). Up to now the phenomenon has been taken into account mainly in image lossless coding methods, see e.g. [26]. Prediction error bias cancellation methods are based on context idea, i.e. for each type of sample neighborhood a different correction value is remembered and updated. Digits of the number representing context are determined by properties of neighbor samples: their variance level, if the previous error is "high" or "low",

sample differences etc. Note that  $c$  such conditions result in  $2^c$  contexts.

In this paper a two-level prediction error bias correction approach is implemented: 12 different bias correction methods are defined, the final correction is obtained by summing up their results with appropriate weights. Only one of four implemented here context determining methods will be presented: it consists in constructing context 10-bit number in the following way (i.e. 1024 contexts are defined): Three first bits are just decisions if samples  $x(n-i) > \hat{x}(n)$  (current prediction), for  $i = 1, 2, 3$  (yes - 1, no - 0). The next three are obtained from conditions  $2x(n-i) - x(n-i-1) > \hat{x}(n)$  for  $i = 1, 2, 3$ . Next two conditions are  $e(n-i) > 0$  for  $i = 1, 2$  (concern prediction error). Finally, a number

$$x_{ave}(n) = \frac{1}{5} \sum_{i=1}^5 |x(n-i) - \hat{x}(n)| \quad (20)$$

is quantized using 4 experimentally found thresholds  $\{50, 250, 700\}$ , ranges defined by thresholds provide two last bits of context number.

It has been observed that the technique results in bit rate reduction up to 1%.

## V. EXPERIMENTS

The described here new algorithm has been used for coding 16 sound samples from the base [12]. These are rather short sequences containing samples of different kind of music and speech. Results are compared to those for other widely used audio codecs, and to those of RAR 5.0, being a general-purpose archiving program, Table I. As can be seen, indeed, in all cases specialized audio codecs are better than RAR 5.0, and the best of all is the new OLS-NLMS algorithm, best results for each audio sequence are in bold.

The presented in the paper entropy coder is rather complicated, nevertheless, its complexity is well matched to that of the implemented system. Namely, its execution time forms 26.9% of the time for the whole coder, its tenfold reduction would result in roughly 24% reduction of coding method computational complexity. This is not a great deal, especially that entropy coder performance do have influence on the technique efficiency. Definitely, this is not an option for simplest possible lossless audio coding systems.

Proper evaluation of the whole coder complexity is rather difficult: RAR 5.0, and MP4-ALS-RM23 are highly optimized software packages not using floating-point arithmetic, which is common for experimental programs. Code for other methods was not written by authors, and execution times were tested on different equipment. It can be only stated that the proposed method has improved performance at the cost of increased computational complexity.

## VI. CONCLUSION

In the paper structure of a new sophisticated entropy coder for audio signals is presented. The coder consists of context adaptive Golomb one, outputs of which are coded by two adaptive context arithmetic coders. The coder has been applied for compressing prediction error from a lossless signal

modeling stage consisting of OLS algorithm, cascaded with three-stage NLMS predictor, then followed by prediction error bias removal stage [11], Figure 1. The resultant technique is a better audio lossless coding method than those reported in the literature, Table I.

#### REFERENCES

- [1] H. Huang, P. Frnti, D. Huang, S. Rahardja, "Cascaded RLS-LMS prediction in MPEG-4 lossless audio coding," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 16, no. 3, 2008, pp. 554-562.
- [2] "Ghido's Data Compression Page." Internet: <http://www.losslessaudio.org/>, Feb 12, 2011.
- [3] "Monkey's Audio." Internet: <http://www.monkeysaudio.com/>, Jan. 20, 2013.
- [4] C. D. Giurcaneau, I. Tabus, J. Astola, "Adaptive context based sequential prediction for lossless audio compression," *Proc. of IX European Signal Processing Conference, EUSIPCO 1998*, vol. 4, 1998, pp. 2349-2352.
- [5] T. Robinson, "SHORTEN: Simple lossless and near-lossless waveform compression," Cambridge Univ. Eng. Dept., Cambridge, UK, Tech. Rep. 156, 1994, pp. 1-17.
- [6] E. Ravelli, P. Gournay, R. Lefebvre, "A Two-Stage MLP+NLMS Lossless coder for stereo audio," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, vol. 5, 2006, pp. V\_177-180.
- [7] R. Yu, S. Rahardja, C. C. Ko, H. Huang, "Improving coding efficiency for MPEG-4 Audio Scalable Lossless coding," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, vol. 3, 2005, pp. III\_169-172.
- [8] Lin Xiao, Li Gang, Li Zhengguo, Chia Thien King, Yoh Ai Ling, "A novel prediction scheme for lossless compression of audio waveform," *Proc. of International Conference on Multimedia and Expo (ICME 2001)*, 2001, pp. 197-201.
- [9] T. Moriya, D. Yang, and T. Liebchen, "Extended Linear Prediction Tools for Lossless Audio Coding," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, vol. 3, 2004, pp. III\_1008-1011.
- [10] T. Liebchen, Y. A. Reznik, "Improved Forward-Adaptive Prediction for MPEG-4 Audio Lossless Coding," in *118th AES Convention*, 28-31 May 2005, Barcelona, Spain, pp. 1-10.
- [11] G. Ulacha, R. Stasinski, "Lossless OLS-NLMS Algorithm for Audio Coding," *Proc. International Conference on Signals and Electronic Systems, ICSES 2014*, Proc. on USB, September 2014.
- [12] [http://www.rarewares.org/test\\_samples/](http://www.rarewares.org/test_samples/)
- [13] K. Sayood, "Introduction to Data Compression," 2nd edition, Morgan Kaufmann Publ., 2002.
- [14] S. W. Golomb, "Run-length encoding," *IEEE Transactions on Information Theory*, July 1966, vol. 12, pp. 399-401.
- [15] R. F. Rice, "Some practical universal noiseless coding techniques," Jet Propulsion Laboratory, JPL Publication 79-22, Pasadena, CA, March 1979.
- [16] Y.A. Reznik, "Coding of prediction residual in MPEG-4 standard for lossless audio coding (MPEG-4 ALS)," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, Montreal, Quebec, Canada, 17-21 May 2004, vol. 3, pp. III\_1024-1027.
- [17] D. Salomon, "Data compression. The complete reference," 3rd ed., New York, Springer-Verlag 2004.
- [18] X. Wu, N. D. Memon, "CALIC - A Context Based Adaptive Lossless Image Coding Scheme," *IEEE Trans. on Communications*, May 1996, vol. 45, pp. 437-444.
- [19] G. Deng, H. Ye, "Lossless image compression using adaptive predictor combination, symbol mapping and context filtering," *Proc. of IEEE 1999 International Conference on Image Process.*, Kobe, Japan, Oct. 1999, vol. 4, pp. 63-67.
- [20] I. Matsuda, N. Ozaki, Y. Umez, S. Itoh, "Lossless coding using Variable Blok-Size adaptive prediction optimized for each image," *Proc. 13th European Signal Processing Conference, EUSIPCO-05 CD*, Sept. 2005.
- [21] H. Huang, S. Rahardja, X. Lin, R. Yu, P. Frnti, "Cascaded RLS-LMS prediction in MPEG-4 lossless audio coding," *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, 2006, pp. V-181-184.
- [22] G. Ulacha, R. Stasinski, "Novel Ideas for Lossless Audio Coding," *Journal of Electronics and Telecommunications*, 2013, vol. 59, no. 1, pp. 17-23.
- [23] G. Ulacha, R. Stasinski, "Performance Optimized Predictor Blending Technique For Lossless Image Coding," *Proc. of the 36th International Conference on Acoustics, Speech and Signal Processing, ICASSP'11*, May 22-27, 2011, Prague, Czech Republic, pp. 1541-1544.
- [24] H. Ye, G. Deng, J. C. Devlin, "Adaptive linear prediction for lossless coding of greyscale images," in *Proc. IEEE International Conference on Image Processing, ICIP 2000, CDROM*, Vancouver, Canada, Sept. 2000.
- [25] Makino S., Kaneda Y., Koizumi N., "Exponentially weighted stepsize NLMS adaptive filter based on the statistics of a room impulse response," *IEEE Trans. on Speech and Audio Processing*, Vol. 1, No. 1, 1993, pp. 101-108.
- [26] G. Ulacha, R. Stasinski, "Context based lossless coder based on RLS predictor adaptation scheme," *Proc. of International Conference on Image Processing, ICIP 2009*, 7-11 November, Egypt, Cairo 2009, pp. 1917-1920.
- [27] <http://www.lossless-audio.com/download.htm>.