INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2013, VOL. 59, NO. 1, PP. 17–23
Manuscript received January 10, 2013; revised March, 2013. DOI: 10.2478/eletel-2013-0002

Novel Ideas for Lossless Audio Coding

Grzegorz Ulacha and Ryszard Stasiński

***Abstract*—Novel ideas for lossless audio coding analyzed in the paper are linked with forward predictor adaptation, and concern optimization of predictors on the basis of zero-order entropy and MMAE criterions, and context sound coding. Direct use of the former criterion is linked with exponential growth of optimization procedure, hence, a suboptimal algorithm having polynomial complexity is proposed. It is shown that on average the new types of predictors are better than those obtained by MMSE technique, while two- and three context systems are on average better than a single predictor one. It also appears that 7-bit PARCOR coefficients in the MPEG-4 ALS standard have insufficient precision for some predictor length, and that for very long frames coding results improve with the predictor rank practically in unlimited way.**

***Keywords*—Lossless audio coding, context coding, LS predictors, MMSE, MMAE, zero-order entropy.**

I. Introduction

LOSSLESS audio coding is used in such applications as archiving of recordings, distribution of highest quality music on disks, or by Internet. It is also useful when recording sound material intended for post-production (advertisements, radio or TV programs, videography, etc), namely, numerous lossy decompression-compression cycles gradually deteriorate it. The most intensive period of research in this domain was linked with call for proposals for MPEG-4 Audio Lossless Coding (ALS) standard in years 2002-2006 [1]. They are also interesting alternative lossless coding systems from that time e.g. OptimFrog [2], and Monkey's Audio [3].

Modeling stage of lossless audio coding algorithms is usually based on predictors [4]–[6], but there exist systems based on DCT (MPEG-4 SLS [7]), or wavelet transforms. Prediction methods can be divided into those with forward and backward predictor adaptation. Good results are obtained when cascading stages of prediction: two stage forward adaptation technique can be found in [8], backward adaptation using cascaded RLS and LMS predictors is presented in [9], in [1] the number of stages reaches five. Further improvement can be obtained by exploiting multichannel dependencies. The coding gain of stereo systems depends on sound characteristics and varies from 0.5% up to 5% [10].

In the paper fundamental properties of forward adaptation prediction systems are analyzed. In section IV performance of non-MMSE optimized predictors is tested. In section V context coding is introduced. Devoted to theoretical background sections II and III contain analyses of predictor length impact, and its coefficient precision on system performance. The conclusion is that indeed, there is still a lot of room for improvement of sound lossless coding systems.

II. Theoretical Background

In modern lossless coding techniques multimedia signal processing begins with data modeling intended for minimization of mutual information between signal samples. If the mutual information is completely removed, then the bit rate at the output of entropy coders may reach the entropy limit [11]. Data modeling algorithms are usually based on predictors. A linear predictor of rank r estimates x_n sample:

$$\hat{x}_n = \sum_{j=1}^{r} b_j \cdot x(j) \tag{1}$$

$x(j)$ are previous sample values, b_j are prediction coefficients [11]. The estimate is used to calculate the prediction error (rounded up in lossless coding):

$$e_n = x_n - \hat{x}_n \tag{2}$$

Probability distribution of the error is usually close to Laplacian, which means that it can be effectively coded by easy to implement Golomb-Rice coder [11]. It forms a part of specially constructed for coding of audio signals Gilbert-Moore code, where it is combined with arithmetic one [12].

Before processing a sound recording is usually divided into frames. Frame length may vary from a few tens to several thousand samples. In MPEG-4 ALS the maximum frame length is bounded by the free access requirement, namely, audio data set apart by 500 ms should be decoded independently. This implies 24 000 samples for a 48 kHz signal. The requirement is important for live transmission, or when the sound material will undergo post-production. For other applications even longer frames can be established, and there are good reasons for that. Namely, for avoiding ill-condition solution of (4) it is important that the frame length N is much greater than predictor rank r. In fact, this is a general property of parametric signal modeling methods, whenever data sample count is restricted a problem arises what is "appropriate" rank of a signal model. Note that highly efficient lossless audio coding methods are based on large signal models, e.g. total number of parameters in the method from [1] is almost 400 (5 predictors, the longest consisting of 300 coefficients, weights of predictor outputs), and MPEG-4 ALS allows predictor ranks up to 1023. Results of our experiments also indicate that long predictors tend to be better than short ones, see Fig. 1. MPEG-4 ALS contains also frame

The work has been supported by the Polish NCN grant nr 4922/B/T02/2011/40.

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

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

length optimization procedure: frame may be halved, and coding results compared to those for the full frame. If the approach succeeds, one or both subframes can be halved, too, down to 1/32 part of the initial lengths.

Prediction coefficients can be fixed, but much better results are obtained when predictors are optimized for estimation of a particular signal. There are two major classes of optimization procedures, named forward and backward adaptation. In forward adaptation approach predictor coefficients are optimized after collecting frame samples, other optimized parameters are frame length, predictor rank, and possibly, number of bits for coefficients coding. The approach is asymmetric from the complexity point of view, decoding is much simpler. The term backward adaptation is used for denoting adaptive predictors (RLS, LMS). The latter techniques are symmetric (similar complexities of coder and decoder), on the other hand they do not require side information accompanying coded data (mainly coefficient values). Both approaches lead to powerful algorithms, currently backward adaptation methods seem to be somewhat more efficient [1].

Predictor coefficients are usually optimized using minimum mean-square error criterion (MMSE), some alternative approaches will be presented in the next section. MMSE theoretical formulation leads to two practical approaches to coefficient optimization: recursive corrections implied by the gradient of cost function, and least-squares (LS) optimization. The most widely used algorithm from the first class is LMS used in backward adaptation algorithms. LS ap-proach can be both recursive (RLS), or non-recursive, and hence, used both in backward and forward adaptation techniques. In non-recursive case it consists in minimization of the sum of squares of the absolute error values, i.e. not mean but total square error is minimized:

$$\min_{b_j} \sum_{n=r}^{N} |e_n|^2 \tag{3}$$

Optimal formulae for calculating vector **B** of predictor coefficients are:

$$\mathbf{B} = \mathbf{R}^{-1} \cdot \mathbf{P}, \tag{4}$$

where **R** is the “experimental” signal autocorrelation matrix:

$$\mathbf{R}(j,i) = \sum_{n=r}^{N} x(n-i) \cdot x(n-j), \tag{5}$$

vector **P** is:

$$\mathbf{P}(j) = \sum_{n=r}^{N} x(n) \cdot x(n-j), \tag{6}$$

samples x are taken from a frame of size N. To avoid matrix inversion in (4) it is often assumed that matrix **R** is Toeplitz (i.e. it is fully defined by one-dimensional autocorrelation function), which allows the use of Levinson-Durbin algorithm for solving (4). Its computational complexity is $O(r^2)$, in contrast to $O(r^3)$ for direct approach. Another convenient solution is to use lattice implementation of the predictor, defined by reflection, or PARCOR coefficients [11]. Its advantage consists in fact that the lattice predictor implementation is usually more robust to rounding errors than the direct one, additionally, coefficient absolute values, if correctly computed, never exceed 1. Then, they can be easily coded using short bit representation, e.g. 7 bits in MPEG-4 ALS. Moreover, Golomb-Rice coding leads to reduction of this number to approximately 4 bits per coefficient [13].

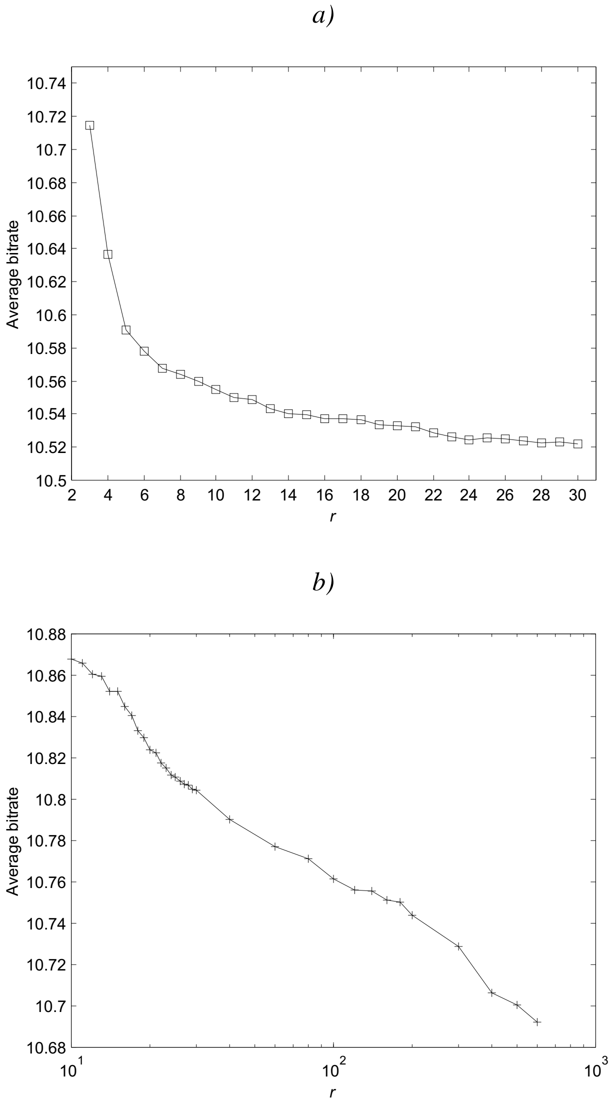

Fig. 1. Average coder bit rate (in bits per sample) as a function of predictor rank: a) for r up to 30, and b) for r from 10 to 600.

III. Optimization of Basic Predictor Properties

Considerations presented in the previous section prompted us to use very long frames in our experiments. More precisely, whole test recordings were processed, these were 16 sequences from the base [14]. Namely, our experiments aimed at introduction of some novel ideas to lossless audio coding, hence, the problem of “appropriate” model rank should not affect results. Secondly, sequences from [14] contain a rather homogenous sound material, in contrast to e.g. radio program, in which music interweaves with speech and “logo” signals, which fact suggests their statistical homogeneity. Finally, the longer the frame the smaller contribution of side information to coded data, as the formula on the coded data bit rate is:

$$L = H(S) + \frac{m \cdot r \cdot h}{N}, \tag{7}$$

where m is the number of bits used for coefficient representation, h is the number of contexts of the algorithm, in sections preceding section V: $h = 1$, and $H(S)$ is the measured data zero-order entropy:

$$H(S) = -\sum_{i=1}^{M} p_i \log p_i, \tag{8}$$

here p_i are probabilities of possible error values, M is their number. Unless stated otherwise the number of bits was not optimized, and was set to $m = 32$, but because of very large N values this had minimal impact on experiment results.

Occasional instability of Levinson-Durbin algorithm prompted us to analyze the problem of sufficient number of bits for coding of defining lattice structure reflection coefficients, signal processing results were compared to those for double precision calculations. In experiments *LifeShatters* from [4] was excluded, as results for it were clearly outliers. It appeared that 7-bit coding was sufficient for 13 from 15 recordings for predictor rank $r = 10$, for 7 recordings when the rank was $r = 30$, and only for 2 for $r = 100$. The problem was solved when numbers of bits were increased to 9, 11, or 15, respectively. Additionally, in the case of instability results preceding instable iteration by 3 steps were replacing the current ones. Longer representations of coefficients means greater side information, nevertheless, the findings suggest that results of MPEG-4 ALS can be improved by optimization of coefficients representation length, especially for long frames.

Another question that has been investigated was if there exists the globally optimal predictor length, or equivalently, if there exists a rank range above which extending predictor length is not practical. The answer was negative, at least for predictors of rank up to $r = 600$, see Fig. 1. This is in spite of excessive side information, as 32-bit representation of coefficients was assumed. Nevertheless, manifestation of the rule of diminishing returns is clearly visible, linear decrease of bit rate is obtained for exponential increase of predictor length. Moreover, when observing results for individual sequences it appears that for half of them the optimal predictor length is shorter than $r = 600$, the best results for individual recordings in Table I are in bold. In this situation a better solution is to implement shorter predictors matched to local properties of a signal, individual predictors optimized for collections of samples having similar properties can be used. This can be done by defining clusters of frames [15], or by implementing context coding, initial results showing advantages of the latter idea can be found in section V.

IV. Non-MMSE Optimized Predictors

The MMSE optimization (3) leads to linear predictors. A linear predictor de-correlates a signal to which it is matched, autocorrelation function of prediction error is the same as that of white noise, and white noise is a perfect example of memory-less data source. The idea works, but only partially. Namely, the consecutive samples of prediction error can be statistically independent only if a signal is fully described by second-order statistics. This is often not true. In this section we are addressing the problem of MMSE criterion sub-optimality by introducing a technique based on direct minimization of bit rate generated by a coder. We will also provide some results for Minimum Mean Absolute Error (MMAE) criterion.

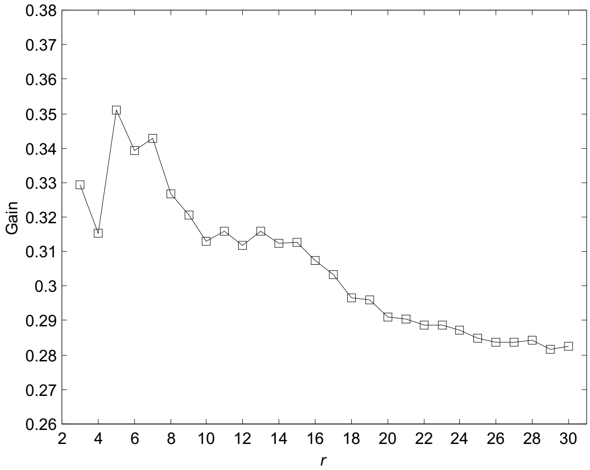

Fig. 2. Average gain in zero-order entropy when comparing iterative selective search approach with those based on MMSE.

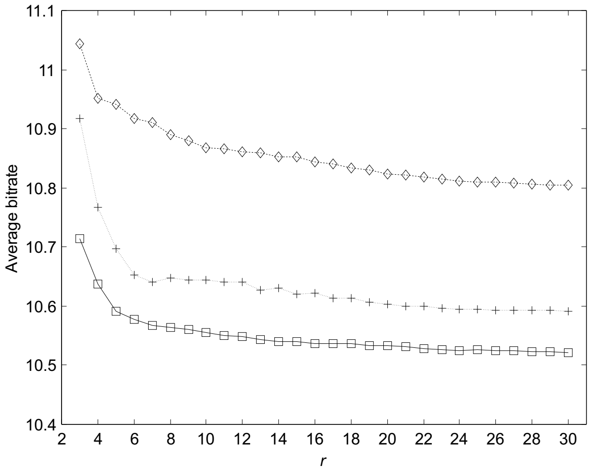

Fig. 3. Average coder bit rates (bits per sample) for predictors optimized using criteria: MMSE (upper curve), MMAE (middle curve), and iterative selective search (lower curve).

The search for the optimal signal model should be done by minimization of zero-order entropy of the model error (8). Namely, this is the lower bound for bit rate generated by an entropy coder processing that error [11]. Unfortunately, such model optimization can be done only by exhaustive search of possible model parameters, hence, its complexity grows exponentially with the model rank and number of bits used for its parameters representation.

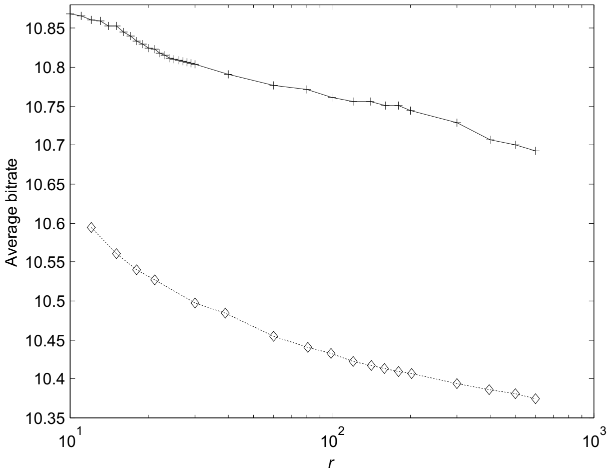

Fig. 4. Average coder bit rates (bits per sample) for single predictor (upper curve), and three contexts approach (lower curve), methods require same amount of side information.

We propose to reduce drastically the search space by the Iterative Selective Search algorithm for predictor coefficients, which gave surprisingly good results for lossless image coding in [16]. We start with coefficient vector $\mathbf{B} = [1, 0, \ldots, 0]$, and test optimality of its sum with each vector $\Delta\mathbf{B} = [\Delta b_1, \Delta b_2, \ldots, \Delta b_r]$, where only two Δb_j are non-zero integers having opposite values: -1, and 1, r is the predictor rank. Note that they are only $r(r-1)$ such vectors, which implies polynomial complexity of the method. The process is repeated a few times (in our experiments: 9). Then, in the next iteration $\Delta\mathbf{B}$ vectors are divided by 2, added to optimal $\mathbf{B}$ from the previous iteration and the sum tested for optimality. For iteration i:

$$\mathbf{B}(t+1) = \mathbf{B}(t) + 2^{-i} \cdot \Delta\mathbf{B}, \tag{9}$$

$i = 0, 1, 2, \ldots, m-3$, where m is the number of bits used for coefficient representation, as iterations are repeated, in general $t \neq i$. We have found that $m = 12$ is sufficient, which is only one bit more than for MMSE optimized predictors of rank $r = 30$, see previous section. Repetitions of iterations increase the maximum value of coefficients, here by two bits. Note that direct implementation of predictors is done.

The averaged difference in bit rate between the iterative selective search and MMSE-optimized predictors is shown in Fig. 2 for predictors ranks up to 30. The results have been obtained for sequences from the base [14], each was treated as a single frame. It can be seen that the difference is positive, i.e. on the average the new method is better than the traditional approach. In Fig. 3 the actual average bit rates for the methods are presented, additionally, results for MMAE-optimized predictors are shown. Results for the last approach lay between those for the preceding ones, similarly, its computational complexity is smaller than for iterative selective search, but greater than for MMSE. This means that MMAE criterion is worth considering when iterative selective search is too complex for an application.

Table II shows that in fact iterative selective search (ISS) is not always the best choice for coding of recordings from [14] (numbers in bold point out better results from the two approaches). Then, optimum is reached for a *hybrid* approach, when data is coded using both methods, and only the results for the better one are considered, see the last column of Table II.

V. Context Coding of Sound

Context coding is widely used in image processing [16]. It consists in implementing independent predictors for disjoint sets of samples. The predictor used for estimating current signal sample is chosen on the basis of context, i.e. a property of the sample neighborhood. The property is usually associated with a value, or range of values of some parameter. Context switching results in "missing data" in sample sequences used for calculating (4) and (5). It appears that due to these gaps in data the Levinson-Durbin algorithm tends to be unstable. The most obvious remedy to overcome this phenomenon is to compute predictor coefficient directly from formula (3), thus somewhat increasing computational complexity of the approach. Precision of predictor coefficients has not be optimized, 32 bits have been used for their representation.

In the experiments two or three contexts have been defined. Firstly, averaged differences of samples were computed:

$$S_{ave} = \frac{1}{N-1} \sum_{n=2}^{N} |x(n) - x(n-1)| \tag{10}$$

Contexts were computed from the parameter:

$$q = \max\{|x(n-j) - x(n-j-1)|\} \text{ for } j = 1, 2, \ldots, 9; \tag{11}$$

In the case of two contexts q was compared to a threshold $\alpha \cdot S_{ave}$, $\alpha = 2.5$. Indeed, the approach resulted in improvement of bit rate: for MMSE optimized single predictor of rank $r = 20$ and very long frames (whole sequences from [14]) the average bit rate was 10.82387, while for two context predictors of rank $r = 10$ (requiring the same amount of side information) it reduced to 10.56121. Even better results were obtained when the best from a set of thresholds was applied, the set of threshold multipliers was $\alpha_i = \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$. Then, the average bit rate reduced to 10.52956.

Context approach can be combined with iterative selective search, which results even in slight reduction of the method complexity, namely, the number of vectors $\Delta\mathbf{B}$ is $r \cdot h \cdot (r-1)$, only (and not $r \cdot h \cdot (r \cdot h - 1)$), where h is the number of contexts. Table III summarizes results obtained for context technique and both coefficient optimization algorithms for predictor ranks $r = 4, 10, 30$. On the average results for MMSE criterion are inferior to those for the iterative selective search (ISS), however, this need not to be true for some particular recordings (better results are in bold). Then, the table contains also bit rates for the hybrid approach for $r = 30$, in which results are taken always for the better of the two methods, last column of Table III.

Promising results for two-context coding prompted us to increase the number of contexts to three. In such a case two thresholds should be applied to q value (7), the optimal pair of S_{ave} multipliers was $\{\alpha_1; \alpha_2\} = \{1.5; 4\}$. This time much longer predictor ranks have been implemented: $r = 200$ for three contexts, and results of their processing were com-pared to those for two context predictors of rank 300 (the same context threshold as in previous experiments), and one predictor of rank 600 (note that all three approaches require the same amount of side information). Obviously, such big ISS predictors have not been tested, which is due to their much larger computational complexity. Results from this stage of experiments are reported in Table IV, as can be seen, average bit rate for three-context approach was only 10.413. Further reduction of bit rate has been obtained when the pair of thresholds has been chosen from the set obtained by multiplication of S_{ave} by multipliers from a set of pairs: $\{\alpha_{1i}; \alpha_{2i}\} = \{\{0.5; 4\}$, $\{1; 2.5\}$, $\{1.5; 4\}$, $\{1.5; 5.5\}$, $\{1.5; 6\}$, $\{1.5; 6.5\}$, $\{2; 4\}$, $\{2; 5\}\}$, average bit rate dropped to 10.39632. As can be seen from Table IV, results for three contexts are generally better than those for two contexts, but differences are not as big as between those for two contexts and single predictor.

Finally, efficiency of three-context approach has been tested for a wide range of predictor ranks, Fig. 4. The results have been compared to those for a single predictor, its rank reached value $r = 600$. Ranks of predictors in three-context method have been three times smaller than for the single predictor, in this way the amount of side information for both techniques was always the same. As previously, the results were averaged bit rates for 16 recordings from [14], each treated as a single frame. Higher efficiency of context approach is clearly demonstrated.

VI. Conclusions

It has been shown in the paper that a revision of basic techniques used in lossless sound coding leads to improvements. Firstly, it is shown that MMSE criterion need not lead to the best predictors, better ones can be obtained on the basis of MMAE, or suboptimal implementation of coder zero-order entropy minimization. Secondly, it appears that the widely used in image coding context approach works also in the audio coding domain. Finally it is suggested to optimize predictor coefficient precision, as their 7-bit representation in MPEG-4 ALS may be too short. An interesting observation is also the fact that for very long frames coder bit rate decreases with increase of predictor rank even for its extremely large values. The new proposals can be implemented in more advanced lossless audio coding systems, like multistage one described in the paper [8]. Moreover, context coding is also well suited for backward adaptation systems, the fact is widely exploited in image coding techniques [16].

References

[1] H. Huang, P. Fränti, D. Huang, and S. Rahardja, "Cascaded RLS-LMS prediction in MPEG-4 lossless audio coding," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 3, pp. 554–562, 2008.

[2] Http://www.losslessaudio.org/.

[3] Http://www.monkeysaudio.com/.

[4] C. D. Giurcaneau, I. Tabus, and J. Astola, "Adaptive context based sequential prediction for lossless audio compression," *Proceedings of IX European Signal Processing Conference EUSIPCO*, vol. 4, pp. 2349–2352, 1998.

[5] T. Robinson, "SHORTEN: Simple lossless and near-lossless waveform compression," Cambridge University, Engineering Department, Cambridge, UK, Tech. Rep. 156, 1994, pp. 1–17.

[6] E. Ravelli, P. Gournay, and R. Lefebvre, "A Two-Stage MLP+NLMS Lossless coder for ste-reo audio," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 177–180, 2006.

[7] R. Yu, S. Rahardja, C. C. Ko, and H. Huang, "Improving coding efficiency for MPEG-4 Audio Scalable Lossless coding," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 169–172, 2005.

[8] L. Xiao, L. Gang, L. Zhengguo, C. Thien King, and Y. Ai Ling, "A novel prediction scheme for lossless compression of audio waveform," in *International Conference on Multimedia and Expo*, 2001, pp. 197–201.

[9] R. Yu, C. Ko, S. Rahardja, and X. Lin, "An RLS-LMS algorithm for lossless audio coding," in *Conference Record of the Thirty-Seventh Asilomar Conference onSignals, Systems and Computers*, Pacific Grove, CA USA, 15–18 June 2003, pp. 300–303.

[10] T. Moriya, D. Yang, and T. Liebchen, "Extended Linear Prediction Tools for Lossless Audio Coding," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1008–1011, 2004.

[11] K. Sayood, *Introduction to Data Compression*, 2nd ed. Morgan Kaufmann Publishers, 2002.

[12] Y. A. Reznik, "Coding of prediction residual in MPEG-4 standard for lossless audio cod-ing (MPEG-4 ALS)," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1024–1027, 17–21 May 2004, Montreal, Quebec, Canada.

[13] T. Liebchen and Y. A. Reznik, "Improved Forward-Adaptive Prediction for MPEG-4 Audio Lossless Coding," in *118th AES Convention*, Barcelona, Spain, 28–31 May 2005, pp. 1–10.

[14] Http://studwww.ugent.be/ jdebock/lossless_audio_compression_test.htm.

[15] B. Aiazzi, S. Baronti, and L. Alparone, "Near-lossless image compression by relaxation-labeled prediction," *Signal Processing*, vol. 82, no. 11, pp. 1619–1631, 2002.

[16] G. Ulacha and R. Stasinski, "Paths to future image lossless coding," in *Proceedings of 54th International Symposium ELMAR*, 2012.

TABLE I
Bitrates for Coding of Test Recordings Using MMSE Predictors of Lenght R

Recordings $\setminus r$	80	100	120	160	200	300	400	500	600
ATrain	8.58021	8.57512	8.57503	8.56941	8.55971	8.53921	8.50841	8.48984	**8.47830**
BeautySlept	10.57049	10.53579	10.50733	10.46563	10.44068	10.31403	10.17328	10.10902	**10.04539**
chanchan	**10.77669**	10.78131	10.78346	10.78483	10.78852	10.79015	10.79518	10.79539	10.79826
death2	10.52375	10.45621	**10.44463**	10.45730	10.44889	10.46112	10.45999	10.47365	10.47433
experiencia	12.28065	12.28066	12.27852	12.27799	12.27611	12.27028	12.26758	**12.26846**	12.26870
female_speech	**9.08764**	9.09377	9.09704	9.11020	9.10818	9.11621	9.10295	9.12900	9.13827
FloorEssence	11.92437	11.92742	11.92380	11.92494	11.88216	11.86746	11.75648	11.73160	**11.67929**
ItCouldBeSweet	11.42029	11.42218	11.41209	11.40981	11.40558	11.40119	11.40129	**11.39997**	11.40019
Layla	11.28738	11.28295	11.27728	11.27071	11.26597	11.23992	11.22720	11.21460	**11.20194**
LifeShatters	11.35815	11.35620	11.35448	**11.35194**	11.35246	11.35316	11.35417	11.35607	11.35826
macabre	10.24626	10.22200	10.19925	10.16446	10.15247	10.11746	10.09494	10.08009	**10.06908**
male_speech	8.15885	**8.15828**	8.16103	8.16608	8.17045	8.16955	8.16503	8.17429	8.18880
SinceAlways	12.50596	12.50201	12.49724	12.48784	12.48117	12.47340	12.47063	12.46921	**12.46821**
thear1	12.15020	12.14384	12.14101	12.13624	12.13088	12.11980	12.11313	12.11025	**12.10544**
TomsDiner	9.83005	9.81760	**9.82232**	9.82881	9.82958	9.83003	9.83119	9.82829	9.83010
velvet	11.63510	11.63116	11.62351	11.61391	11.60931	11.59395	11.58246	11.57626	**11.57260**
Mean	10.77100	10.76166	10.75613	10.75126	10.74388	10.72856	10.70649	10.70037	10.69232

TABLE II
Bitrates for Coding of Test Recordings by Lossless Coding Algorithms Based on MMSE andD Zero-Order Entropy Criteria

Recordings	$r = 4$ MMSE	$r = 4$ ISS	$r = 10$ MMSE	$r = 10$ ISS	$r = 30$ MMSE	$r = 30$ ISS	$r = 30$ hybrid
ATrain	8.91167	8.79620	8.76946	8.65652	8.63126	**8.60030**	8.60030
BeautySlept	10.77671	10.77771	10.65276	10.77799	**10.61465**	10.77892	10.61465
chanchan	10.83554	10.75549	10.78105	10.69354	10.77273	**10.67699**	10.67699
death2	10.65435	7.67246	10.76663	7.62757	10.50570	**7.61280**	7.61280
experiencia	12.43585	12.39013	12.36401	12.30981	12.29114	**12.27914**	12.27914
female_speech	9.17665	8.40976	9.10551	8.39996	9.12055	**8.38485**	8.38485
FloorEssence	11.93834	11.71944	11.93762	11.69006	11.93354	**11.67771**	11.67771
ItCouldBeSweet	11.57115	11.53639	11.43563	11.39748	11.42475	**11.38397**	11.38397
Layla	11.53873	11.12507	11.46153	11.05162	11.35873	**11.01863**	11.01863
LifeShatters	11.74715	11.77339	11.49690	11.60983	**11.40618**	11.58364	11.40618
macabre	10.59138	10.55426	10.48036	10.44517	10.36933	**10.36095**	10.36095
male_speech	8.21611	8.05836	8.19195	8.02602	8.16954	**7.99309**	7.99309
SinceAlways	12.68133	12.54488	12.56598	12.43454	12.53041	**12.39162**	12.39162
thear1	12.31405	12.32544	12.18869	12.20509	**12.16881**	12.19133	12.16881
TomsDiner	10.03412	10.01663	9.96577	9.93011	**9.85311**	9.85813	9.85311
velvet	11.81099	11.73434	11.72198	11.62428	11.71904	**11.55450**	11.55450
Mean	10.95213	10.63687	10.86787	10.55497	10.80434	**10.52166**	10.49858

TABLE III
BITRATES FOR CODING OF TEST RECORDINGS BY CONTEXT LOSSLESS CODING ALGORITHMS DESCRIBED IN THE PAPER

Recordings	$r = 4$ MMSE	$r = 4$ ISS	$r = 10$ MMSE	$r = 10$ ISS	$r = 30$ MMSE	$r = 30$ ISS	$r = 30$ hybrid
ATrain	8.80231	8.78217	8.63898	8.62292	**8.49718**	8.59164	8.49718
BeautySlept	10.75315	10.77411	10.62730	10.69497	**10.58834**	10.67810	10.58834
chanchan	10.76229	10.75030	10.70422	10.68329	10.69596	**10.66451**	10.66451
death2	7.93517	7.58073	7.86730	7.50807	7.88867	**7.49779**	7.49779
experiencia	12.36798	12.37780	12.29399	12.29208	**12.21996**	12.27190	12.21996
female_speech	8.41155	8.33839	8.36915	8.30661	8.33745	**8.28207**	8.28207
FloorEssence	11.78775	11.69423	11.76905	11.64805	11.76125	**11.63595**	11.63595
ItCouldBeSweet	11.53763	11.52886	11.38934	11.38531	11.37762	**11.37288**	11.37288
Layla	11.11172	11.07128	11.03172	10.98918	**10.92846**	10.96726	10.92846
LifeShatters	11.73243	11.77542	11.48833	11.59230	**11.39835**	11.58469	11.39835
macabre	10.56913	10.54993	10.45326	10.44257	**10.32636**	10.37267	10.32636
male_speech	8.08683	8.02902	8.04872	8.01347	8.01976	**8.00076**	8.00076
SinceAlways	12.50905	12.43729	12.39446	12.31305	12.35412	**12.30428**	12.30428
thear1	12.29912	12.31615	12.16994	12.19706	**12.15053**	12.18588	12.15053
TomsDiner	10.01650	10.00734	9.91345	9.91728	**9.78393**	9.85293	9.78393
velvet	11.98682	11.69222	11.82014	11.59635	11.70327	**11.56743**	11.56743
Mean	10.66684	10.60658	10.56121	10.51266	10.50195	**10.48942**	10.45117

TABLE IV
BITRATES FOR CODING OF TEST RECORDINGS BY CONTEXT LOSSLESS CODING ALGORITHMS DESCRIBED IN THE PAPER

Recordings	r = 600 no context	r = 300 2 contexts, 1 threshold	r = 300 2 contexts, 1 of 8 thresholds	r = 200 3 contexts, 1 threshold pair	r = 200 3 contexts, 1 of 8 threshold pairs
ATrain	8.47830	8.40454	8.39622	8.40549	8.39570
BeautySlept	10.04539	10.27163	10.26925	10.41231	10.40850
chanchan	10.79826	10.72185	10.70160	10.68729	10.67666
death2	10.47433	7.92506	7.69276	7.65953	7.65932
experiencia	12.26870	12.21207	12.20447	12.20256	12.19955
female_speech	9.13827	8.31041	8.29872	8.22219	8.21096
FloorEssence	11.67929	11.66518	11.65198	11.62388	11.62388
ItCouldBeSweet	11.40019	11.34135	11.33980	11.33286	11.32874
Layla	11.20194	10.83219	10.82088	10.79939	10.79495
LifeShatters	11.35826	11.35419	11.35343	11.35632	11.35483
macabre	10.06908	10.03100	9.96025	9.99353	9.99334
male_speech	8.18880	8.02271	8.01902	7.98155	7.96473
SinceAlways	12.46821	12.31252	12.30736	12.30567	12.28545
thear1	12.10544	12.11046	12.11046	12.12290	12.11858
TomsDiner	9.83010	9.74519	9.74135	9.70890	9.69194
velvet	11.57260	11.64013	11.57596	11.79359	11.63399
Mean	10.69232	10.43128	10.40272	10.41300	10.39632