INTEGER WAVELET TRANSFORM BASED LOSSLESS AUDIO COMPRESSION

Ciprian Doru Giurcăneanu , Ioan Tăbuş and Jaakko Astola

Signal Processing Laboratory, Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland
e-mail: {cipriand,tabus,jta}@cs.tut.fi

ABSTRACT

In this paper we propose the use of *integer wavelet* [2] as a decorrelation stage for adaptive context based lossless audio coding. The original wideband audio signal is first decomposed in wavelet subbands. The resulted coefficients are integer valued and therefore can be transmitted using an adaptive context based method, in a lossless manner, the decoder being able to reconstruct them and afterwords to perfectly restore the audio waveform. Several ways to encode the integer wavelet coefficients are explored and the results are compared with those obtained in fullband contex adaptive coding.

1. INTRODUCTION

The integer wavelet transform (IWT) of an audio signal provides the decomposition of the original signal into a set of integer coefficients, from which by inverse wavelet transform the original signal can be recovered without any loss. The problem of encoding the original signal can therefore be transferred to encoding the wavelet coefficients, which provide a time-frequency description of the original signal. Encoding of wavelet coefficients instead of the original signal is very attractive, since the separation of the signal into a time-frequency representation allows to cope separately with different audio features and events. However, the task of making use efficiently of the time-frequency representation asks for different coding strategies in each subband, which leads to a difficult apriori modelling problem, and furthermore will increase the complexity of the overall scheme. Here we resort to adaptive encoding techniques, where no apriori information on the statistics of the coefficients in each band is needed. We will compare the results of adaptive encoding with an embedded encoding technique [7], which became the most used lossless encoding technique for wavelet coefficients. The sequences of coefficients produced by the wavelet transform still has some correlation between the coefficients at different decomposition levels and inside each band. This fact became apparent also in our experiments, where the zero order entropy of the coefficients resulted in a larger value than the actual rate obtained by adaptive techniques taking into account coefficient dependencies.

Since our focus is on different encoding techniques for the wavelet coefficients, we use a single wavelet transform, namely the Deslauriers-Dubuc symmetric biorthogonal interpolating wavelets $(4, \tilde{2})$ (the high pass analysis filter has 4 vanishing moments and the synthesis high pass filter filter has 2 vanishing moments). The impulse responses of the anlysis filters are

$$\begin{aligned} \underline{\tilde{h}} &= (\frac{1}{16}, 0, -\frac{9}{16}, 1, -\frac{9}{16}, 0, \frac{1}{16}) \\ \underline{\tilde{g}} &= (\frac{1}{64}, 0, -\frac{1}{8}, \frac{1}{4}, \frac{23}{32}, \frac{1}{4}, -\frac{1}{8}, 0, \frac{1}{64}) \qquad (1) \end{aligned}$$

We considered two different encoding methods for the IWT coefficients. In the first one contexts are used for adaptively collecting the statistics which are used in a Golomb-Rice coding scheme. The second one has the merit of implementing an embedded [7][8] encoder, which allows progressive transmission, at any fidelity level up to lossless quality, namely the method of set partitioning in hierarchical trees (SPIHT) proposed in [7] for image progressive transmission. The use of embedding in audio applications will become more and more important, since broadcasters can send an embedded stream which can be decoded up to the exact original, but if the receiver wants to stop earlier the reception he will still have a version of the whole file, with the least possible distortion at the given bit budget.

Experimental results compare the compressed size of several test files for the cases of different encoding methods, including fullband prediction [5].

2. INTEGER WAVELET DECOMPOSITION

The integer wavelet transform IWT $(4, \tilde{2})$ has the important property that IWT coefficients have the same dynamical range as the original signal (in our experiments 16 bits audio signal). These makes easier the implementation considerations regarding the size of the variables to be used and the ranges to provide for in the coding algorithm.

The Deslauriers-Dubuc symmetric biorthogonal interpolating wavelet $(4, \tilde{2})$ (1) has a lifting implementation amenable to a integer wavelet transform [1] by using the round-off function Int:

$$\begin{aligned} d_l^{\ell+1} &= s_{2l+1}^{\ell} - Int\left(\frac{9}{16}(s_{2l}^{\ell} + s_{2l+2}^{\ell}) - \right. \\ &\qquad \left. - \frac{1}{16}(s_{2l-2}^{\ell} + s_{2l+4}^{\ell})\right) \\ s_l^{\ell+1} &= s_{2l}^{\ell} + Int\left(\frac{1}{4}(d_{l-1}^{\ell+1} + d_l^{\ell+1})\right) \end{aligned} \qquad (2)$$

where the signals at the $\ell + 1$-th decomposition level are: s_i^{ℓ} - the input at time i, $d_l^{\ell+1}$ - the high frequency output, and $s_l^{\ell+1}$ - the low frequency output at time l.

The number of decomposition levels used e.g. in image coding, is usually less than 10. To avoid an excessive complexity of the coding scheme we selected and experimented here with $N_L = 3$ to 6 decomposition levels, the best results being obtained with 6 levels, which indicates that increasing even further the number of levels some improvements of the rate have to be expected, but they are somehow marginal compared to the cost incurred by the increase in complexity. We note also that the encoder delay increases with the number of decomposition levels.

Again reffering to wavelet based image coding, where the use of IWT was carrefully investigated, an important issue is the border information needed to perform the transform. In the case of audio signal, the bordering effects are not too critical, since they manifest only at the begining and end of the file, which leads to a marginal effect on the coding rate.

However, a proper initialization of the filters involved in the lifting steps is essential, since we perform frame-wise the transform and coding (to allow for a low coding delay). For the particular IWT we are using, we need to use the samples at the input of that decomposition level from time moments $t - 9, \ldots, t - 1$ in order to perform one level decomposition.

The original signal is frame-wise coded, the length of the frame being a power of two, 2^N (we experimented with N between 10 to 18). There is no overlapping between frames and we do not apply padding operation, each frame being treated separately during coding operation. The initialization mask needed is read from the end of the previous frame.

3. CONTEXT BASED CODING

The Context algorithm used in conjunction with arithmetic or Huffman coding proved to produce good results in lossless image and audio compression[10] when applied to the baseband original signal. In this section we describe a method to apply Context algorithm and Golomb-Rice codes to the transformed original signal in order to encode the wavelet coefficients resulting from the IWT decomposition of audio signal.

3.1. Golomb-Rice codes

The Golomb-Rice codes are very fast and efficient codes when applied to image and audio predictive lossless coding [5][9]. To encode an integer n, the Golomb code with parameter m first transmits $\lfloor n/m \rfloor$ as a unary code, then the value $(n \ mod \ m)$ is transmitted using a binary code. Rice coding is the particular case of Golomb coding when $m = 2^{\ell}$. With the Rice codes of parameter k, an integer n is coded by sending first the unary code of $\lfloor n/2^{\ell} \rfloor$ and then the k-bit representation of $(n \ mod \ 2^{\ell})$ is sent. If the distribution of the source is geometrical, $P(i) = (1 - \theta)\theta^i$, the Golomb code with the parameter $k = 2^{\lfloor \log_2 l \rfloor + 1} - l$ is optimal [4], where l is the integer for which the inequality $\theta^l + \theta^{l+1} \leq 1 < \theta^l + \theta^{l-1}$ holds.

The selection of the proper value of the parameter k will play an important role in the performance of Golomb-Rice codes. Two important estimation techniques have been previously considered: *block-oriented approach* and *sequential approach*. In the first case the data to be encoded are divided in blocks, a unique parameter is computed for each block and the parameter is transmitted as side information. In a *sequential approach* when encoding a data sample the encoder determines the code parameter from the data in a subset S of the past sample set. A good estimation of k is given by $k = \lceil log_2(E|s|) \rceil$ where the expectation is estimated using all the samples in the considered subset S [9]. We use a running mean of the absolute value $m_{|s|}$ instead of $E|s|$.

$$k = \lceil log_2(m_{|s|}) \rceil \qquad (3)$$

Two variables are necessary in each context for estimating the running mean: a counter $N_{|s|}$ which stores the cardinality of subset S and a variable $A_{|s|}$ accumulating the absolute values of the elements of S. To alleviate the effect of systematic bias we are using in each context two supplementary variables for ”centering” the distributions, as proposed in[9].

3.2. Wavelet coefficients modeling and context selection

Solely applying Golomb encoding to IWT coefficients does not produce a good result, mainly because we do not make use of dependencies between coefficients in the same level and across different decomposition levels. An improvement of coding rate can be obtained by applying the Context algorithm to make use of the above mentioned dependencies. Our approach is to cluster the coefficients in classes according to the statistics the most relevant for the coding length, which for Golomb-Rice codes is the running mean of the absolute value $m_{|s|}$. The contextual information is obtained from the past (i.e. those coefficients already available at the decoder) and we take advantage of the fact that the wavelet transforms localize signal energy in both frequency and spatial domains; large wavelet coefficients in different frequency subbands tend to be produced at the same spatial location.

Data and coefficient segments

At each level of wavelet decomposition the number of output coefficients is equal to the number of input samples (coefficients) and we arrange the coefficients obtained from the decomposition of a 2^N-length input segment in the following order: the positions from 1 to 2^{N-N_L} hold the coefficients from the N_L-th low-pas band s^{N_L} and continue in decreasing order of l ($N_L \geq l \geq 1$ to fill the positions $2^{N-l}+1$ to 2^{N-l+1} with the l-th high-pass band coefficients d^l. Such a 2^N-length segment of coefficients is treated by the coder as an independent entity.

Selection of context masks

The context for the current coefficient d^ℓ_l is choosen from two context masks: a principal context mask, C_M, contains the most recent N_T coefficients in the same frequency subband $C_M = \{d^\ell_{l-1}, \dots, d^\ell_{l-N_T}\}$ and a secondary context mask, C_{M_p}, contains the most recent N values of the *parent* coefficients $C_{M_p} = \{d^{\ell+1}_{\lfloor l/2 \rfloor -1}, \dots, d^{\ell+1}_{\lfloor l/2 \rfloor - N_T}\}$. At the finest resolution, when encoding s^{N_L} we use only a principal context mask $C_M = \{s^{N_L}_{l-1}, \dots, s^{N_L}_{l-N_T}\}$ and when encoding d^{N_L} we also use only the principal context. For each coefficient indexed with i, e.g. d^ℓ_i, its parent is the coefficient having the index $\lfloor \frac{i}{2} \rfloor$, e.g. $d^{\ell+1}_{\lfloor i/2 \rfloor}$.

Computation of context index

The primary context index is computed using the data in the primary context mask as follows:

$$con_1 = \left\lceil log_2 \frac{\sum_{c_i \in C_M} |c_i|}{N_T} \right\rceil \tag{4}$$

with the convention $log_2(0) = 0$ and a similar formula applies for the secondary context:

$$con_2 = \left\lceil log_2 \frac{\sum_{c_i \in C_{M_p}} |c_i|}{N_T} \right\rceil \tag{5}$$

The final context index is given by $con = con_1$ for the coefficients from the last decomposition level and by

$$con = con_1 + 16 con_2 \tag{6}$$

for the rest of coefficients (note there are 16 different contexts con_1). In this way the number of contexts is nearly 300. Since the number of contexts is not too large the problems related with *context dilution* are avoided.

Coefficient remapping and coding

Prior to encoding by a Golomb code, the coefficient d^ℓ (or s^{N_L}) should be remaped to a positive integer using the invertible mapping:

$$(d^\ell)' = \begin{cases} 2d^\ell & \text{if } d^\ell \geq 0 \\ 2|d^\ell| - 1 & \text{otherwise} \end{cases} \tag{7}$$

The wavelet coefficient can now be encoded with a Golomb code whose parameter k (3) is computed from the counters stored in the current context con.

Updating the parameters in the current context

After coding (decoding) the encoder (decoder) has to update the parameters in the current context con according to the value of the current wavelet coefficient. This is done by incrementing the visit counter $N_{|c|}(con) = N_{|c|}(con) + 1$ for the current context, updating the sum of absolute values $A_{|c|}(con) = A_{|c|}(con) + |(d^\ell)'|$ and updating the suplementary variables accounting for the bias. Note that the memory of the running counters is halved at regular intervals.

This stage completes one encoding loop. Next we are presenting the rationale behind our proposed selection of the context.

The heuristics of proposed context selection

A key issue in context selection [10] is that the distribution of wavelet coefficients in the context mask has to match the distribution of coefficients previously included in the selected class. We associate to each context (cluster) the Golomb-Rice code parameter (estimated from the past). We have to include a wavelet coefficient in that class which ensures the shortest code length for coding it.

For every positive integer i there is a unique positive integer p such that $2^{p-1} \leq i < 2^p$. We can evaluate the length of code, $CL(i)$, for encoding i using Golomb-Rice codes with different parameters k. We can observe in Table 1 that $k = p = \lceil \log_2 i \rceil$ is the optimum parameter and we can nottice what is the supplementary cost paid when the parameter k is not adequately choosen. Now with the context selection we have to estimate the value $\lceil \log_2 |d^\ell_l| \rceil$ for the proper selection of the coding parameter. A good estimate of $\lceil \log_2 |d^\ell_l| \rceil$ is $\lceil \log_2 m_{|s|}(con) \rceil$, which constitute the reason of the context selection procedure presented above.

Golomb-Rice with parameter k	Codelength $CL(i)$
$k > p$	$CL(i) \geq p + 1$
$k = p$	$CL(i) = p + 1$
$k = p - q$ for $q = 1, \ldots, p - 1$	$CL(i) \geq p + 1$

Table 1: The codelength $CL(i)$ of encoding the integer i when using a Golomb-Rice code of parameter k (where $p = \lceil \log_2 i \rceil$). The minimum codelength is given by the code with $k = p$

4. EMBEDED CODING

We have adopted an embeded coding algorithm which follows the procedure described by [7], but adapted it for 1-D signals. The main feature of the algorithm is the progressive transmission, the receiver having the possibility to decode the received signal at different fidelity levels until lossless decoding. In the first step the signal is segmented and decomposed by using IWT $(4, \tilde{2})$ transform as we described in Section 2. The modified variant of SPIHT is applied to segments of length 2^N of the original signal, the length of the segments affecting the overall coding rate obtained.

Embeded coding[7] is a greedy procedure based on two policies: (1) the coefficients should be transmitted in such a order that at the reconstructed signal the distortion is optimally decreased, the distortion becoming zero after receiving all coefficients. (2) the bits of each coefficient should also be transmitted in a progressive manner, the most importants being encoded first.

Therefore, in order to build a progressive datastream, the algorithm has to perform two important operations: ordering the coefficients by magnitude and transmitting the most significant bits first.

If we suppose the ordering is explicitly transmitted, the n-th iteration has two steps:

(1) **Sorting pass**: For all levels ℓ and all time instants $\eta_k$find the the coefficients $c^{\ell}_{\eta(k)}$ for which $2^n \leq c^{\ell}_{\eta(k)} < 2^{n+1}$ and output μ_n-the number of coefficients having this property, followed by the sample coordinates η_k and the sign for each coefficient;

(2) **Refinement pass**: output the nth most significant bit for all coefficients that had their coordinates transmitted in the previous sorting steps, in the same order used to send the coordinates.

In the algorithm we can replace the explicit transmitting of ordering with a set of comparisons that are performed in the same order at the coder and decoder. For an efficient implementation of comparisons a tree structure is used. Each node of the tree corresponds to a value from the segment containing the coefficients and is identified by its position in the buffer described in the subsection on data and coefficient segments. We adapted here the efficient implementation using lists [7].

5. EXPERIMENTAL RESULTS

The above presented coding algorithms were tested for six audio files sampled at 48 KHz and having different lengths from 1.4 to 2.7 Mbytes. The samples of the original audio files are represented using 16 bit unsigned integers.

The decomposition step is implemented by using IWT $(4, \tilde{2})$ wavelet. Different variants were tried for the total number of levels in the decomposition tree (3 to 6 levels). The resulting coefficients are recorded in 2^N-length segments (without overlapping) as described in Section 3.2, the experiments being performed with segment length from 2^{10} to 2^{18}.

The entropy of coefficients in each subband for every decomposition level is evaluated according to the frequency of occurence and the overall entropy is formed by properly weighting the entropies in each subband:

$$H = \frac{1}{2^{N_L}} H_{HP}^{N_L} + \sum_{l=1}^{N_L} \frac{1}{2^l} H_{LP}^{l} \qquad (8)$$

where H^l_{sb} denotes the entropy of wavelet coefficients at decomposition level l, sb corresponding to low-pass (LP) or high-pass (HP) subband and N_L is the number of decomposition levels. The entropy is compared with real code lengths obtained by using context based and then embedded coding (Table 2). The results are also compared with FSML-PD algorithm [5]. This algorithm performs context based coding of linear prediction errors. A different predictor is associated with each context.

The embedded coding offers the advantage of progressive transmission, but the compression ratios (computed for losslessly recovered signal) are generally worse in comparison with fullband context adaptive based algorithm. In the case of embedded coding we investigated the effect on coding rate of two important parameters:

- The length of segments 2^N, which also affects the decoding delay.
- The number of levels N_L of the wavelet decomposition which also affects the decoding delay, and furthermore the speed.

The tests were performed by considering segment lengths from 2^{10} to 2^{14} samples and decomposition levels from 3 to 6. We observe in Figure 1 that the code length is nearly independent of the segment length and a 6-level decomposition seems to be a good tradeoff between compression rate and complexity.

	Entropy	Embeded	Context based	FSML-PD
		coding	coding	
	$IWT(4,\tilde{2})$	$IWT(4,\tilde{2})$	$IWT(4,\tilde{2})$	
harpsichord	8.50	8.35	7.73	6.99
castanets	8.71	8.97	8.28	7.04
male speech	7.82	7.93	7.30	6.70
bagpipe	10.50	10.48	9.78	8.53
glockenspiel	8.02	8.31	7.21	4.78
pitchpipe	8.74	8.46	7.74	7.23
average	8.72	8.75	8.00	6.88

Table 2: Wavelets coefficient entropy and compression rates for SPIHT based and context based coding. All values are indicated in bits/sample.

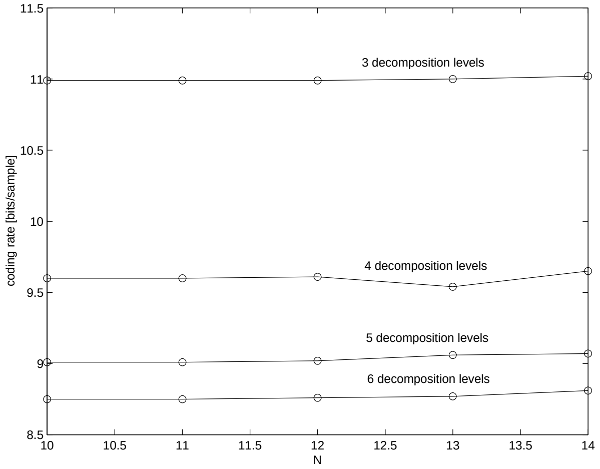

Figure 1: Code length as a function of segment length and number of wavelet decomposition levels

6. REFERENCES

[1] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Lossless image compression using integer to integer wavelet transforms. In *Proc. ICIP-97, IEEE International Conference on Image*, volume 1, pages 596–599, Santa Barbara, California, Oct. 1997.

[2] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal.*, 5(3):332–369, 1998.

[3] I. Daubechies and W. Sweldens. Factoring wavelet transform into lifting steps. Technical report, Bell Laboratories, Lucent Technologies, 1996.

[4] 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.

[5] C.D. Giurcăneanu, I. Tăbuş, and J. Astola. Adaptive context based sequential prediction for lossless audio compression. In *Proc. Eusipco-98, IX European Signal Processing Conference*, volume 4, pages 2349–2352, Rhodes, Greece, Sept. 1998.

[6] C.D. Giurcăneanu, I. Tăbuş, and J. Astola. Linear prediction from subbands for lossless audio compression. In *Proc. Norsig-98, 3rd IEEE Nordic Signal Processing Symposium*, pages 225–228, Vigso, Denmark, June 1998.

[7] A. Said and W.A. Pearlman. A new, fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Tr. on Circuits and Systems for Video Technology*, 6(3):243–250, June 1996.

[8] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41:3445–3462, Dec. 1993.

[9] 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.

[10] 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.