FAST LOW BIT RATE LATTICE ENTROPY CODING FOR SPEECH AND AUDIO CODING

Adriana Vasilache

Nokia Research Center, Visiokatu 1, 33720 Tampere, Finland
adriana.vasilache@nokia.com

ABSTRACT

In this work we study various methods for low bit rate encoding of long subvectors of transform coefficients from speech and audio codecs. Given the data characteristics, we propose a lattice based method that takes better into account the sparsity of the signal. We also propose two generic lattice encoding techniques addressing the encoding efficiency when encoding several lattice codevectors together and computational complexity of the lattice codebook nearest neighbor search. We examine as well the effect of different core codecs for speech and audio on the differential data characteristics and how the input signal type influences the coding efficiency of the considered methods.

1. INTRODUCTION

Encoding long transform coefficients is a common challenge for coding applications such as speech, audio, image, video coders. In the context of a large information amount to be encoded and transmitted, the low bit rate encoding is a fundamental aspect. Usually this data appears under the form of long vectors that need to be encoded at a fixed bit rate per frame or allowing small variations around an average bit rate value per frame. It can correspond directly to transform coefficients of the signal or to transform coefficients of a differential layer. The differential layer is usually the difference between the original signal and the reconstructed signal corresponding to a lower bit rate.

In [2] only parts of the long vector were encoded with non-null codevectors. Golay codewords, in addition to several outlier components, were used to encode the more significant part of the vector. The Golay codewords allowed only 0 and +/-1 amplitudes and only the outlier components added some more flexibility in the amplitude values, choice justified by the low bit rate argument. Alternatively, in this work we propose to use lattice codevectors, which allow for a broader domain of sample amplitudes. Lattice codewords will thus be used for a couple of selected subvectors and the higher bit rate range issued from the amplitudes choice will be compensated through variable bit rate coding, while keeping a constant bit rate for the long vector.

Lattice codewords have already been used in state of the art speech and audio coders [3], [4], [8]. We will also introduce in the present work two lattice quantization related techniques that improve the coding efficiency on one hand, and reduce the nearest neighbor search complexity on the other hand. Results of the two techniques inserted in the lattice entropy coding scheme will be presented for different data types.

The paper is structured as follows: after the introduction, the overall entropy lattice based encoding method is presented and the techniques addressing the coding efficiency when encoding several lattice codevectors at once and the low complexity nearest neighbor search are detailed. Experimental results with various artificial and real speech and audio transform data for several encoding methods are presented and compared.

2. METHOD DESCRIPTION

We propose a generic method for encoding long vectors at bit rates within 0.4-0.8 bits per sample (bps). By long vectors we refer to vectors with hundreds of components. Consider that *NB* bits are available for encoding a long vector of length *N*. The long vector is scaled by a scaling factor and divided into *D*-dimensional sub-vectors. Each sub-vector is quantized with a lattice quantizer. The information relative to the sub-vectors that are quantized to null vectors is represented by means of a binary string signaling the null subvectors. The rest of the bit rate is used for information related to the lattice codevectors. The lattice quantizer is considered as being a union of leader classes [5]. The number of bits to encode a non-null lattice codevector from a leader class is given by the sum between the number of bits to encode one codevector from the class and the number of bits to encode the index of the considered leader class [10]. An entropy code is used to encode the leader class index.

The encoding procedure has the following steps:

1. Estimate the overall gain based on the subvector energy distribution.
2. The vector is split in 10 dimensional subvectors which are to be lattice quantized. A set of leader vectors from the D_{10}^{+} lattice is considered for quantization.
3. The number of bits necessary for encoding is computed, knowing that the following entities are encoded:
   (a) Position of null subvectors
   (b) Indexes of leader vectors for each non-null subvector
   (c) Indexes of the lattice codevectors within the corresponding leader class for the non-null sub-vectors

As such, the method does not guarantee that the number of bits used is exactly *NB* or even that it is less than *NB*. Two cases can be differentiated here: when the number of bits is larger than the available number of bits and when the number of bits is lower than the available number of bits.

In case the number of bits is larger than allowed, a straightforward approach would be to define an order of the sub-vectors and to set to zero the sub-vectors for which there are not enough bits to encode the info relative to their corresponding lattice codevector and leader class. Even though the complexity of such an approach is reduced, a better and not much more complex way to gracefully degrade the quality of the quantization, is to use for some of the sub-vectors leader classes requiring less bits. During the search of the nearest neighbor for a given input vector, the distortion for all the leader classes is evaluated anyway, therefore it is pos-

© EURASIP, 2011 - ISSN 2076-1465

Sub-vector	Rate-distortion points					
	R	D	R	D	R	D
SV1	$R_{1,1}$	$D_{1,1}$	$R_{1,...}$	$D_{1,...}$	R_{1,N_1}	D_{1,N_1}
SV2	$R_{2,1}$	$D_{2,1}$	$R_{2,...}$	$D_{2,...}$	R_{2,N_2}	D_{2,N_2}
...	...	...	...	...	...	...
SVn	$R_{n,1}$	$D_{n,1}$	$R_{n,...}$	$D_{n,...}$	R_{n,N_n}	D_{n,N_n}

Table 1: Ordered rate-distortion points.

sible to construct a table with rate-distortion points like in Table 1. The distortion values for the considered input sub-vector SVi and the number of bits for each leader class are stored as $D_{i,j}$ and $R_{i,j}$ respectively.

For each sub vector, the rate distortion points are sorted in increasing order of the bitrate, such that the distortion of two successive points is decreasing. In other words if, for a point of higher bitrate, the distortion increases or stays the same with respect to the previous point in the list, the higher rate point is eliminated. Within this ordered list of points a gradient based search is performed starting from the lowest distortion, such that the available number of bits is not exceeded.

The gradient search is summarized by the following pseudo-code:

```
For i=1:n
    k(i) = Ni
End For
While (1)
    For i = 1:n
      Grad(i) = (Ri,k(i) -Ri,(k(i)-1))/
               (Di,(k(i)-1) - Di,k(i)));
    End For
    i* = arg(max(Grad));
    If k(i*) == 0
       Update bits_pos;
    End If
    R = R - Ri*, k(i*) + Ri*,(k(i*)-1);
    k(i*) = k(i*)-1;
    If R+bits_pos < NB Stop, Output k
End While
```

The general idea behind the algorithm above is that the selection of the subvector for which the bitrate is reduced is made such that the overall bit rate reduction over the objective quality loss (distortion increase) is maximized. In other words, the question is what is the way to reduce as much bitrate at once, while having the less distortion increase. The number of bits value that is checked against NB consists of the number of bits to encode the position of the significant sub-vectors, $bits\_pos$ and the number of bits to encode the leader class index and the lattice codevector index within the class, R.

In the second case, if the resulting number of bits is lower than NB one approach can be to change the initial scaling factor and redo the encoding. In a lower complexity case, the encoding can be left as it is. This is the approach taken in the experimental part.

2.1 Encoding the position of the null sub vectors

A binary string is formed with zero signaling a null sub vector and one a non-null subvector. A combined approach consisting of Huffman coding trained for the first order entropy and a run-length coded with Golomb Rice coding is used. In the run-length approach, if the last group of subvectors consists of null sub-vectors, the corresponding run can be omitted.

2.2 Encoding the significant sub vectors

The significant sub vectors, i.e. those that are encoded by a non null code vector, are represented by the leader vector index and the index of the lattice code vector. The leader vectors are ordered in decreasing order of probability of occurrence which allows for a very complexity efficient encoding of the leader index through Golomb Rice encoding of fixed parameter. Since the first part of the vector is most energetic and uses leaders of higher indexes and the second part of the vector contains lower leader indexes, two regions are defined in the vector containing the leader vector indexes and they are encoded with different fixed Golomb Rice parameters.

2.3 Low complexity nearest neighbor search

The lattice codebooks have inherently a lower complexity search procedure than the unstructured vector quantizers. We propose to reduce further the nearest neighbor (NN) search complexity for lattice codebooks expressed as unions of leader classes by combining the triangle inequality philosophy with the search on leaders [1].

The general use of the triangle inequality for fast search in a non-structured codebook is sketched in the following paragraph.

Consider x, the input point for which the nearest neighbor should be found in the codebook C. In a general full search procedure for a non structured codebook, the distortion between the input and each codevector from the codebook should be evaluated and the codevector having the lowest distortion is chosen as nearest neighbor. If the current codevector to calculate the distortion for is c_j and the best codevector so far is c_i the triangle inequality states that:

$$|\|c_i - x\| - \|c_j - c_i\|| \leq \|c_j - x\| \leq \|c_i - x\| + \|c_j - c_i\| \quad (1)$$

Consequently if

$$|\|c_j - c_i\| - \|c_i - x\|| \geq \|c_i - x\| \quad (2)$$

there is no need to calculate $\|c_j - x\|$ because it will be larger than the current best distortion. The test from Equation 2 needs the distances between each two codevectors. These distances can be precomputed and stored.

This being the general idea of the fast NN search based on the triangle inequality, it can be applied to a codebook formed as a union of leader classes. In this case the distortion between the input and the nearest neighbor from each leader class is computed by sorting the input vector and calculating the distortion between the sorted input vector and the leader vector of the considered leader class. If used as such, the fast search based on the triangle inequality condition would need the distances between any two codevectors from the lattice codebook to be precomputed and stored. The main idea is that only the distances between the leader vectors from each leader class two by two need to be precomputed and stored which reduces the number of vectors to be considered by a factor larger than 10. For instance for bitrate 1 bit per sample and vector dimension 10 there are 1024 lattice codevectors which can be covered typically [1] by less than hundred leader vectors. For higher dimensions the reduction factor is even larger. The fact that only the leader vectors should be considered when storing the paired distances comes from the fact that for each leader class the sorted input vectors are compared with the leader vectors.

After performing the NN search for each subvector from the long vector, the indexing of the lattice codevectors within the leader classes is performed using a binomial enumeration approach [6].

2.4 Encoding efficiency improvement

To counteract the fact that the lattice leader classes are not exact power of two, all the indexes, $I_i, i = 0 : n-1$ of the lattice codevectors corresponding to significant subvectors can be encoded in the same composed index

$$I = I_0 + N_0 I_1 + N_0 N_1 I_2 + \ldots + N_0 \ldots N_{n-2} I_{n-1}.$$

$N_i, i = 0, n-1$ are the number of codevectors in each leader class corresponding to the significant subvectors, and I_i are the corresponding codevector indexes. There are n significant subvectors in the long subvector.

The gain in bitrate comes from the difference

$$\sum_i \lceil \log_2 N_i \rceil - \lceil \sum_i \log_2 N_i \rceil.$$

3. EXPERIMENTAL RESULTS

In addition to the method proposed in this work, two more methods have been tested. The first one is using lattice quantizers of dimension 8 and it is based on the Voronoi extension of the rotated lattice E_8 [7], RE8. The second one is using binary Golay codes [2]. We will briefly describe the methods here.

3.1 Lattice RE8 based coding

Long input vectors $\mathbf{x} = [x_1 \ \ldots x_N]$ with N real valued entries need to be encoded at a given bit rate. The long vector is divided into 8-dimensional subvectors. Each subvector is quantized [7] in the lattice RE8 and represented as belonging to one of the predefined finite lattice codebooks or to the Voronoi extension of one of the predefined lattice codebooks. For each sub-vector the index of the codebook or extension is sent together with the lattice codevector index within that codebook.

3.2 Golay code based coding

Long input vectors $\mathbf{x} = [x_1 \ \ldots x_N]$ with N real valued entries need to be encoded at a given bit rate. The quantization is performed in two stages [2]. In the first stage some entries of the vector are identified as outliers and they are quantized and transmitted together with their number, n_o, and their locations in the vector. In the second stage the remaining entries of $\mathbf{x}$ are grouped as $n_b = \lfloor (N - n_o)/L \rfloor$ subvectors of length L. Some of the subvectors are encoded as full zero vectors, while the rest (e.g., a generic subvector $\mathbf{z} = [z_1, \ldots, z_L]^T$), are encoded using the following elements: (1) a Golay code word $\mathbf{c} \in \mathscr{G}_{23}$ for conveying the significance information ($c_i = 0$ signaling that z_i is quantized to zero); (2) the signs for all significant entries (for all i for which $c_i = 1$ a bit encodes $sign(z_i)$); and (3) an overall gain to be used with all quantized values in $\mathbf{x}$ during the reconstruction process. A mask $\{t_1, \ldots, t_{n_b}\}$ of n_b bits specifies which of the subvectors are encoded as full-zeros, and which of them are vector quantized.

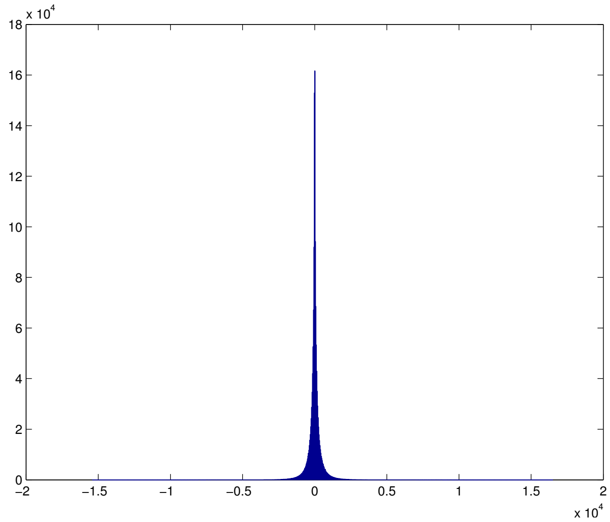

Figure 1: Example histogram of the experimental data.

3.3 New method

The lattice codebook is composed of the first 32 leader vectors of the pyramidal truncation of the lattice D_{10}^+. A pyramidal lattice truncation contains all the lattice points whose ℓ_1 norm is less or equal to a given value. When using the triangle inequality based search method and the null vector as reference for the condition 2, the complexity is reduced from 1.325 WMOPS (weighted milion operations per second - ITU-T Software Tool Library 2009 [9]) to 0.576 WMOPS. Updating the reference point to the current best reduces further the search complexity, but for the considered scenario, not only the nearest neighbor was needed, but also several close to nearest codevectors in order to adapt to the allowed number of bits.

Overall the complexity of the resulting method is 1.4 WMOPS compared to the 1.6 WMOPS for the RE8 based method and 4.8 WMOPS for the Golay based method.

The use of the composed index, I gives an average SNR (signal to noise ratio) increase of 0.1dB.

3.4 Numerical results

We have considered differential MDCT transform data with respect to AMR-WB codec. Wideband speech, noisy speech and music data has been considered. In Figure 1 the considered data histogram is presented. A very important peak in zero can be observed.

AMR-WB is considered as a core layer of an experimental coder and the differential layer is MDCT transformed and encoded in the transform domain. Under normal functioning of the codec, 144 bits are reserved per frame for the encoding of the transform coefficients. As we were also interested in comparing the methods under study on a more general level, we have considered a larger bit rate domain. Both real and artificial data are considered.

3.4.1 *Artificial data*

We have considered first artificial normal data. In Figure 2 the SNR values for the three methods together with the rate distortion limit are plotted. It can be observed that the Golay based method performs the best in this case. The size of the

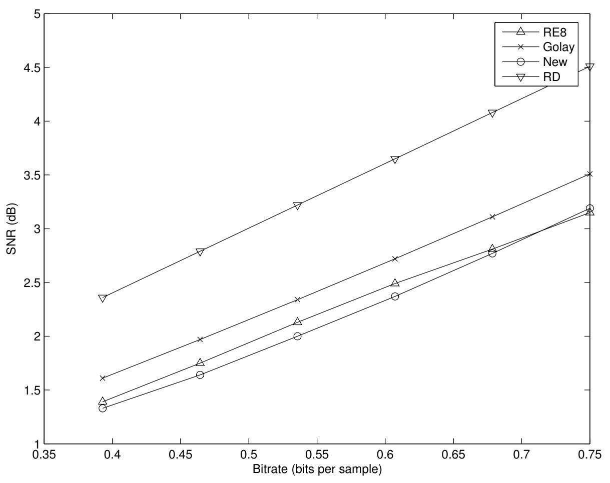

Figure 2: Method comparison for Gaussian data.

codevectors (23) is larger than for the other two methods. The coding efficiency of the Golay codewords given that at these bit rates and type of data the unitary amplitudes are not a big restriction is a second explanation of the good results.

3.4.2 Real data

We have considered first wideband speech. Several modes of AMR-WB have been used as core codecs.

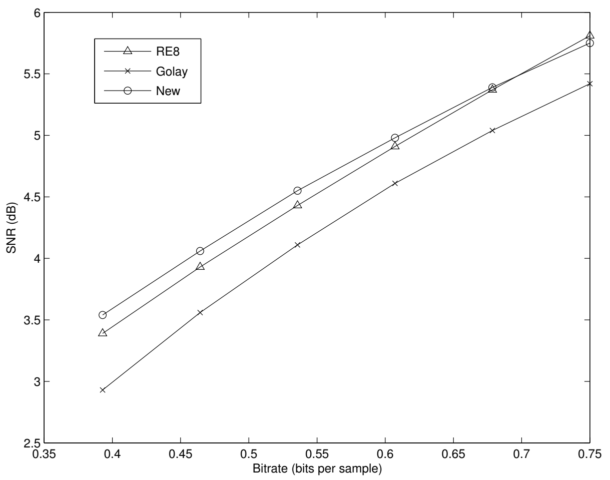

Figure 3: Method comparison for speech signal differential transform data. AMR-WB at 6.60 kbps is used as core codec.

The first case examined was the AMR-WB at 6.60 kbps. In this case the data is less evenly distributed than the Gaussian case (the pdf has longer tail, see Figure 1) and the lattice based methods outperform the Golay based coding approach (Figure 3). Here, the limitation to unitary amplitudes for most of the significant components in the transform vector, is clearly seen in the performance of the Golay based method. The more efficient coding of the zero regions for the proposed method is visible especially for the lower bit rates and it attenuates at higher bit rates, where there are less zero regions.

We also consider music signals, and the SNR for the three considered methods is plotted in Figure 4. Still the proposed method outperforms the other two ones. Comparing the results from Figure 3 and 4 it can be observed that overall the SNR values for music are overall lower than for the speech case. This can be explained by the different statistic of the data in the two cases and also confirmed by the fact that the average shape factor for speech is 0.76 while for music is 0.81.

Three more AMR-WB profiles have been considered, for speech signals and the resulting SNR values for RE8 based coding versus the proposed method are presented in Table 2. It can be seen that for all studied cases, the SNR for the proposed method is better than for the RE8 based method.

AMR-WB mode (kbps)	RE8 method SNR(dB)	New method SNR(dB)
6.60 kbps	3.95	4.07
8.85 kbps	3.52	3.69
12.65 kbps	3.47	3.53
14.25 kbps	3.49	3.55

Table 2: Method comparison with respect to SNR, for different modes of AMR-WB

It is interesting to compare the Golay based method and the RE8 based method, in the case of G.729.1 as core codec, and observe that preference is given to the Golay based method as seen in Figure 5 [2]. The average shape factor of the differential transform data in this case is 1.5 which is much closer to a Gaussian and in accordance with the results presented for the Gaussian data in Figure 2.

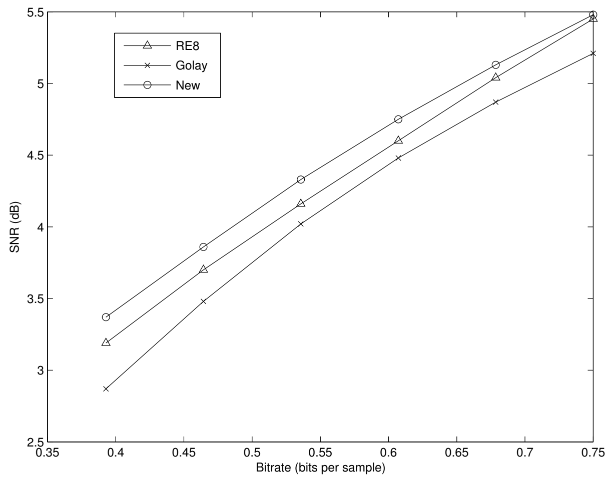

Figure 4: Method comparison for music signal differential transform data. AMR-WB at 6.60 kbps is used as core codec.

4. CONCLUSION

Together with the proposal of a low complexity coding method, we have conducted a study on the influence of the data type and core codec on the coding efficiency of the differential transform data.

The Golay based method has better performance at low bit rates for Gaussian or close to Gaussian data, case in which the limitation induced to the amplitude values is not critical

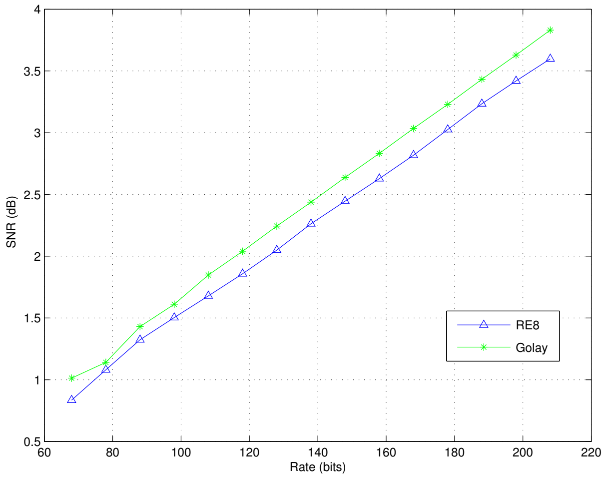

Figure 5: Method comparison for music signal differential transform data. G.729.1 at 12 kbps is used as core codec.

and the Golay codes efficiency is fully utilized. For data with longer tails and higher pick in zero the lattice based methods perform better. Out of the two studied lattice based methods, the proposed one, using entropy encoding of lattice D_{10}^{+} codevectors performs better having higher SNR and lower complexity. The low complexity has been enabled by a new fast search algorithm that can be used in lattice codebooks that are defined as union of leader classes.

As overall conclusion, it can be stated that in the coding of transform differential data, the choice of the core codec is of major importance in the statistical distribution of the data, and consequently of the coding method choice and results.

REFERENCES

[1] A. Vasilache, B. Dumitrescu, and I. Tabus, "Multiple-scale leader-lattice VQ with applications to LSF quantization," in *Signal Processing*, vol. 82(4):47-70, 2002.

[2] I. Tabus and A. Vasilache, "Low bit rate vector quantization of outlier contaminated data based on shells of Golay codes," in *Proceedings of Data Compression Conference*, Snowbird, USA, March 15-18 2009, pp. 302–311.

[3] T. Vaillancourt, M. Jelínek, A. E. Ertan, J. Stachurski, L. Laaksonen A. Rämö, J. Gibbs, U. Mittal, S. Bruhn, V. Grancharov, M. Oshikiri, H. Ehara, D. Zhan, F. Ma, D. Virette, and S. Ragot, "ITU-T EV-VBR: A robust 8-32 kbit/s scalable coder for error prone telecommunications channels," in *Proceedings of EUSIPCO 2008*, Lausanne, Switzerland, August, 25-29 2008.

[4] S. Ragot, B. Kövesi, R. Trilling, D. Virette, N. Duc, D. Massaloux, S. Proust, B. Geiser, M. Gartner, S. Schandl, H. Taddei, Y. Gao, E. Shlomot, H. Ehara, K. Yoshida, T. Vaillancourt, R. Salami, M. S. Lee, , and D. Y. Kim, "ITU-T G.729.1: An 8-32 kbits/s scalable coder interoperable with G.729 for wideband telephony and voice over IP," in *Proceedings of ICASSP 2007*, Honolulu, Hawaii, USA, April, 15-20 2007.

[5] J.-P. Adoul, "La quantification vectorielle des signaux: approche algébrique," *Annales des Télécommunications*, vol. 41, no. 3-4, pp. 158–177, 1986.

[6] A. Vasilache, *On vector quantization with regular geometric structure*, Ph.D. thesis, Tampere University of Technology, Tampere, Finland, 2003.

[7] S. Ragot, B. Bessette and R. Lefebvre, "Low-complexity multi-rate lattice vector quantization with application to wideband TCX speech coding at 32 kbit/s," in *Proceedings of ICASSP 2004*, Montreal, Canada, May, 17-21 2004, vol. 1.

[8] Minjie Xie, P. Chu, A. Taleb and M. Briand, "ITU-T G.719: A new low-complexity full-band (20 kHz) audio coding standard for high-quality conversational applications," in Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2009, pp.265-268, 18-21 October 2009.

[9] Recommendation ITU-T G.191, "Software tools for speech and audio coding standardization," March 2010.

[10] A. Vasilache and I. Tăbuş, "LSF quantization with MSLLVQ for transmission over noisy channels," in *Proceedings of EUSIPCO 2002*, Toulouse, France, September, 3-6 2002.