16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

# ENTROPIC ENCODING OF LATTICE CODEVECTORS BASED ON PRODUCT CODE INDEXING

*Adriana Vasilache*

Nokia Research Center, Media Laboratory
33720 Tampere, Finland
adriana.vasilache@nokia.com

## ABSTRACT

In this paper we introduce a new entropy encoding for lattice codevectors. The encoding method is based on the product code indexes of lattice codevectors and, in addition to a very efficient encoding (up to 30% bitrate saving with respect to the fixed rate encoding), it enables practical use of lattice entropy coding for higher dimensions.

## 1. INTRODUCTION

Lattice quantization has attracted interest of the lossless compression community, during the last years due to its reduced memory requirements for storage and low complexity encoding and decoding algorihms. State of the art speech codecs as AMR-WB+ [1] and G.EV-VBR[2] codec make use of its advantages. Audio coding methods based on lattice quantization have been as well proposed [3].

Most of the lattice based coding methods rely on fixed rate coding or on a semi-variable rate coding where the vector to be quantized is split in several sub-blocks for which the rate is variable, but the overall bit rate for the global vector is fixed [1]. There exist also variable rate encoding of lattice codevectors. Most of these methods rely on the grouping of codevectors on classes such as leader classes or shells [4], [5] or apply directly entropy coding methods to the lattice codevector components [6]. However, the former method becomes less practical when the number of classes increases (with the increase of the bit rate and for some of the truncation shapes), while the latter is from the start less efficient than a direct entropy coding of the lattice vectors indexes, but obviously less complex.

We propose in the present paper a new indexing method for lattice vectors that makes use of the product code indexing method. The proposed method is exemplified on rectagular truncation of lattices, where the number of leader classes is relatively high, but the shape of the truncation is successfully used in conjuction with companding.

The paper is presenting first several lattice definitions and terms, followed by a short description of the product code indexing that enables the key method of the work, the new entropy encoding of lattice vectors. The proposed method will be exemplified within an audio coding scheme that will be briefly presented prior to the final results and conclusions.

## 2. LATTICE QUANTIZATION: TERMINOLOGY AND DEFINITIONS

### 2.1 Lattice definition

Geometrically, a lattice is an infinite regular array of points which uniformly fills the n-dimensional space.

Algebraically, an n-dimensional lattice $\Lambda$ is a set of real vectors whose coordinates are integers in a given basis $\{b_i \in R^n\}_{i=1,n}$

$$\Lambda = \left\{ \mathbf{v} \in \mathbf{R}^n | \mathbf{v} = \sum_{i=1}^{n} \alpha_i b_i, \alpha_i \in \mathbf{Z} \right\}$$

When used as fixed rate quantizer a lattice should be truncated to a finite number of points corresponding to the selected bit rate. Even if, in principle, for the variable rate case, when entropy coding is applied, the lattice can be considered infinite, for practical reasons (i.e. indexing algorithms and numerical aspects of entropy coding), a finite support for the lattice should be specified.

### 2.2 Lattice truncation

Generally, the lattice support, or truncation is defined by means of a norm $N(\mathbf{x})$ of the lattice points which should be less than a given value $K$:

$$\Lambda_K = \left\{ (\mathbf{x}) = (x_1, x_2, \ldots, x_n) \in \Lambda \ \ |N(\mathbf{x}) \leq K \right\}$$

The truncation shape is spherical if $N$ is the Euclidean norm, or pyramidal if the $N$ is $l_1$, or rectangular if $N$ is the maximum norm i.e. the maximum absolute value of the lattice vector components.

A generalization of the rectangular truncation is the truncation having different maximum absolute norms, $\{K_i\}_{i=1:n}$ along different dimensions.

$$\Lambda_{K_i} = \left\{ (\mathbf{x}) = (x_1, x_2, \ldots, x_n) \in \Lambda \ \ ||x_i| \leq K_i \right\}. \qquad (1)$$

The generalization is exemplified in Fig. 1 for the lattice $Z_2$ with $K_1 = 3$ and $K_2 = 2$. The truncation includes all $Z_2$ points inside the smaller rectangle as well as the points from the border.

A given norm defines, in addition to the lattice truncation, the lattice shell, as the set of lattice points that have the same norm value, $K$:

$$\Lambda_K = \left\{ (\mathbf{x}) = (x_1, x_2, \ldots, x_n) \in \Lambda \ \ |N(\mathbf{x}) = K \right\}.$$

Consequently, the lattice truncation can be seen as a union of lattice shells.

A division of the lattice into even finer sets is obtained starting from the definition of a leader vector and that of a leader class. A leader vector is a positive integer vector $\mathbf{v} = (v_m, ..., v_m, ..., v_i, ..., v_i, ..., v_1, ..., v_1)$ where $0 \leq v_1 < ... < v_i < ... < v_m$. The leader class of the leader vector $\mathbf{v}$ is the set of

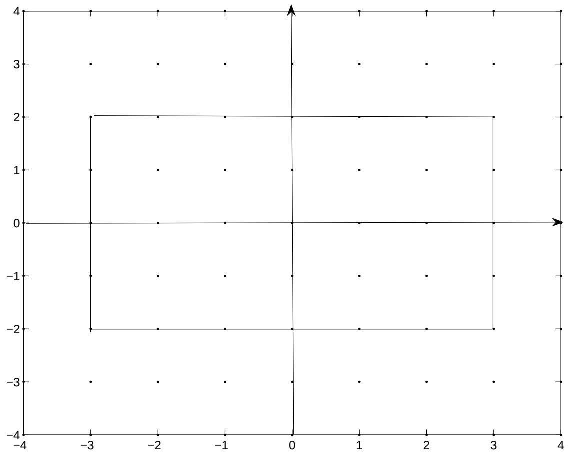

Figure 1: Illustration of the generalized rectangular truncation of $Z_2$.

all the vectors obtained through signed permutations, with some possible constraints, of the vector $\mathbf{v}$. The leader class notion has been proposed originally in [8], [9].

Most of the lattices used for quantization can be defined as union of leader classes [13].

### 2.3 Lattice entropy coding

Ideally, the entropy coding of lattice codevectors should consider each codevector individually. However, the use of lattice codebooks is most usefull for high dimensions, where even for bit rates relatively small, the number of codevectors easily becomes large, making the individual consideration of each codevector impractical. Practical solutions to this problem have been the grouping of codevectors into sets (i.e. shells or leader classes) and entropy encode the index of the set while the vector index within the set is encoded using enumerative coding [5].

Another approach has been to entropy encode the lattice vector components [6], but for lattices where there exist constraints relative to the values of a lattice vector (e.g. sum of components should be even) this approach is not very efficient with respect to the entropy coding of the lattice vector indexes.

Figures 2 and 3 show rate-distortion curves for the lattices $D_4$ and $D_8$ respectively. The curve marked as "comp" corresponds to the case when the components are supposed to be entropy encoded, while the curve marked with "idx" corresponds to the case when the codevector indexes are supposed to be entropy encoded. The rate is assimilated to the entropy, to consider the best achievable case and the entropy values are estimated from the data. Gaussian data with unitary variance is used for test. Also the curve corresponding to the $Z_4/Z_8$ lattice is depicted in the graphs, and as expected, for this lattice the rate-distortion curves are the same whether the entropy coding is applied to the components or to the indexes.

## 3. ENTROPY CODING BASED ON PRODUCT CODE LATTICE CODEVECTOR INDEXING

Many lattice indexing algorithms existing in the literature are based on the definition of the lattice as a union of leader

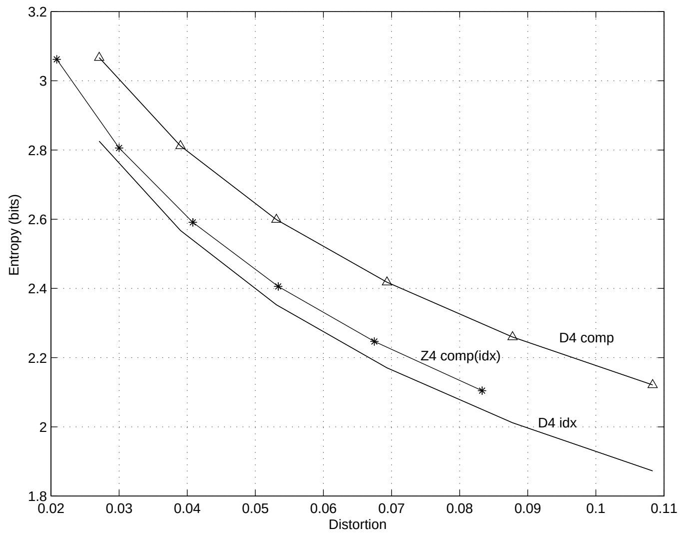


Figure 2: Comparison of rate-distortion curves for $Z_4$ and $D_4$ lattice when the lattice vector components are entropy encoded ("comp") and when the lattice vector indexes are entropy encoded ("idx").

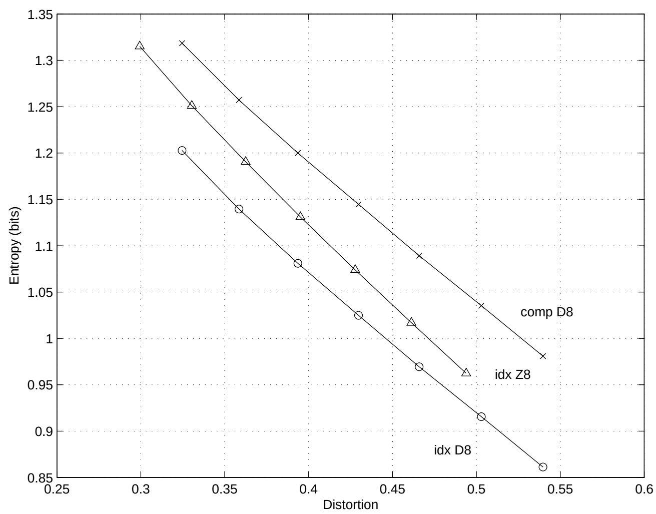


Figure 3: Comparison of rate-distortion curves for $Z_8$ and $D_8$ lattice when the lattice vector components are entropy encoded ("comp") and when the lattice vector indexes are entropy encoded ("idx").

classes [7], [13], [12]. However, the large number of leader classes for some particular truncation shapes, make their use less practical.

### 3.1 Product code lattice codevector indexing

In [11] the use of a product code type index for pyramidal truncation, in which at least the sign bits were separated has been proposed and shown to have good error resilience performance.

Using a similar approach, the idea of a product code has been extended to spherical lattice truncations [14] and to rectangular lattice truncations [10].

We propose in the present paper the use of the product code indexing from [10] for the entropy coding of the lattice codevectors. The rectangular truncation uses the maximum absolute norm of a vector $\mathbf{y} = (y_1, y_2, ..., y_n) \in \mathbf{R}^n$ defined as

$$N(\mathbf{y}) = \max_{i=1:n}(|y_i|).$$

The idea of the product code is to extract different informational entities from the vector to be indexed and concatenate their respective codes. The information contained in the vector from a rectangular $Z_n$ lattice truncation is represented by the following entities [10]:

- The number of the significant (non zero) components (A);
- The number of maximum valued components (in absolute value) (B);
- The position of the maximum valued components (C);
- The values of the significant non-maximum components (D);
- The position of the significant non maximum values (E);
- The signs of the significant components (F).

The borders between the bits corresponding to different entities that form the index are not strict, except for the bits corresponding to the signs. The strict border of the sign bits is due to the fact that they are situated at an extreme of the index and the cardinality of the set describing all the sign combinations is a power of two. The indexing corresponds to the bits ordering | A / B / C / D / E | F | . The delimiter "|" represents a strict border.

### 3.2 Entropy encoding

The different informational entities extracted from the vector, can be also interpreted as means of classifying the vectors into different sets. The existence of several entities implies the division of all the vectors into sets, sub-sets and so forth. If the index corresponding to all or part of the set(sub-set) types are entropy encoded, an entropy code can be obtained for the initial lattice vector.

For instance, given the 4 dimensional vector (2 -3 0 -1), having maximum norm equal to 3, it has three significant components (A), one maximum valued component (B), index 1 for the position of the maximum valued component (C) and index 1 for the position of the non maximum valued components (E) (see [10]). There is at least one significant value and four at the most, therefore there are four possible symbols for the number of significant components, which can be entropy encoded. Furthermore, the number of maximum valued components can be entropy encoded, as well as the position indexes of the maximum valued components and so on. There is a practical limit to the number of entities that can be entropy encoded, which is activated when the number of symbols for the considered entity becomes prohibitively large.

#### *3.2.1 Bit rate calculation*

Consider the $n$-dimensional vectors from the $Z_n$ rectangular truncation of norm $K$. Any vector from this set can be represented on $N_0$ bits, where

$$N_0 = \lceil \log_2((2K+1)^n) \rceil. \quad (2)$$

If the entity corresponding to the number of significant values is entropy encoded on $n_1$ bits, the current vector from the set of vectors can be represented on $N_1$ bits instead of $N_0$, where

$$\begin{aligned} N_1 \quad = \quad & n_1 + \left\lceil \log_2 \left( 2^S \left( \binom{n}{1} \binom{n-1}{S-1} (K-1)^{S-1} + \right. \right. \right. \\ & \left. \left. \left. \binom{n}{2} \binom{n-2}{S-2} (K-1)^{S-2} + \ldots + \binom{n}{S} \right) \right) \right\rceil, \quad (3) \end{aligned}$$

$S$ is the number of significant components.

If the number of significant components is entropy encoded on $n_1$ bits, the number of maximum valued components is encoded on $n_2$ and the index of positions for the maximum valued components is encoded on $n_3$ bits then the current vector from the set of vectors can be represented on $N_3$ bits, where

$$\begin{aligned} N_3 \quad = \quad & n_1 + n_2 + n_3 + \\ & \left\lceil \log_2 \left( 2^S \binom{n-M}{S-M} (K-1)^{S-M} \right) \right\rceil \quad (4) \end{aligned}$$

where $M$ is the number of maximum valued components whose position is already coded on $n_3$ bits.

## 4. LATTICE QUANTIZATION FOR AUDIO CODING

We exemplify the potential of the proposed method within an audio encoding algorithm. For the sake of completeness, we present briefly the overall audio encoding framework that uses rectangular lattice truncations for quantization. For a detailed description see [3]. The overall performance of the audio coding method is similar to the MPEG4-AAC for higher bitrates (128kbits/s down to 64kbits/s) and better than MPEG4-AAC for lower bitrates.

The global encoding framework is similar to the one used in the AAC. Within the bit pool mechanism, at each frame a given number of bits is available for the quantization of the modified discrete cosine transform (MDCT) coefficients grouped in several scale factor bands, according to the perceptual model. Roughly, only half of the coefficients are actually quantized, the coefficients corresponding to the higher frequencies being set to zero. The number of spectral coefficients, the number of scale factor bands and their lengths depend upon the sampling frequency of the input audio signal.

The normalized MDCT coefficients from each scale factor band $i$, are multiplied with $b^{-s_i}$ and the result is further encoded. The encoding consists of companding the scaled coefficients and quantizing using a rectangular truncation of the lattice $Z_n$. The companding function is trained off-line.

| Name | Description |
|---|---|
| es01 | Vocal (S. Vega) |
| es02 | German male speech |
| es03 | English female speech |
| sc01 | Trumpet solo and orch. |
| sc02 | Classical orch. music |
| sc03 | Contemp. pop music |
| si01 | Harpsichord |
| si02 | Castanets |
| si03 | Pitch pipe |
| sm01 | Bagpipes |
| sm02 | Glockenspiel |
| sm03 | Plucked strings |

Table 1: Test samples.

| File | BS32[bits] | BS48[bits] |
|---|---|---|
| es01 | 33 | 38 |
| es02 | 40 | 45 |
| es03 | 44 | 48 |
| sc01 | 56 | 56 |
| sc02 | 36 | 42 |
| sc03 | 24 | 29 |
| si01 | 27 | 22 |
| si02 | 45 | 56 |
| si03 | 53 | 54 |
| sm01 | 37 | 32 |
| sm02 | 42 | 42 |
| sm03 | 24 | 26 |

Table 2: Bitrate savings, in bits per frame, when the number of significant values is entropy encoded.

| File | BS32[bits] | BS48[bits] |
|---|---|---|
| es01 | 351 | 425 |
| es02 | 189 | 241 |
| es03 | 246 | 293 |
| sc01 | 174 | 174 |
| sc02 | 301 | 390 |
| sc03 | 383 | 511 |
| si01 | -109 | -50 |
| si02 | -99 | -6 |
| si03 | 70 | 77 |
| sm01 | 177 | 225 |
| sm02 | -4 | -4 |
| sm03 | 231 | 323 |

Table 3: Bit savings, in bits per frame, when the number of significant values, the number of maximum valued components, and their position index are entropy encoded.

The information to be encoded consists of the scale factor exponents $\{s_i\}$, the lattice codevector indexes, and side information providing the number of bits on which each index is represented. The maximum absolute value, i.e. the maximum norm of the scale factor band codevector, is used to calculate the number of bits on which the index of the scale factor band codevector is represented. We denote in the following $\{s_i\}$ by scales.

The scales are integers from a finite domain and they are entropy coded, same as the maximum norms of the lattice codevectors. The scale values are optimized such that the total number of bits to encode a frame is within the available number of bits given by the bit pool mechanism. Since the maximum absolute norm of the lattice codevectors is encoded separately, the indexing of the lattice codevectors is done within the corresponding rectangular shell.

## 5. RESULTS

We consider as test samples the 44.1kHz, mono samples presented in Table 1.

We have considered two encoding bit rates 32kbits/s and 48kbits/s for the audio codec from [3]. The number of bits for the quantized spectral coefficients is calculated according to the formulas from Equations 2 and 3. The difference between the average per frame number of bits $N_1$ and $N_0$ for all the spectral scale-factor bands is given numerically in Table 2. It corresponds to the case when the number of significant values is entropy encoded for all the scale-factor bands. The average codelength for $n_1$ is estimated based on the entropy.

Given that, for the considered bitrates the average number of bits per frame for the spectral quantization only is approximately 500 and 800 for 32kbits/s and 48kbits/s respectively, the absolute bit savings are not very significant yet.

However, when the first three entities (number of significant values, number of maximum valued components, and their position index) are entropy encoded, the bit savings become significant.

The difference between the average per frame number of bits $N_3$ and $N_0$ for all the spectral scale-factor bands is given numerically in Table 3. The number of bits for the quantized spectral coefficients are calculated according to the formulas from Equations 2 and 4.

Compared with number of bits per frame available for spectral quantization only in the fixed rate case, for the considered bitrates, the values in Table 3 give an average of 30% bitrate reduction without any loss of quality.

The method(labeled as 'LatVQ') without entropy coding was compared in [3], against the quantization procedure from the MPEG4-AAC codec, in a MUlti Stimulus test with Hidden Reference and Anchor (MUSHRA). A particularity of the AAC codec framework was the 11kHz bandwidth considered for quantization for all the bitrates. The files used in the tests are listed in Table 1. The files es01 and sm01 were used only in the training experiment and the remaining files were used in each of the three testing experiments. The mono files are sampled at 44.1kHz. There were 11 expert listeners.

Since the addition of the proposed entropy coding does not change the quality of the LatVQ method, it means that the conditions LatVQ_48 and LatVQ_32 (Figure 4) should actually correspond to bitrates of approximately 30 % less than 48 kbits/s and 32 kbits/s respectively.

The proposed entropy encoded method was used in this case only for the scale-factor bands with dimensions up to 24, the higher dimensional ones generating too many symbols, at least for the position index of the maximum valued components. However, previous entropy coding methods of lattice vector indexes were generally on dimension 10 or lower [5].

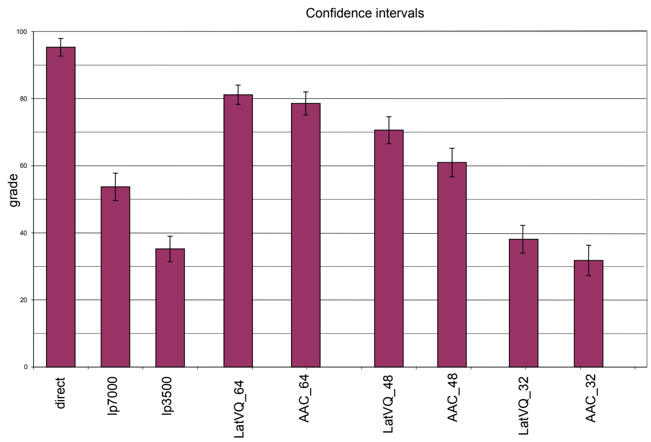


Figure 4: Listening test results [3]

## 6. CONCLUSION

We have presented a new method for entropy encoding of lattice codevectors. It is based on the lattice vector set partitioning generated by the product code indexes of such vectors. It can provide bitrate savings up to 30% within an audio coding scenario with respect to the fixed rate lattice quantization. In addition to the improved compression efficiency, the proposed method enables the use of lattice entropy encoding in higher dimensions.

## REFERENCES

[1] 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*, vol. 1, Montreal, Canada, May 17-21, 2004.

[2] A. Rämö, H. Toukomaa, S. C. Greer, L. Laaksonen, J. Stachurski, A. E. Ertan, J. Svedberg, J. Gibbs and T. Vaillancourt "Quality evaluation of the G.EV-VBR speech codec, " in *Proceedings of ICASSP 2008*, Las Vegas, USA, March 2008.

[3] A. Vasilache and H. Toukomaa "Vectorial spectral quantization for audio coding," in *Proceedings of ICASSP 2006*, Toulouse, France, May 14-17, 2006.

[4] T.R. Fischer, "Entropy-constrained geometric vector quantization for transform image coding" in *Proceedings of ICASSP 1991*, pp. 2269 - 2272, vol.4

[5] A. Vasilache and I. Tăbuş, "Indexing and entropy coding of lattice codevectors," in *Proceedings of ICASSP 2001*, vol. 4, pp. 2605–2608, 2001.

[6] D. Y. Zhao, J. Samuelsson and M. Nilsson, "GMM-based Entropy-Constrained Vector Quantization " in *Proceedings of ICASSP 2007*, Hawaii, USA, April, 2007.

[7] P. Rault and C. Guillemot, "Indexing algorithms for $Z_n$, $A_n$, $D_n$, and $D_n^{++}$ lattice vector quantizers," *IEEE Trans. on Multimedia*, vol. 3, no. 4, pp. 395–404, Dec. 2001.

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

[9] J.-P. Adoul and M. Barth, *"Nearest Neighbor Algorithm for Spherical Codes from the Leech Lattice"*, IEEE Transactions on Information Theory, vol. 38, No. 5II, September 1988, pp. 1188-1202.

[10] A. Vasilache "Indexing of lattice codevectors applied to error resilient audio coding, " in *Proc. of the AES* $30^{th}$ *International Conference*, Saariselkä, Finland, March, 15-17, 2007.

[11] A. C. Hung, E. K. Tsern and T. H. Meng, *"Error-resilient pyramid vector quantization for image compression,"* IEEE Trans. on Image Processing, vol. 7, no. 10, October, 1998.

[12] P. Loyer, J.-M. Moureaux and M. Antonini, *"Lattice codebook enumeration for generalized Gaussian source,"* IEEE Trans. on Information Theory, vol. 49, no. 2, February, 2003.

[13] J.-M. Moureaux, P. Loyer, and M. Antonini, *"Low-complexity indexing method for* $Z^n$ *and* $D_n$ *lattice quantizers,"* IEEE Trans. on Communications, vol. 46, no. 12, December, 1998.

[14] M. E. Constantinescu, *"Lattice VQ for image compression"*, Internal report, Tampere University of Technology, Tampere, Finland, 2001.