

New directions in lattice based lossy compression

Adriana Vasilache
Nokia Research Center
Finland

1. Introduction

One of the first articles that have addressed lattice quantizers in practical applications is the work of Sayood et al. (Sayood et al., 1984). The lattice quantization has been strongly stimulated by the articles of Conway and Sloane (Conway & Sloane, 1982), (Conway & Sloane, 1983) presenting fast algorithms for nearest neighbor search algorithms. The principal factors that have brought the lattices in the attention of the quantization community are:

- The lattices are uniformly distributed structures in the n -dimensional space, therefore they are immediately applicable as quantizer structures for uniform sources. This affirmation is based on the, now widely accepted, conjecture of Gersho (Gersho, 1979), stating that, when the rate is high, the optimal quantizer of a uniform source will have the partition cells all congruent to some polytope. This is equivalent to saying that the optimal quantizer of a uniform source is a tessellating¹ quantizer, i.e. it completely fills the space with copies of a same polytope. Gersho (Gersho, 1979) has shown that this polytope must have the lowest normalized second order moment for the considered space dimension.
- The asymptotic equipartition property, used in the context of source coding, suggests that in a high-dimensional space almost all points will lie in a region of high probability specified by the entropy of the source (Cover & Thomas, 1991). The region of high probability will have a shape dependent on the source (Fischer, 1989) (e.g. the hypersphere for the memoryless Gaussian source (Sakrison, 1968), the hyper-pyramid (hyper-octahedron) for the memoryless Laplacian source (Fischer, 1986)). The pdf of the points, f , being almost constant on that region of high probability, the formula under the high-rate assumption, for the point density of the optimal quantizer (Gersho, 1979) indicates, that the codebook should also be uniform in that region.
- The regular structure of a lattice VQ reduces considerably the memory requirements for the storage of the codebook.
- Fast nearest neighbor search algorithms exist for the lattices which are generally used as quantizers (Conway & Sloane, 1992).

State of the art speech codecs as AMR-WB+ (Ragot et al., 2004) and G.718 (Rämö et al., 2008) codec make use of its advantages. Audio coding methods based on lattice quantization have been as well proposed (Vasilache & Toukoma, 2006).

¹ All the lattices form tessellations, but not all tessellations are obtained from lattices.

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 (Ragot et al., 2004). There exist also variable rate encoding techniques of lattice codevectors. Most of these methods rely on the grouping of codevectors on classes such as leader classes or shells (Fischer, 1991), (Vasilache & Tabus, 2001) or apply directly entropy coding methods to the lattice codevector components (Zhao et al., 2007). 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 discuss in the present work a new approach for entropy encoding of lattice codevectors that can be applied for higher dimensional lattices without additional storage requirements and that allows parameterization of the lattice truncation size. The proposed method is based on the indexing method for lattice vectors that makes use of the product code indexing method. The presented approach is exemplified on rectangular truncation of lattices, where the number of leader classes is relatively high, but the shape of the truncation is accounted for through companding.

This work 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 results. Future research directions will be discussed and conclusions of the work will make the object of the last section.

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 Λ is a set of real vectors whose coordinates are integers in a given basis $\{b_i \in \mathbf{R}^n\}_{i=1,n}$

$$\Lambda = \left\{ \mathbf{v} \in \mathbf{R}^n \mid \mathbf{v} = \sum_{i=1}^n \alpha_i b_i, \alpha_i \in \mathbf{Z} \right\}. \quad (1)$$

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 bit 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 :

$$\Lambda_K = \left\{ (\mathbf{x}) = (x_1, x_2, \dots, x_n) \in \Lambda \mid N(\mathbf{x}) \leq K \right\}. \quad (2)$$

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. Also other, more general norms, can be considered.

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

$$\Lambda_{K_i} = \left\{ (\mathbf{x}) = (x_1, x_2, \dots, x_n) \in \Lambda \mid |x_i| \leq K_i \right\}. \tag{3}$$

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.

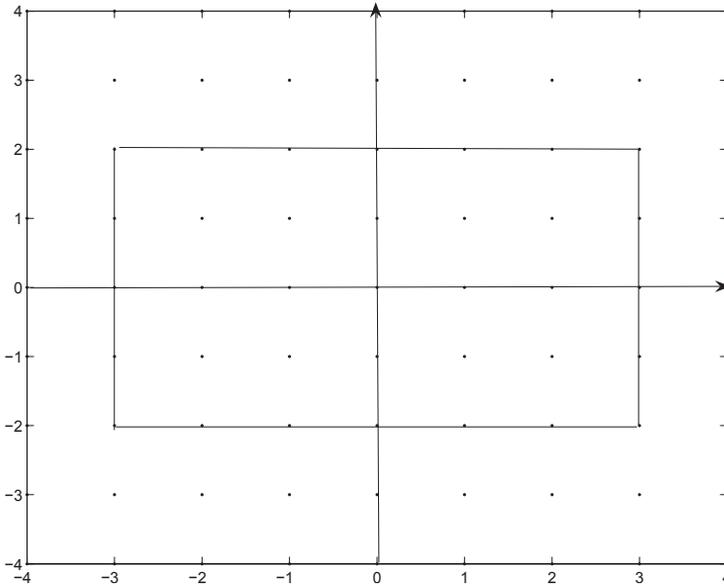


Fig. 1. Illustration of the generalized rectangular truncation of Z_2 .

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 :

$$\Lambda_K = \left\{ (\mathbf{x}) = (x_1, x_2, \dots, x_n) \in \Lambda \mid N(\mathbf{x}) = K \right\}. \tag{4}$$

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 $\mathbf{v} = (v_m, \dots, v_m, \dots, v_i, \dots, v_i, \dots, v_1, \dots, v_1)$ where $0 \leq v_1 < \dots < v_i < \dots < v_m$. The leader class of the leader vector \mathbf{v} is the set of 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 (Adoul, 1986), (Adoul & Barth, 1988).

Most of the lattices used for quantization can be defined as union of leader classes (Moureaux et al., 1998).

2.3 Counting lattice points

The use of lattice truncations as quantizers implies knowing the number of lattice codevectors inside the considered truncation. Following the definition of a lattice truncation as a union of shells, counting the lattice points reduces to finding expressions for the cardinality of a shell, i.e. the number of lattice points at a given distance from the origin, under the specified norm. The solution of this problem is given by the *theta* functions for ℓ_2 norm for many standard lattices in (Conway & Sloane, 1992). In (Solé, 1993) the *theta* functions have been generalized for the norm ℓ_p and in (Moureaux et al., 1995), (Barlaud et al., 1993) for weighted ℓ_2 norms. In (Vasilache et al., 1999) the *theta* series approach is used to count the lattice points on spherical (ℓ_2 norm) shells and generalized to pyramidal (ℓ_1 norm) shells.

A second method of counting the points from a truncated lattice is based on the notion of leader classes: from the definition of the leader class, the number of vectors belonging to that class can be easily deduced using polynomial coefficients (Moureaux et al., 1998), (Rault & Guillemot, 2001). This approach is also more appropriate for applications where the indexing of lattice points is also required. There exist other methods for counting the lattice points, but they will be treated in the section dedicated to the indexing of lattice points.

2.4 Indexing the vectors in truncated lattices

Several lattice enumeration techniques have been proposed over the years for different truncations and lattice types. One of the first papers to present an indexing algorithm for lattices was (Conway & Sloane, 1983), but it was restricted to Voronoi truncated lattices. Few years later, Fischer introduced the first enumeration technique on pyramidal truncated lattice in (Fischer, 1986) which he subsequently generalized for weighted pyramids in Z_n (Fischer, 1989), (Fischer & Pan, 1995). This method, which we dub Fischer enumeration, is based on the iterative counting

$$N(l, k) = \sum_{i=-k}^k N(l-1, k-|i|) \quad (5)$$

where $N(l, k)$ is the number of vectors in the pyramidal shell of norm k of the lattice Z_l . $N(l, k)$ can be viewed as the number of ways l integer values can sum up in absolute value to k . For maximum efficiency, the numbers $N(k, l)$ must be stored, resulting in a table of size logarithmic in the codebook size. Alternatively, methods of deriving the values of $N(l, k)$ are presented in (Hung et al., 1998). A second type of indexing method, also based on an iterative counting of the points having a certain property has been presented in (Hung et al., 1998). There are four significant quantities of a codevector, which can be iteratively numbered, finally their juxtaposition forming a product code. These quantities are:

1. $D(s, l)$: the number of possible distinct distributions of s elements in l locations,
2. $S(s, k)$: the number of possible combinations of s non-zero elements that sum up to k (distinct additive partitions),
3. $B(s)$: the number of sign combinations for s non-zero elements.

In terms of a lattice codevector, s is the number of non-zero components of the l dimensional vector. The number of points for a given ℓ_1 norm k is thus given by (Hung et al., 1998)

$$N(l, k) = \sum_{s=1}^m B(s)D(s, l)S(s, k) = \sum_{s=1}^m 2^s \binom{l}{s} \binom{k-1}{s-1} \quad (6)$$

where m is the maximum number of non-zero elements in the lattice vectors included in truncation. The use of a product code enhances the error resilience over noisy channels, when compared to the original enumeration proposed by Fischer (Hung et al., 1998). These algorithms, as described in (Hung et al., 1998) apply mainly to Z_n lattices or D_n with pyramidal truncations. The product code of (Hung et al., 1998) can be generalized to spherical truncations, but with some additional storage requirements for the term $S(s, k)$ (Constantinescu, 2001).

In (Serra-Sagrista, 2000) combinatorial formulas like in (6) have been proposed for the A_n , D_n^* and D_n^+ lattices with pyramidal truncation. A generalization of Fischer's method to lattices derived from binary linear block codes through *Construction A* and *B* (Conway & Sloane, 1992) has been presented for pyramidal truncations in (Wang et al., 1998). This method has $O(nK)$ computational complexity, where n is the lattice dimension and K the truncation maximum ℓ_1 norm, and it is based on a Fischer type enumeration of pyramidal truncations of Z_n and of translated $2Z_n$.

An indexing technique based on the notion of leader vector of a lattice was developed in (Moureaux et al., 1998) for Z_n and D_n lattices. In (Vasilache & Tabus, 2002) a method based on leader vectors for lattices that can be defined as unions of leader classes (including Z_n , D_n , D_n^* and D_n^+ lattices) has been proposed. Rault and Guillemot (Rault & Guillemot, 2001) have presented an enumeration based on signed leaders or generated signed leaders valid for a large class of lattices (Z_n , A_n , D_n and D_n^{++}). The principle of the methods based on leaders, is to count the signed permutations generating the vectors in a leader class. The methods described in (Moureaux et al., 1998) and (Rault & Guillemot, 2001) are based on the lexicographical or inverse lexicographical order of vectors. The methods proposed in (Vasilache & Tabus, 2002) utilize also a second possible order of the vectors within a leader class, based on binomial coefficients.

3. Lattice entropy coding

Allowing variable bit rate encoding through entropy encoding brings substantial compression efficiency increase. Moreover, in the case of vector quantization, lattice vector quantization in particular, it is more efficient to entropy encode the codevector indexes than the codevector components.

This fact is illustrated in figures 2 and 3 where experimental compression performance in the rate-distortion plane is drawn for the lattices D_4 and D_8 respectively. The curve marked as "comp" corresponds to the case when the lattice codevector 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. Zero mean 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.

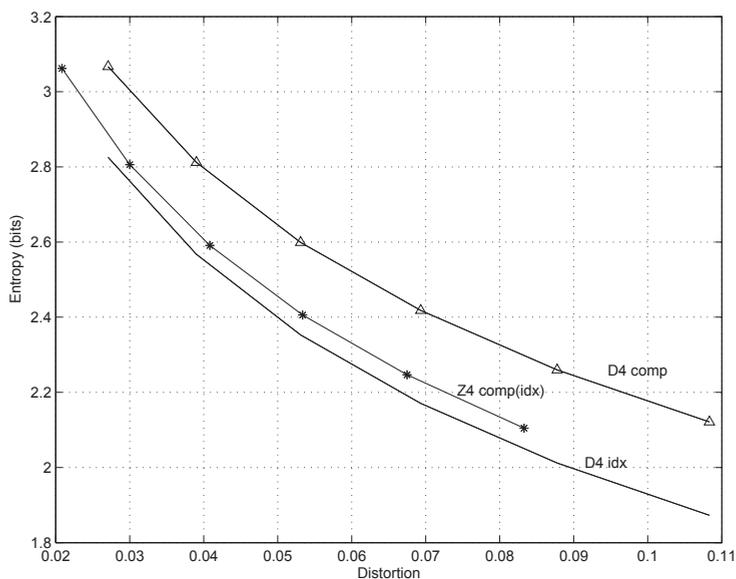


Fig. 2. Comparison of rate-distortion curves for Z_4 and D_4 lattice when the lattice vector components are entropy encoded (“comp”) and when the lattice vector indexes are entropy encoded (“idx”).

Ideally, the entropy coding of lattice codevectors should consider each codevector individually. However, the use of lattice codebooks is most useful 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 encoding of the index of the set while the vector index within the set is encoded using enumerative coding (Vasilache & Tabus, 2001), (Rault & Guillemot, 2001), (Moureaux et al., 1998), (Loyer et al., 2003). However, the large number of leader classes for some particular truncation shapes, makes their use less practical.

Another approach has been to entropy encode the lattice vector components (Zhao et al., 2007), 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.

3.1 Product code lattice codevector indexing

In (Hung et al., 1998) 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 (Constantinescu, 2001) and to rectangular lattice truncations (Vasilache, 2007).

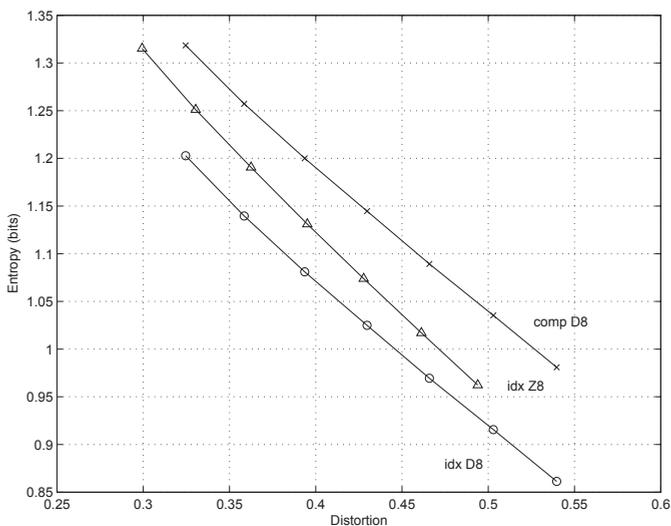


Fig. 3. Comparison of rate-distortion curves for Z_8 and D_8 lattice when the lattice vector components are entropy encoded (“comp”) and when the lattice vector indexes are entropy encoded (“idx”).

We propose in the present study the use of the product code indexing from (Vasilache, 2007) for the entropy coding of the lattice codevectors. The rectangular truncation uses the maximum absolute norm of a vector $\mathbf{y} = (y_1, y_2, \dots, y_n) \in \mathbf{R}^n$ defined as

$$N(\mathbf{y}) = \max_{i=1:n} (|y_i|). \tag{7}$$

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:

- 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 within the lattice codevector (C);
- The values of the significant non-maximum components (D);
- The position of the significant non maximum values within the lattice codevector without the maximum valued components (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 $\boxed{A / B / C / D / E \mid F}$. The delimiter “|” represents a strict border.

3.1.1 Alternate approaches

There are more possible representations into information units. For instance, an equivalent representation can be (representation II):

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

or as representation III:

- 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 within the lattice codevector (C'');
- The positions of the significant non-maximum values within the lattice codevector without the maximum components (D'');
- The values of the significant non-maximum components (E'');
- The signs of the significant components (F'').

3.2 Entropy coding based on product code lattice codevector indexing

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) (Vasilache, 2007). There are at least one and at most four significant values, 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. For instance, for the encoding of the index of non-maximum significative values there are $(K - 1)^{S-M}$ possible symbols, where S is the number of significative components and M the number of maximum components. For high truncation size (K) and/or high number of non-maximum significative values ($(S - M)$), this number becomes large and the probability of the index to be encoded very hard to model.

For small lattice codevector dimension the proposed lattice codevector entropy encoding method might become less efficient than the fixed rate encoding because there are more sources of inexact modelling.

3.2.1 Lattice truncation size parametrization

In the previously presented representations of the index, the lattice truncation size, given by its norm, is considered to be fixed. A more flexible approach for data with wide range of variation is obtained if the value of the current maximum is considered as side information that is entropy encoded.

3.2.2 Context entropy encoding of the index information units

The encoding of the information units should be done context based, because there is a strong correlation between the different units involved.

For instance, let's consider representation II, when the value of the maximum for each lattice codevector is transmitted as side information and the variables to be encoded (maximum value, number of significant components, number of maxima, position of significant components, position of maxima, index of non-maximum significant values, signs of significant components) are denoted respectively by

$$K, S, M, pS, pM, nM, sg.$$

Then the probability models for each variable are:

$$p(K), p(S|K), p(M|S, K), p(pS|S), p(pM|S, M), p(nM|S, M), p(sg).$$

For the first variables (K, S, M) their actual values are encoded. For pS and pM a position index specifying the location of l components out of n possible locations is created. A position vector $\mathbf{r} = (r_0, \dots, r_{l-1}) \in 0, \dots, n-1^l, r_0 < \dots < r_{l-1}$ is created, which specifies the exact location of each of the l components. Since there are $\binom{n}{l}$ such vectors, they can be enumerated like binomial coefficients following the algorithm given by the next equations:

$$I_{pos}(l, n, \mathbf{r}) = \sum_{i=1}^{r_0} \binom{n-i}{l-1} + I_{pos}(n-r_0-1, l-1, (r_1, \dots, r_{l-1}) - r_0 - 1) \tag{8}$$

$$I_{pos}(l', 1, [i]) = i, \quad 0 \leq i < l' \leq l. \tag{9}$$

The resulting index I_{pos} is the number to be encoded for pS and pM .

The index to be encoded for the values of non-maximum significant components is calculated as

$$I_{nM} = \sum_{i=1}^{S-M} (K-1)^{i-1} (y_i - 1) \tag{10}$$

where $y_i, i = 0, S - M - 1$ are the non-maximum significant values.

3.2.3 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

$$N_0 = \lceil \log_2((2K + 1)^n) \rceil. \tag{11}$$

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

$$N_1 = n_1 + \left[\log_2 \left(2^S \left(\binom{n}{1} \binom{n-1}{S-1} (K-1)^{S-1} + \binom{n}{2} \binom{n-2}{S-2} (K-1)^{S-2} + \dots + \binom{n}{S} \right) \right) \right], \quad (12)$$

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 bits 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

$$N_3 = n_1 + n_2 + n_3 + \left[\log_2 \left(2^S \binom{n-M}{S-M} (K-1)^{S-M} \right) \right] \quad (13)$$

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

If, in addition, the positions of the non-maximum significant values are entropy encoded on n_4 bits, then the current vector from the set of vectors can be represented on N_4 bits, where

$$N_4 = n_1 + n_2 + n_3 + n_4 + \left[\log_2 \left((K-1)^{S-M} \right) \right] + S. \quad (14)$$

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 (Vasilache & Toukoma, 2006). The overall performance of the audio coding method is similar to the MPEG4-AAC for higher bitrates (128kbts/s down to 64kbts/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.

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

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.

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 32kb/s and 48kb/s for the audio codec from (Vasilache & Toukoma, 2006). The number of bits for the quantized spectral coefficients is calculated according to the formulas from Equations 11 and 12. 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 percentages 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. 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 11 and 13.

File	BS32[%]	BS48[%]
es01	6.60	4.75
es02	8.00	5.62
es03	8.80	6.00
sc01	11.20	7.00
sc02	7.20	5.25
sc03	4.80	3.62
si01	5.40	2.75
si02	9.00	7.00
si03	10.60	6.75
sm01	7.40	4.00
sm02	8.40	5.25
sm03	4.80	3.25

Table 2. Bitrate savings, in percentage, when the number of significant values is entropy encoded.

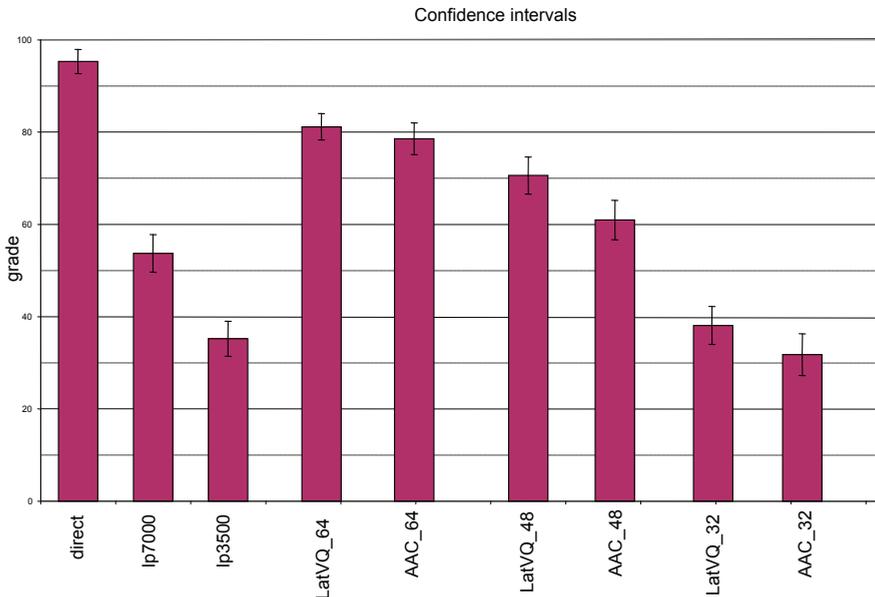


Fig. 4. Listening test results (Vasilache & Toukoma, 2006)

Compared with the 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 (Vasilache & Toukoma, 2006), against the quantization procedure from the MPEG4-AAC codec, in a Multi Stimulus test with Hidden Reference and Anchor (MUSHRA) (BS.1534-1, 2003). A particularity of the AAC codec framework was the 11kHz bandwidth considered for quantization for

File	BS32[%]	BS48[%]
es01	70.20	53.12
es02	37.80	30.12
es03	49.20	36.62
sc01	34.80	21.75
sc02	60.20	48.75
sc03	76.60	63.87
si01	-21.80	-6.25
si02	-19.80	-0.75
si03	14.00	9.62
sm01	35.40	28.12
sm02	-0.80	-0.50
sm03	46.20	40.37

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

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. 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 (Vasilache & Tabus, 2001).

5.1 Further discussions

A very delicate matter related to the enumeration of lattice points is the error resilience over a noisy channel. Few papers (Hung et al., 1998), (Vasilache & Tabus, 2002), (Vasilache & Tabus, 2003) have dealt with the error resilience over the channel for lattice codebooks.

In (Hung et al., 1998) the channel error resilience is obtained through the use of product code based indexing while in (Vasilache & Tabus, 2003) lexicographical and binomial families of indexing methods are proposed, allowing the optimization of the indexing with respect to the channel distortion (or some other criterion) within a given family.

The proposed lattice coding, being an entropy encoding method, is on one side sensitive to channel errors but is has built-in error concealment mechanisms due to the dependencies existing between the information units that are encoded. In addition, extending the observations from (Vasilache, 2007), the proposed method can made scalable in bitrate through the control of the variables that are encoded, allowing thus an approximate representation of the original lattice codevector and the use of the corresponding bits for channel protection, for instance. The potential of this approach needs to be investigated through future studies.

Another potential direction of investigation is to study the time correlation of each information unit to be encoded, that should be easier to exploit than the time correlation of the lattice codevector indexes if it exists.

The method presented here can be applied wherever product code indexing is applicable (pyramidal or spherical truncations as well), but it is limited so far to Z_n, D_n, D_n^* and D_n^+ lattices.

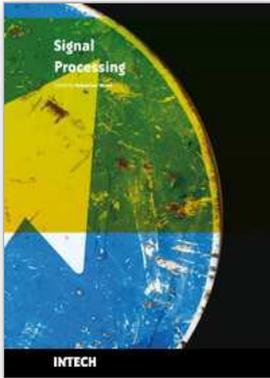
6. Conclusion

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.

7. References

- Adoul, J.-P. (1986). La quantification vectorielle des signaux: approche algébrique, *Annales des Télécommunications* **41**(3-4): 158–177.
- Adoul, J.-P. & Barth, M. (1988). Nearest neighbor algorithm for spherical codes from the leech lattice, *IEEE Transactions on Information Theory* **38**(5II): 1188–1202.
- Barlaud, M., Solé, P., Moureaux, J. M., Antonini, M. & Gauthier, P. (1993). Elliptical codebook for lattice vector quantization, *Proceedings of the ICASSP 1993, Minneapolis, USA*, pp. 590–593.
- BS.1534-1, I.-R. (2003). Method for the subjective assessment of intermediate quality level of coding systems.
- Constantinescu, M. E. (2001). Lattice VQ for image compression, Internal report, Tampere University of Technology, Tampere, Finland.
- Conway, J. H. & Sloane, N. J. A. (1982). Fast quantizing and decoding algorithms for lattice quantizers and codes, *IEEE Transactions on Information Theory* **IT-28**(2): 227–232.
- Conway, J. H. & Sloane, N. J. A. (1983). A fast encoding method for lattice codes and quantizers, *IEEE Transactions on Information Theory* **37**(5): 820–824.
- Conway, J. H. & Sloane, N. J. A. (1992). *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York.
- Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*, John Wiley & Sons, Inc.
- Fischer, T. R. (1986). A pyramid vector quantizer, *IEEE Transactions on Information Theory* **IT-32**(4): 568–583.
- Fischer, T. R. (1989). Geometric source coding and vector quantization, *IEEE Transactions on Information Theory* **35**(1): 137–144.
- Fischer, T. R. (1991). Entropy-constrained geometric vector quantization for transform image coding, *Proceedings of ICASSP 1991, Vol. 4*, pp. 2269–2272.
- Fischer, T. R. & Pan, J. (1995). Enumeration encoding and decoding algorithms for pyramid cubic lattice and trellis codes, *IEEE Transactions on Information Theory* **41**(6): 2056–2061.
- Gersho, A. (1979). Asymptotically optimal block quantization, *IEEE Transactions on Information Theory* **IT-25**(4): 373–380.
- Hung, A. C., Tsern, E. K. & Meng, T. H. (1998). Error-resilient pyramid vector quantization for image compression, *IEEE Transactions on Image Processing* **7**(10): 1373–1386.

- Loyer, P., Moureaux, J. M. & Antonini, M. (2003). Lattice codebook enumeration for generalized Gaussian source, *IEEE Transactions on Information Theory* **49**(2).
- Moureaux, J., Antonini, M. & Barlaud, M. (1995). Counting lattice points on ellipsoids: application to image coding, *Electronics letters* **31**(8): 1224–1225.
- Moureaux, J. M., Loyer, P. & Antonini, M. (1998). Low complexity indexing method for Z_n and D_n lattice quantizers, *IEEE Transactions on Communications* **46**(12): 1602–1609.
- Ragot, S., Bessette, B. & Lefebvre, R. (2004). Low-complexity multi-rate lattice vector quantization with application to wideband txc speech coding at 32 kbit/s, *Proceedings of ICASSP 2004*, Vol. 1, Montreal, Canada.
- Rämö, A., Toukoma, H., Greer, S. C., Laaksonen, L., Stachurski, J., Ertan, A. E., Svedberg, J., Gibbs, J. & Vaillancourt, T. (2008). Quality evaluation of the g.ev-vbr speech codec, *Proceedings of ICASSP 2008*, Las Vegas, USA.
- Rault, P. & Guillemot, C. (2001). Indexing algorithms for Z_n , A_n , D_n and D_n^{++} lattice vector quantizers, *IEEE Transactions on Multimedia* **3**(4): 395–404.
- Sakrison, D. J. (1968). A geometric treatment of the source encoding of a Gaussian random variable, *IEEE Transactions on Information Theory* **IT-14**(3): 481–486.
- Sayood, K., Gibson, J. D. & Rost, M. C. (1984). An algorithm for uniform vector design, *IEEE Transactions on Information Theory* **IT-30**(6): 805–814.
- Serra-Sagrasta, J. (2000). Enumeration of lattice points in ℓ_1 norm, *Information processing letters* **76**: 39–44.
- Solé, P. (1993). Generalized theta functions for lattice vector quantization, *Proceedings of IEEE International Symposium on Information Theory*, pp. 174–174.
- Vasilache, A. (2007). Indexing of lattice codevectors applied to error resilient audio coding, *Proceedings of the AES 30th International Conference*, Saariselkä, Finland.
- Vasilache, A. & Tabus, I. (2001). Indexing and entropy coding of lattice codevectors, *Proceedings of ICASSP 2001*, Salt Lake City, Utah, USA.
- Vasilache, A. & Tabus, I. (2002). LSF quantization with MSLLVQ for transmission over noisy channels, *Proceedings of EUSIPCO 2002*, Toulouse, France.
- Vasilache, A. & Tabus, I. (2003). Robust indexing for lattices and permutation codes over binary symmetric channels, *Signal Processing* **83**(7): 1467–1486.
- Vasilache, A. & Toukoma, H. (2006). Vectorial spectral quantization for audio coding, *Proceedings of ICASSP 2006*, Toulouse, France.
- Vasilache, A., Vasilache, M. & Tabus, I. (1999). Predictive multiple-scale lattice VQ for LSF quantization, *Proceedings of ICASSP 1999*, Phoenix, Arizona, USA, pp. 657–660.
- Wang, C., Cao, H. Q., Li, W. & Tzeng, K. K. (1998). Lattice labeling algorithms for vector quantization, *IEEE Transactions on Circuits and Systems for Video Technology* **8**(2): 206–220.
- Zhao, D. Y., Samuelsson, J. & Nilsson, M. (2007). GMM-based entropy-constrained vector quantization, *Proceedings of ICASSP 2007*, Hawaii, USA.



Signal Processing

Edited by Sebastian Miron

ISBN 978-953-7619-91-6

Hard cover, 528 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

This book intends to provide highlights of the current research in signal processing area and to offer a snapshot of the recent advances in this field. This work is mainly destined to researchers in the signal processing related areas but it is also accessible to anyone with a scientific background desiring to have an up-to-date overview of this domain. The twenty-five chapters present methodological advances and recent applications of signal processing algorithms in various domains as telecommunications, array processing, biology, cryptography, image and speech processing. The methodologies illustrated in this book, such as sparse signal recovery, are hot topics in the signal processing community at this moment. The editor would like to thank all the authors for their excellent contributions in different areas of signal processing and hopes that this book will be of valuable help to the readers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Adriana Vasilache (2010). New Directions in Lattice Based Lossy Compression, Signal Processing, Sebastian Miron (Ed.), ISBN: 978-953-7619-91-6, InTech, Available from: <http://www.intechopen.com/books/signal-processing/new-directions-in-lattice-based-lossy-compression>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.