Gosset Low Complexity Vector Quantization with Application to Audio Coding

Hauke Krüger, Bernd Geiser, Peter Vary

Institute of Communication Systems and Data Processing (ind), RWTH Aachen University, Aachen, Germany
E-Mail: {krueger,geiser,vary}@ind.rwth-aachen.de
Web: www.ind.rwth-aachen.de

Abstract

This paper introduces a novel and highly efficient realization of a spherical vector quantizer (SVQ), the "Gosset Low Complexity Vector Quantizer" (GLCVQ). The GLCVQ codebook is composed of vectors that are located on spherical shells of the eight-dimensional Gosset lattice E_8. A high encoding efficiency is achieved by representing the spherical vector codebook as aggregated *permutation codes*. Compared to previous algorithms, the computational complexity and memory consumption is further reduced by grouping the so-called *classleader vectors* into a significantly lower number of *classleader root vectors* based on separate handling of signs and magnitudes for each vector coordinate. The GLCVQ concept can be generalized to vector dimensions that are multiples of eight. In particular, GLCVQ for 16-dimensional vectors is used in Amd. 6 to ITU-T Rec. G.729.1 for super-wideband speech and audio coding.

1 Introduction

In gain-shape vector quantization [12], the input vector $\mathbf{x} \in \mathbb{R}^n$ is decomposed into a gain factor $g \geq 0$ and a shape vector $\mathbf{c} \in \mathbb{R}^n$ which are then quantized independently by means of a scalar and a vector quantizer, respectively. Typically, the euclidean norm is used and therefore the gain factor g and the shape vector $\mathbf{c}$ are computed as

$$g = |\mathbf{x}| = ||\mathbf{x}||_2 \qquad \text{and} \qquad \mathbf{c} = g^{-1}\mathbf{x}. \tag{1}$$

With this normalization, all vectors $\{\mathbf{c}\}$ computed from a series of input vectors $\{\mathbf{x}\}$ are located on the surface of the n-dimensional unit sphere. In order to match the properties of the normalized vectors $\{\mathbf{c}\}$, the codevectors of the vector quantizer should be located on the surface of the unit sphere as well. Such a vector quantizer is referred to as a *spherical vector quantizer* (SVQ). In [8] and [7] it has been shown that, when combined with logarithmic scalar quantization of the gain factor g, the quantization performance of the gain-shape approach is independent from the statistical characteristics of the input signals. Therefore, this concept is particularly suited for the encoding of audio signals which usually have unknown characteristics. Practical realizations of SVQ have been described in the literature, e.g., [9, 10], and several others are part of recent speech and audio coding standards, for instance ITU-T Rec. G.729.1 [6, 11], ITU-T Rec. G.718 [4, 15], ITU-T Rec. G.719 [5, 16], and 3GPP AMR-WB+ [1]. In particular, an approach for SVQ based on the well-known *Gosset lattice* E_8 has first been applied to speech coding in [2]. Here, lattice points that are located on spherical shells of constant radius form the basis of the SVQ codebooks. An efficient realization of the nearest-neighbor quantization routine can be achieved by grouping the codevectors into classes that can be interpreted as *permutation codes*. Each class is then represented by an associated *classleader vector*. Even though this approach is computationally efficient compared to a full codebook search, it is still not applicable for higher bit rates since the number of classleader vectors increases too much.

In this paper, a novel technique for nearest-neighbor quantization based on spherical codevectors that are taken from shells of the Gosset lattice is proposed. Thereby, the computational complexity and the memory consumption is significantly reduced by grouping the classleader vectors according to a significantly lower number of *classleader root vectors* based on separate handling of the signs and the magnitudes of the codevector coordinates. This quantizer, which is referred to as "Gosset Low Complexity Vector Quantizer" (GLCVQ), has been successfully applied for super-wideband speech and audio coding in the codec proposal of [3]. It is now part of Amd. 6 to ITU-T Rec. G.729.1.

This work has been conducted in cooperation with Huawei Technologies Co., Ltd., Core Network Research Dept., Beijing, P.R. of China

2 SVQ Based on the Gosset Lattice

The Gosset lattice is defined in eight dimensions, as the superposition of the checkerboard lattice D_8 and a shifted version thereof,

$$E_8 \doteq D_8 \cup (D_8 + \mathbf{v}), \mathbf{v} = \left[\tfrac{1}{2} \quad \dots \quad \tfrac{1}{2}\right]^{\mathrm{T}}. \tag{2}$$

The checkerboard lattice is defined for arbitrary dimensions n as

$$D_n \doteq \left\{ \mathbf{x} = [x_0 \ \dots \ x_{n-1}]^{\mathrm{T}} \in \mathbb{Z}^n : (\sum_{i=0}^{n-1} x_i) \bmod 2 \equiv 0 \right\}. \tag{3}$$

Lattice vectors with a constant distance to the origin define a *shell of a lattice*. The spherical vector codebook of the SVQ to be investigated in the following is composed of all N vectors which fulfill the Gosset lattice condition (2) and at the same time are located on a shell with a specific radius, normalized to have unit absolute value. Targeting a nearest-neighbor quantization with low complexity and memory, due to the invariance of (2) against permutation of the vector coordinates, the N codevectors populating the SVQ codebook can be represented by permutations codes as shown in [2].

Let the vector $\tilde{\mathbf{x}}_i \in \mathbb{R}^n$ be one out of p *classleader vectors*. Each classleader vector is composed of $L \leq n$ different real valued amplitudes μ_l distributed over the n vector coordinates in decreasing order $\mu_0 > \mu_1 > \cdots > \mu_{L-1}$, i.e.,

$$\begin{aligned} \tilde{\mathbf{x}}_i &= [\tilde{x}_{i,0} \quad \tilde{x}_{i,1} \quad \dots \tilde{x}_{i,n-1}]^{\mathrm{T}} \\ &= \left[\overset{\leftarrow w_0 \rightarrow}{\mu_0 \quad \mu_0} \quad \overset{\leftarrow w_1 \rightarrow}{\mu_1 \quad \mu_1} \dots \overset{\leftarrow w_{L-1} \rightarrow}{\mu_{L-1} \quad \mu_{L-1}}\right]^{\mathrm{T}}. \end{aligned} \tag{4}$$

Each of the real values μ_l can occur w_l times within the vector. A *permutation* of the vector $\tilde{\mathbf{x}}_i$ is defined as another vector $\tilde{\mathbf{x}}_i'$ that is composed of the same real values μ_l but in a different order. An *equivalence class* is defined as the set of codevectors which can be produced by arbitrary permutations of a single classleader vector. Finally, the SVQ codebook is defined as the aggregation of the codevectors of the equivalence classes related to all p classleader vectors, normalized to have unit absolute value.

The advantage of the permutation code representation of the codebook is that an efficient nearest-neighbor quantization routine can be employed as proposed in [14] where only the classleader vectors must be evaluated rather than all vectors in the codebook in order to find the optimal codevector.

Examples of the number of spherical codevectors and corresponding classleader vectors for codebook designs at different effective bit rates per vector coordinate are listed in Table 1.

ISBN 978-3-8007-3300-2

Table 1: Exemplary numbers of codevectors, equivalence classes and equivalence root classes for the Gosset lattice E_8.

Bit rate $\log_2(N)/n$	Codevectors N	Equiv. classes p (cf. [2])	Equiv. root classes q
0.988	240	8	2
1.762	17520	42	5
2.042	82560	103	8
2.257	272160	162	11

3 Gosset Low Complexity VQ

The computational complexity of the SVQ approach as described in the previous section is still quite high, in particular at higher bit rates. E.g, in the example shown in the last row of Table 1, 162 classleader vectors must be evaluated in the nearest neighbor quantization procedure to find the optimal codevector for a given input vector $\mathbf{x}$. To further reduce the complexity, $q < p$ *classleader root vectors* are defined in the GLCVQ approach, cf. Section 3.1 . The achievable computational complexity is significantly lower than with the conventional approach since only the q classleader root vectors must be evaluated rather than the p classleader vectors in order to find the optimal codevector. Examples for the relation of the number of classleaders p to the number of classleader root vectors q are also given in Table 1.
Usually, the Gosset lattice is defined for eight dimensions. Due to its construction rule which is based on the general D_n lattice, formally, the proposed GLCVQ concept can be generalized to arbitrary dimensions. High quantization performance, however, can only be achieved for dimensions which are multiples of eight.

3.1 Definition of Classleader Root Vectors

A *classleader root vector* is defined in analogy to (4) but contains only positive real valued amplitudes $\mu_l \geq 0$. Given a classleader root vector, sets of classleader vectors described earlier can be constructed by combining the classleader root vector coordinates with a specific distribution of positive and negative signs. However, in order to fulfill the lattice constraint (2), a *sign parity condition* must be considered for two classes of classleader root vectors, described in the following. In analogy to Section 2, all codevectors which can be produced based on a specific classleader root vector form an *equivalence root class*.

Type A classleader root vectors fulfill the constraint defined as the first part of (2) (the definition of the D_8 lattice). According to this definition, given a valid vector $\mathbf{x_A} = [x_{A,0} \quad \cdots \quad x_{A,7}]$ (a vector which fulfills the lattice constraint (2)), another valid vector $\mathbf{x_A}'$ can be constructed by inverting one sign of a vector coordinate at an arbitrary position $i_{A,0}$ of $\mathbf{x_A}$ since

$$(\sum_{i=0}^{n-1} x'_{A,i}) \bmod 2 = ((\sum_{i=0}^{n-1} x_{A,i}) - 2 \cdot x_{A,i_{A,0}}) \bmod 2 \equiv 0 \qquad (5)$$

and $x_{A,i_{A,0}} \in \mathbb{Z}$. Since the inversion of a sign of a coordinate does not change the absolute value of a vector, vectors $\mathbf{x_A}'$ and $\mathbf{x_A}$ are located on the same shell. As a conclusion, valid codevectors are produced from type A classleader root vectors by setting arbitrary combinations of positive and negative signs at all vector coordinates, followed by an (optional) permutation of the vector coordinates and normalization.

Type B classleader root vectors fulfill the constraint defined as the second part of (2) (the definition of the **shifted** D_8 lattice). According to this definition, given a valid vector $\mathbf{x_B} = [x_{B,0} \quad \cdots \quad x_{B,7}]$ (a vector fulfilling this definition), a vector $\mathbf{x_B}'$ produced by inverting one sign at an arbitrary coordinate position $i_{B,0}$ of $\mathbf{x_B}$ would **not** be valid with respect to the definition in (4). By inverting the signs at **two** different vector coordinates $i_{B,0}$ and $i_{B,1}$ of $\mathbf{x_B}$, however, another valid vector $\mathbf{x_B}''$ can be constructed from $\mathbf{x_B}$ to fulfill the codevector construction constraint since

$$\begin{aligned}(\sum_{i=0}^{n-1} x''_{B,i}) \bmod 2 &= \left((\sum_{i=0}^{n-1} x_{B,i}) - 2 \cdot x_{B,i_{B,0}} - 2 \cdot x_{B,i_{B,1}}\right) \bmod 2 \\ &= (0-1-1) \bmod 2 \equiv 0. \end{aligned} \qquad (6)$$

Since the inversion of a sign of a coordinate does not change the absolute value of a vector, vectors $\mathbf{x_B}''$ and $\mathbf{x_B}$ are located on the same shell.
As a conclusion, valid vectors are produced from type B classleader root vectors by setting such combinations of positive and negative signs that fulfill a *sign parity constraint*, followed by an (optional) coordinate permutation and normalization. The sign parity constraint can be *even* or *odd* according to the definition

$$\text{parity}(\mathbf{x}) = \left(\sum_{i=0}^{n-1} \text{sign}(x_i)\right) \bmod 2 = \begin{cases} 1 & \text{odd} \\ 0 & \text{even} \end{cases} \qquad (7)$$

with

$$\text{sign}(x_i) = \begin{cases} 0 & \text{if } x_i >= 0 \\ 1 & \text{if } x_i < 0 \end{cases}. \qquad (8)$$

An example to demonstrate how groups of classleader vectors can be expressed by means of 2 type A classleader root vectors and 1 type B classleader root vector with odd sign parity constraint is exemplified in Table 2.

3.2 Nearest-Neighbor Quantization

Note that following the gain-shape approach (1), the vector to be quantized is assumed to be normalized to have unit absolute value and is therefore written as $\mathbf{c}$ in the remainder of this section. The classleader root vectors will be denoted as $\tilde{\mathbf{x}}_m$ with m as the classleader root index and —by definition— all with the same (not necessarily unit) absolute value. The SVQ codebook is populated with codevectors $\tilde{\mathbf{c}}_i$ which are derived from the classleader root vectors as described in Section 3.1.
A full nearest-neighbor quantization procedure could be realized such that, given a vector $\mathbf{c}$ to be quantized and the SVQ codevectors as $\tilde{\mathbf{c}}_i$, a *similarity metric* $\mathscr{M}_i$

$$\mathscr{M}_i = \mathbf{c}^{\mathrm{T}} \cdot \tilde{\mathbf{c}}_i \qquad (9)$$

is evaluated at first for all codevector entries addressed by index $0 \leq i < N$. The optimal candidate codevector is determined as that codevector which leads to the highest similarity metric,

$$i_{\text{opt}} = \arg\max_i \mathscr{M}_i. \qquad (10)$$

The proposed novel nearest-neighbor quantization procedure to be described in the following takes advantage of the representation of all valid codevectors by means of type A and B classleader root vectors and utilizes a sign distribution as well as a permutation matrix. Since only the q classleader root vectors must be evaluated, it can be realized with a complexity and memory consumption which is significantly lower than that of the full search approach (N candidates) and that of the conventional approach [2] (p candidates).
Given the vector to be quantized as $\mathbf{c}$, for the quantization procedure at first all signs are separated from the magnitudes to produce vectors

$$\begin{aligned}\mathbf{c}_{\text{mag}} &= [\,|\,c_0\,| \quad |\,c_1\,| \quad \cdots \quad |\,c_7\,|\,]^{\mathrm{T}} \\ \mathbf{c}_{\text{sign}} &= [\,\text{sign}(c_0) \quad \text{sign}(c_1) \quad \cdots \quad \text{sign}(c_7)\,]^{\mathrm{T}}, \end{aligned} \qquad (11)$$

involving the definition of the sign function from (8). In the next step, the amplitudes are rearranged involving the $n \times n$ permutation matrix $\mathbf{P_c}$ to produce

$$\begin{aligned}\mathbf{c}^*_{\text{mag}} &= \mathbf{P_c} \cdot \mathbf{c}_{\text{mag}} \\ \mathbf{c}^*_{\text{sign}} &= \mathbf{P_c} \cdot \mathbf{c}_{\text{sign}}. \end{aligned} \qquad (12)$$

ISBN 978-3-8007-3300-2

Table 2: Type A and B classleader root vectors and corresponding classleader vectors for E_8.

Classleader root vector	Associated classleader vectors
$\begin{bmatrix}1&0&0&0&0&0&0&0\end{bmatrix}^{\mathrm{T}}$ Type A	$\begin{bmatrix}1&0&0&0&0&0&0&0\end{bmatrix}^{\mathrm{T}}$ $\begin{bmatrix}0&0&0&0&0&0&0&-1\end{bmatrix}^{\mathrm{T}}$
$\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&0&0&0&0\end{bmatrix}^{\mathrm{T}}$ Type A	$\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&0&0&0&0\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&0&0&0&0&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&0&0&0&0&-\frac{1}{2}&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{1}{2}&0&0&0&0&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}0&0&0&0&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$
$\begin{bmatrix}\frac{3}{4}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$ Type B (odd parity)	$\begin{bmatrix}\frac{3}{4}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{3}{4}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{3}{4}&\frac{1}{2}&\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{3}{4}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&-\frac{3}{4}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{3}{4}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{3}{4}\end{bmatrix}^{\mathrm{T}}$, $\begin{bmatrix}\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{1}{2}&-\frac{3}{4}\end{bmatrix}^{\mathrm{T}}$

such that all coordinates in $\mathbf{c}^*_{\mathrm{mag}}$ are in **decreasing order**.
Instead of computing the *similarity metric* metric for all codevectors following the full nearest-neighbor search approach, only the classleader root vectors are evaluated. The similarity metric is hence written as

$$\mathscr{M}_m = {\mathbf{c}^*_{\mathrm{mag}}}^{\mathrm{T}} \cdot \tilde{\mathbf{x}}^m. \tag{13}$$

with the index $m \in \{0,\ldots,q-1\}$ to address all classleader root vectors and $\tilde{\mathbf{x}}^m$ as the classleader root vector with index m (either type A or type B). Considering the computation of the similarity metric, however, the sign parity constraint discussed earlier demands special consideration:

Type A Classleader Root Vector:

The similarity metric (13) is based on the magnitude vectors $\mathbf{c}^*_{\mathrm{mag}}$ and on the classleader root vectors $\tilde{\mathbf{x}}^m$ which are, by definition, also composed of positive coordinate values only. Considering that a classleader root vector represents a group of codevectors by combining the magnitudes with signs at arbitrary positions and a permutation of coordinates afterwards, the computed similarity metric (13) is the maximum achievable similarity metric value for all codevectors to be produced from the type A classleader root vector. This maximum similarity is achievable if the classleader root vector is combined with positive and negative signs such that it matches the vector $\mathbf{c}^*_{\mathrm{sign}}$. The resulting maximum similarity metric value for the group of codevectors represented by the classleader root vector (equivalence root class) is

$$\mathscr{M}^*_m = \mathscr{M}^{(A)}_m = \mathscr{M}_m. \tag{14}$$

The optimal codevector within this equivalence root class is constructed from the classleader root vector by adding signs at positions identical to the sign distribution of vector $\mathbf{c}^*_{\mathrm{sign}}$, permutation (inverse to matrix $\mathbf{P_c}$) and normalization afterwards.

Type B Classleader Root Vector:

Given a type B classleader root vector, the similarity metric (13) is the maximum among the similarities related to all codevectors that can be constructed from the classleader root vector. Due to the sign parity constraint, however, not all sign combinations are allowed. Therefore two cases must be distinguished:

- Case 1: $\mathrm{parity}(\mathbf{c}^*_{\mathrm{sign}}) = \mathrm{parity}(\tilde{\mathbf{x}}_m)$

  The optimal distribution of signs can be achieved so that the maximum similarity metric for the group of codevectors represented by this classleader root vector is

  $$\mathscr{M}^*_m = \mathscr{M}^{(B)}_m = \mathscr{M}_m. \tag{15}$$

  The optimal codevector within this equivalence root class is constructed from the classleader root vector by adding signs at positions identical to the sign distribution of vector $\mathbf{c}^*_{\mathrm{sign}}$, permutation (inverse to matrix $\mathbf{P_c}$) and normalization afterwards.

- Case 2: $\mathrm{parity}(\mathbf{c}^*_{\mathrm{sign}}) \neq \mathrm{parity}(\tilde{\mathbf{x}}_m)$

  If the parity of the signs related to the input vector does not comply with the classleader root vector sign parity constraint, the sign of one coordinate with unknown index j_0 of the classleader vector must be different to that in $\mathbf{c}^*_{\mathrm{sign}}$. As a consequence, the similarity metric (13) must be updated by

  $$\mathscr{M}^*_m = \mathscr{M}^{(B)}_m = \mathscr{M}_m - 2 \cdot c^*_{\mathrm{mag},j_0} \cdot \tilde{x}_{m,j_0} \tag{16}$$

  with c^*_{mag,j_0} and $\tilde{x}_{m,j_0}$ as single vector coordinate values at the position j_0. The maximum modified metric is achieved for index

  $$j_0 = \underset{0 \le j < n : \tilde{x}_{m,j} \neq 0}{\arg\min} \; c^*_{\mathrm{mag},j} \cdot \tilde{x}_{m,j}. \tag{17}$$

  Since in both vectors $\mathbf{c}^*_{\mathrm{mag}}$ and $\tilde{\mathbf{x}}_m$ the magnitudes are in decreasing order, the minimum is given for the index j_0 of the last non-zero vector coordinate of $\tilde{\mathbf{x}}_m$. The optimal codevector is hence the classleader root vector combined with the distribution of signs identical to that given by vector $\mathbf{c}^*_{\mathrm{sign}}$ but with a sign inversion at the j_0-th vector coordinate, followed by permutation (inverse to matrix $\mathbf{P_c}$) and normalization.

3.2.1 The Optimal Classleader Root Vector

In order to find the optimal codevector, the similarity metric $\mathscr{M}^*_m$ is evaluated for all q classleader root vectors according to (14),(15) or (16). The optimal among the classleader root vectors is the one with index

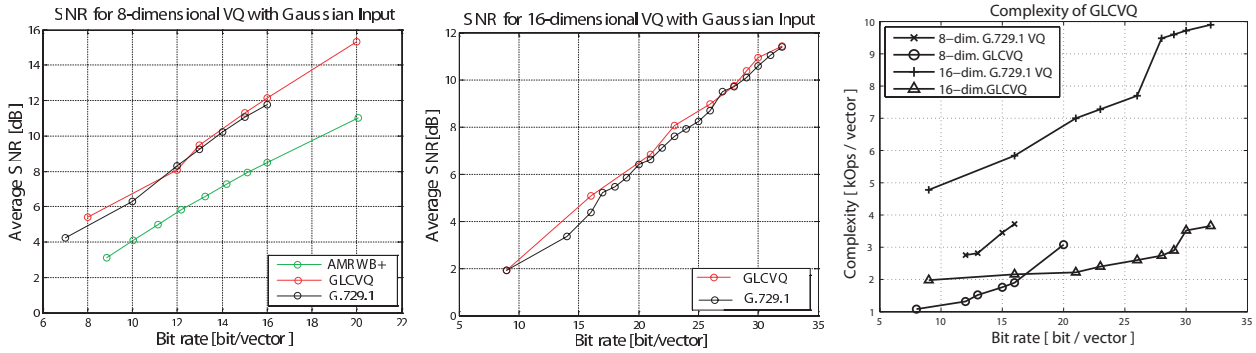
$$m_Q = \underset{m}{\arg\max} \; \mathscr{M}^*_m \tag{18}$$

which can be transformed into the optimal codevector as described in Section 3.2.
Note that in this section for the computation of the similarity metric the classleader root vectors are not normalized to have unit value. This, however, does not have any impact on the quantization performance since all classleader root vectors have identical absolute value.

3.3 Codevector-to-Index-Mapping

Once the optimal codevector has been computed, this information must be transformed into a codevector index. In order to retain a high quantization efficiency, quantization index offset tables are stored for all classleader root vectors. In order to transform a specific permutation of a classleader vector into a quantization index, in the literature, the *Schalkwijk* algorithm has been proposed [13]. In [3], a more efficient alternative has been used. These aspects, however, shall not be discussed in detail here.

ISBN 978-3-8007-3300-2



(a) (b) (c)

Figure 1: (a) and (b): SNR performance of 8- and 16-dimensional GLCVQ compared with lattice VQ from ITU-T Rec. G.729.1 [6, 11] (and 3GPP AMR-WB+ [1]) — (c): Comparison of computational complexity measured in 1000 weighted operations per quantized vector (encoder and decoder).

4 Evaluation & Test Results

The quantization performance and computational complexity of GLCVQ has been compared with the lattice-based SVQ which is used for transform coding in the TDAC module of ITU-T Rec. G.729.1, see Fig. 1. The GLCVQ achieves a slightly better signal-to-quantization-noise-ratio than the reference VQ module which is in fact close to the theoretical optimum for the considered vector dimensions of 8 and 16 [7]. However, a considerable reduction in computational complexity ($\approx$ factor $2-3.5$) is achieved which is due to the efficient representation of the code in terms of classleader root vectors and the particularly efficient indexing procedure of [3] for the lattice points (which is not described in this paper, though). On the other hand, since the resulting codebooks do not represent embedded codes, a little flexibility is sacrificed concerning the available bit rates.

The GLCVQ has been applied to quantize transform coefficients in the Huawei/ETRI candidate codec [3] for the super-wideband extensions of ITU-T Rec. G.729.1 and G.718. Thereby, mainly a vector dimension of 16 has been used. An adaptive bit allocation procedure assigns the available budget to a number of 16-dimensional subbands in the MDCT (Modified Discrete Cosine Transform) domain. The allocation is based on the so called "spectral envelope" which provides a logarithmically quantized gain factor for each subband. In this codec, the GLCVQ contributed significantly to the excellent performance that could be shown in the subjective ITU-T tests. In fact, all requirements for mono input signals were passed with only half of the the allowable (overall) complexity. Consequently, the GLCVQ module has been included in the final ITU-T recommendation as an additional VQ module to enhance the wideband MDCT coefficients.

5 Conclusions

We have proposed a new algorithm for spherical vector quantization with codebooks that are based on shells of the Gosset lattice. While maintaining excellent quantization performance, a considerable reduction of the computational complexity could be achieved by grouping the p equivalence classes of the original algorithm [2] into $q < p$ equivalence root classes. For example, at a bit rate of approximately 2.257 bit/vector, only a fraction of $q/p \approx 6.8\%$ of the candidate vectors have to be evaluated compared to [2]. In addition, the GLCVQ concept offers flexibility w.r.t. the vector dimension n (multiples of 8) while maintaining many favorable properties.

The GLCVQ has been successfully applied for super-wideband speech and audio coding in the candidate codec which is described in [3]. Recently, it has been included in Amd. 6 to ITU-T Rec. G.729.1.

Literature

[1] 3GPP TS 26.290. Extended adaptive multi-rate - wideband (AMR-WB+) codec; transcoding functions, 2004.

[2] J. P. Adoul, C. Lamblin, and A. LeGuyader. Baseband speech coding at 2400 bps using spherical vector quantization. In *Proc. of IEEE ICASSP*, San Diego, CA, USA, March 1984.

[3] B. Geiser, H. Krüger, H. W. Löllmann, P. Vary, D. Zhang, H. Wan, H.T. Li, and L.B. Zhang. Candidate proposal for ITU-T super-wideband speech and audio coding. In *Proc. of IEEE ICASSP*, Taipei, Taiwan, April 2009.

[4] ITU-T Rec. G.718. Frame error robust narrowband and wideband embedded variable bit-rate coding of speech and audio from 8-32 kbit/s, 2008.

[5] ITU-T Rec. G.719. Low-complexity, full-band audio coding for high-quality, conversational applications, 2008.

[6] ITU-T Rec. G.729.1. G.729 based embedded variable bit-rate coder: An 8-32 kbit/s scalable wideband coder bitstream interoperable with G.729, 2006.

[7] H. Krüger. *Low Delay Audio Coding Based on Logarithmic Spherical Vector Quantization*. PhD thesis, RWTH Aachen, 2010.

[8] H. Krüger, R. Schreiber, B. Geiser, and P. Vary. On Logarithmic Spherical Vector Quantization. In *Proc. of International Symposium on Information Theory and its Applications (ISITA)*, Auckland, New Zealand, December 2008.

[9] H. Krüger and P. Vary. SCELP: Low Delay Audio Coding with Noise Shaping based on Spherical Vector Quantization. In *Proc. of EUSIPCO*, Florence, Italy, September 2006.

[10] B. Matschkal, F. Bergner, and J.B. Huber. Joint Signal Processing for Spherical Logarithmic Quantization and DPCM. In *Proc. of 4th International Symposium on Turbo Codes and Related Topics*, Munich, Germany, April 2006.

[11] S. Ragot et al. ITU-T G.729.1: An 8-32 kbit/s scalable coder interoperable with G.729 for wideband telephony and Voice over IP. In *Proc. of IEEE ICASSP*, Honolulu, Hawai'i, USA, April 2007.

[12] M.J. Sabin and R.M. Gray. Product Code Vector Quantizers for Waveform and Voice Coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32(3):474–488, June 1984.

[13] J. P. M. Schalkwijk. An algorithm for source coding. *IEEE Transactions on Information Theory*, 18(3):395 – 399, May 1972.

[14] D. Slepian. Permutation Modulation. *Proceedings of the IEEE*, 53(3):228–236, 1965.

[15] T. Vaillancourt et al. ITU-T EV-VBR: A robust 8-32 kbit/s scalable coder for error prone telecommunications channels. In *Proc. of EUSIPCO*, Lausanne, Switzerland, August 2008.

[16] M. Xie et al. ITU-T G.719: A new low-complexity full-band (20 kHz) audio coding standard for high-quality conversational applications. In *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 265–268, New Paltz, NY, USA, October 2009.

ISBN 978-3-8007-3300-2