Logarithmic Cubic Vector Quantization

C. Rohlfing, H. Krüger and P. Vary

Institute of Communication Systems and Data Processing
RWTH Aachen University, 52074 Aachen, Germany
Email: {rohlfing,krueger,vary}@ind.rwth-aachen.de
Web: www.ind.rwth-aachen.de

Abstract

In this paper a novel type of gain-shape vector quantization (GSVQ) is presented, denoted as Logarithmic Cubic Vector Quantization (LCVQ). LCVQ is based on a decomposition of the vector to be quantized into a gain factor and a shape vector which is a normalized version of the input vector. Both components are quantized independently and transmitted to the decoder.
Compared to other GSVQ approaches, in LCVQ the input vectors are normalized such that all shape vectors are located on the surface of the unit hypercube. As a conclusion, the shape vector quantizer can be realized based on uniform scalar quantizers. This yields low computational complexity as well as high memory efficiency even in case of very high vector dimensions.
In order to demonstrate the coding efficiency of the proposed quantization scheme, LCVQ is compared to existing quantization schemes, in particular the recently proposed Logarithmic Spherical Vector Quantization (LSVQ) [1].

1 Introduction

In general, the quantizer is the key element in compression schemes for lossy speech and audio coding. In the design of a quantizer in practice, a good trade-off between complexity and quantization performance has to be achieved. On the one hand, scalar quantizers can be realized with low complexity but have only a moderate quantization performance. On the other hand, vector quantization techniques show promising results close to the theoretically achievable performance but have larger implementation costs.
One of the most prominent examples of a highly efficient scalar quantizer is the well-known Adaptive Quantization Backward (AQB) [2] which dates back to the early days of speech coding and has become part of numerous speech coding standards such as, e.g., the G.726 speech codec [3]. Most of today's speech codecs make use of high-dimensional algebraic vector quantizers [4]. Due to the employment of sparse codebooks, however, algebraic vector quantizers as used in today's speech codecs are only suitable for very low bit rates (≤ 1 bit per vector dimension).
Recently, Logarithmic Spherical Vector Quantization (LSVQ), has been theoretically investigated in [5] and [1]. In that context, LSVQ represents a **class** of gain-shape vector quantizers (GSVQs) which may involve different practical realizations of Spherical Vector Quantizers (SVQ), e.g., described in [6] and [7]. In [8] and [9] novel realizations of LSVQ have been proposed which achieve a high quantization performance with low computational cost based on the well-known Gosset Lattice and the so-called Apple-Peeling approach, respectively. However, the proposed approaches for LSVQ require the storage of vector codebooks which are computed in offline manner and also limit the available bit rates to low values ($\leq 2-3$ bits per sample).

Logarithmic Cubic Vector Quantization (LCVQ) is a novel type of gain-shape vector quantization which offers some advantages over LSVQ and AQB: It achieves high quantization performance and operates with low complexity. Furthermore it can be realized with high memory efficiency even in case of very high input-vector dimensions and higher bit rates since no vector codebook is required. In addition to that, it is flexible regarding the possibility to adjust the bit rate at runtime.
In this paper, LCVQ shall be investigated and compared to LSVQ and AQB: A generalization of gain-shape vector quantization is given in Section 2 as well as brief descriptions of logarithmic scalar quantization (LSQ), LSVQ and the novel LCVQ. Section 3 briefly discusses AQB which shows interesting similarities to GSVQ and in particular the previously mentioned LSVQ and LCVQ. The quantization performances of all quantization techniques are evaluated in Section 4. Finally, conclusions are given in Section 5.

2 Generalized Gain-shape Vector Quantization

In gain-shape vector quantization (GSVQ) [10], the input vector $\mathbf{x} \in \mathbb{R}^L$ with dimension L is decomposed into a gain factor $g \geq 0$ and a shape vector $\mathbf{c} \in \mathbb{R}^L$ which are then quantized independently by means of a scalar quantizer for g and a vector quantizer for $\mathbf{c}$. Both parts are transmitted to the decoder (refer to Figure 1).
Gain value and shape vectors are computed as

$$g = \|\mathbf{x}\|_p = \left(\sum_{i=1}^{L} |x_i|^p\right)^{\frac{1}{p}} \qquad \text{and} \qquad \mathbf{c} = \frac{1}{g} \cdot \mathbf{x} \tag{1}$$

with $\|\mathbf{x}\|_p$ denoting the p-norm of a vector.

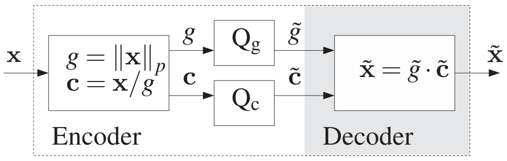


Figure 1: Parallel quantization of the gain and shape component with $p = 2$ and $\mathrm{Q_c} = \mathrm{Q_{svq}}$ for LSVQ and $p = \infty$, $\mathrm{Q_c} = \mathrm{Q_{cvq}}$ for LCVQ.

In the decoder, the quantized version of the gain factor, $\tilde{g} = \mathrm{Q_g}(g)$, and of the shape-vector, $\tilde{\mathbf{c}} = \mathrm{Q_c}(\mathbf{c})$, are combined to produce the overall reconstruction vector $\tilde{\mathbf{x}}$.
The main difference between LSVQ and LCVQ lies in equation (1) and can be illustrated based on Figure 2 for the example of $L = 2$: In LSVQ, setting $p = 2$ results in a

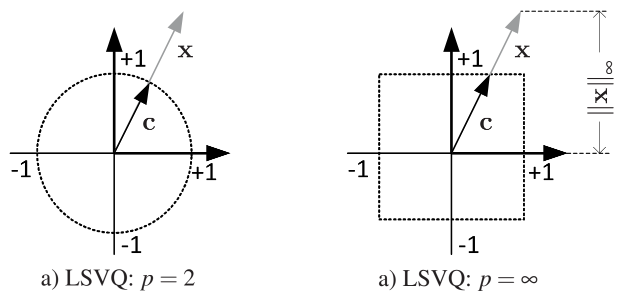


Figure 2: Transforming $\mathbf{x}$ into $\mathbf{c}$: Projection onto the surface of a unit sphere (a) and a unit cube (b) due to different p-norms in LSVQ ($p = 2$) and LCVQ ($p = \infty$), respectively.

normalization of input vector $\mathbf{x}$ to its Euclidean norm leading to the shape vectors $\mathbf{c}$ being located on the surface of the L-dimensional unit **hypersphere**. In contrast to this, LCVQ makes use of the uniform (maximum) norm with $p = \infty$ instead of the Euclidean norm. As a conclusion, the normalized vectors $\mathbf{c}$ are located on the surface of the L-dimensional unit **hypercube**.
The advantage of LCVQ compared to LSVQ lies in the fact that the design of a high-dimensional vector quantizer for vectors located on the surface of a high-dimensional geometric shape can be realized with significantly lower complexity and higher flexibility for the hypercube than for the hypersphere.

2.1 Logarithmic Quantization of the Gain Factor

Both approaches, LCVQ and LSVQ, have in common that the gain factor g is quantized using Logarithmic Scalar Quantization (LSQ) Q_g. In this paper, the A-law companding algorithm is used with the number of reconstruction levels N_g, compression factor A and stepsize Δg, as described in [2]. It can be shown, that in the case of LSQ, the stepsize scales with the reconstruction value $\tilde{g}$ [11]:

$$\Delta g(\tilde{g}) = \frac{1+\ln(A)}{N_g} \cdot \tilde{g}\,. \tag{2}$$

The signal-to-noise ratio (SNR) of LSQ is independent of the input vector probability density function (PDF)

$$\begin{aligned}\mathrm{SNR}_{\mathrm{lsq,A}}|_{\mathrm{dB}} &= 6.02 \cdot R_{\mathrm{eff}} + 10\log_{10}(3) \\ &\quad - 20\log_{10}(1+\ln(A))\end{aligned} \tag{3}$$

with R_{eff} denoting the bit rate per sample of the quantizer. This independence comes at the price of a penalty-term of $10\log_{10}(3) - 20\log_{10}(1+\ln(A))$ compared to the well-known 6 dB-per-bit-rule [11]. LSQ can be considered as a special case of LSVQ and LCVQ for input-vector dimension $L = 1$. Therefore, it will be referenced in Section 4 as the *lower limit* for the SNR plots for both LCVQ and LSVQ.

2.2 Logarithmic Spherical VQ (LSVQ)

In LSVQ, the shape vectors $\mathbf{c}$ are quantized using a spherical vector quantizer (SVQ) denoted by $Q_c = Q_{\mathrm{svq}}$. The corresponding SVQ vector codebook is composed of N_{svq} spherical codevectors which are located on the surface of the L-dimensional unit hypersphere. The overall design goal of the codebook of Q_{svq} is to distribute the codevectors uniformly over the unit sphere surface.
In [1], it is shown, that using LSQ for quantization of the gain factor $g = \|\mathbf{x}\|_2$ and SVQ for the shape vector $\mathbf{c}$, the overall quantization SNR is independent of the PDF of the input vector,

$$\begin{aligned}\mathrm{SNR}^{(I)}_{\mathrm{lsvq}} &= \frac{(N_{\mathrm{svq}} \cdot N_g)^{\frac{2}{L}}}{C_{\mathrm{lsvq}} \cdot \pi} \cdot \left(\frac{\Gamma\left(\frac{L}{2}\right)}{2 \cdot (1+\ln(A))} \right)^{\frac{2}{L}} \\ &\quad \cdot \underbrace{\frac{\int\limits_{\mathbf{x}\in\mathbb{R}^L} p(\mathbf{x}) \cdot \|\mathbf{x}\|_2^2 \,\mathrm{d}\mathbf{x}}{\int\limits_{\tilde{\mathbf{x}}\in\mathbb{R}^L} p(\tilde{\mathbf{x}}) \cdot \|\tilde{\mathbf{x}}\|_2^2 \,\mathrm{d}\tilde{\mathbf{x}}}}_{\approx 1}\,.\end{aligned} \tag{4}$$

Equation (4) was derived from the normalized quantizer point-density function under high bit rate assumptions and is a function of the overall number of LSVQ codevectors

$$N_{\mathrm{lsvq}} = N_{\mathrm{svq}} \cdot N_g. \tag{5}$$

It is, however, only a qualitative formula for the SNR due to C_{lsvq} denoting an unknown quantization cell form-factor. $\Gamma(z) = \int_0^\infty e^{-t} \cdot t^{z-1} \mathrm{d}t$ denotes the gamma-function [12].
Assumptions about the shape of the resulting SVQ quantization cells as well as high bit rate approximations finally yield the following quantitative formula for an estimate of the maximum achievable SNR in dB in [1] as

$$\begin{aligned}\mathrm{SNR}^{(II)}_{\mathrm{lsvq}}|_{\mathrm{dB}} &= 6.02 \cdot R_{\mathrm{eff}} \\ &\quad - 10\log_{10}\left(\frac{L}{(L+1)^{\frac{L-1}{L}}} \cdot \left[2 \cdot \sqrt{\pi} \cdot \frac{\Gamma\left(\frac{L+1}{2}\right)}{\Gamma\left(\frac{L}{2}\right)} \right]^{\frac{2}{L}} \right. \\ &\quad \left. \cdot \left[\frac{(1+\ln(A))^2}{12} \right]^{\frac{1}{L}} \right)\,.\end{aligned} \tag{6}$$

This formula is used in Section 4 to compare the performance of LSVQ with the novel LCVQ scheme and is expressed as a function of R_{eff} which is the effective bit rate per vector dimension based on which the number of LSVQ codevector (5) can be computed as

$$N_{\mathrm{lsvq}} = 2^{R_{\mathrm{eff}} \cdot L} = N_g \cdot N_{\mathrm{svq}}. \tag{7}$$

The maximum performance is reached for

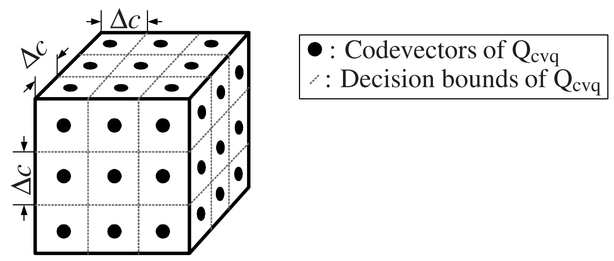
$$\lim_{L\to\infty} \mathrm{SNR}^{(II)}_{\mathrm{lsvq}}|_{\mathrm{dB}} = 6.02 \cdot R_{\mathrm{eff}} \tag{8}$$

which will be referred to as the *upper limit* in the SNR plots in Section 4. The design of spherical vector-codebooks used for Q_{svq} is rather complex and shall not be discussed in detail in this paper. Due to practical reasons, in most cases, vector dimensions higher than $L = 16$ can be realized only for low bit rates ($\leq 2-3$ bits per vector dimension).

2.3 Logarithmic Cubic VQ (LCVQ)

In Logarithmic Cubic Vector Quantization, the input vector is normalized by its uniform norm

$$g = \|\mathbf{x}\|_\infty = \max(|x_1|, \ldots, |x_L|) = x_{l_0} \tag{9}$$



Δc
Δc
Δc
● : Codevectors of Q_{cvq}
⟋ : Decision bounds of Q_{cvq}

Figure 3: Exemplary hypercube for $L = 3$ with codevectors and quantization decision bounds of Q_{cvq} for $N_c = 3$.

with $l_0 = \arg\max_l(|x_l|)$ denoting the index of the maximum element value of $\mathbf{x}$ such that the normalized vector component $c_{l_0} = \pm 1$. As a result, the shape vectors $\mathbf{c}$ are lying on the surface of the L-dimensional unit hypercube with edge length equal to 2.
The quantizer of the shape vector $Q_c = Q_{\mathrm{cvq}}$ can be realized in a very efficient way as it consists of $L-1$ uniform scalar quantizers Q_{sq} for each dimension with index $l \neq l_0$

$$\tilde{c}_l = \begin{cases} Q_{\mathrm{sq}}(c_l), & \text{if } l \neq l_0 \,, \\ \pm 1, & \text{if } l = l_0 \,. \end{cases} \tag{10}$$

Each uniform scalar quantizer Q_{sq} has the same number of reconstruction levels N_c and the stepsize

$$\Delta c = \frac{2}{N_c} \,. \tag{11}$$

The total number of LCVQ reconstruction values is calculated as

$$N_{\mathrm{lcvq}} = 2^{R_{\mathrm{eff}} \cdot L} = N_g \cdot N_{\mathrm{cvq}} = N_g \cdot 2 \cdot L \cdot N_c^{L-1} \tag{12}$$

with N_{cvq} denoting the number of reconstruction values per cube shell and $2 \cdot L$ as the number of surfaces of a L-dimensional hypercube. An example of the hypercube covered by quantization cells and reconstruction vectors related to Q_{cvq} for $L = 3$, $N_c = 3$ and $N_{\mathrm{cvq}} = 2 \cdot L \cdot N_c^{L-1} = 54$ is given in Figure 3.
Using only scalar quantizers, the LCVQ can be realized with very low memory and implementation cost. Also, since no codebook is involved, the bit rate can be easily adapted during runtime.
In comparison to LSVQ, LCVQ has no practical limitations regarding the input-vector dimension as well as the effective overall bit rate.

3 Adaptive Quantization Backward (AQB)

Compared to the approaches for vector quantization described in the previous sections, the Adaptive Quantization Backward (AQB) approach is a **scalar** quantizer which is for example described in [2] and shown in Figure 4: Instead of operating on vectors $\mathbf{x}$ of the input signal, in AQB, a uniform scalar quantizer Q_{sq} operates directly on the signal $x(k)$ with a quantization stepsize Δx which is dynamically adapted to an estimate of the instantaneous signal power $\tilde{\sigma}_x^2$. The output of the scalar quantizer is an integer representation $Z(k)$ of the quantized input signal which is transmitted to the decoder for the reconstruction of the input signal as $\tilde{x}(k)$.

The term "backward" comes from the fact that the estimate of the instantaneous signal power $\tilde{\sigma}_x$ for the adaptation of the quantizer stepsize is derived from the **quantized** input signal $\tilde{x}(k)$ and hence from the quantization history in the past: Given k as the current time index, the instantaneous signal power $\tilde{\sigma}_x^2$ is estimated recursively using the integer representation of the quantized input signal $Z(k-1)$ transmitted at time index $k-1$:

$$\frac{\Delta x(k)}{\Delta x(k{-}1)} = \frac{\tilde{\sigma}_x^2(k)}{\tilde{\sigma}_x^2(k{-}1)} = f\{Z(k-1)\} = M(k-1) \,. \tag{13}$$

$M(k)$ denotes the so-called stepsize multiplier which is a function of $Z(k)$. $M(k)$ is in general calculated in advance. For all evaluations in this paper, values of $M(k)$ for speech signals have been taken from [11].

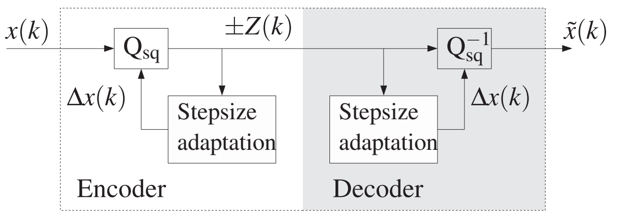


Figure 4: Block diagram of the adaptive quantization backward scheme.

The number of reconstruction levels of Q_{sq} is $N_{\mathrm{aqb}} = 2^{R_{\mathrm{eff}}}$ with R_{eff} as the effective bit rate in analogy to the previous sections. The stepsize-adaptation block is realized in both encoder and decoder and uses equation (13) to estimate the instantaneous stepsize $\Delta x(k)$ on both sides. On the decoder side, the block Q_{sq}^{-1} represents the operation to convert the integer representation of the quantized input signal into the reconstructed signal $\tilde{x}(k)$.
The AQB shall be considered as a reference quantizer in Section 4 as it shows similarities to LCVQ and LSVQ in the sense that all these algorithms have in common to estimate the instantaneous signal variance: AQB uses a recursive estimation of the instantaneous variance, whereas LCVQ and LSVQ estimate a gain factor based on a segmentation of the input signal into short vectors of length L (the input vectors to be quantized).

4 Evaluation

For the evaluation of the proposed quantization schemes, a large number of signal vectors was generated following a normal or uniform distribution. These vectors were then quantized with LCVQ for different vector dimensions $L \in \{2, 8, 48\}$ or AQB. For the assessment of LCVQ and AQB the signal-to-quantization-noise ratio (SNR) was calculated. In case of the LCVQ the optimal allocation of bits for the quantizer of the gain and the vector quantizer for the normalized input vectors (N_g and N_c, respectively) was determined experimentally for each value of L. In order to consider also the LSVQ, the theoretical values according to Equation (6) were taken into account.
The simulated SNR values were measured for different bit rates $R_{\mathrm{eff}} \in [1.5, 5]$ and a companding factor $A = 5000$ for the logarithmic quantizers of the gain factors. The latter was found in [5] as a reasonable trade-off between dynamic range and performance for audio signals. Figure 5

shows the measured SNR values in dB as a function of the effective bit rate R_{eff} for uniformly and in Figure 6 for normally distributed input vectors. LCVQ achieves almost the

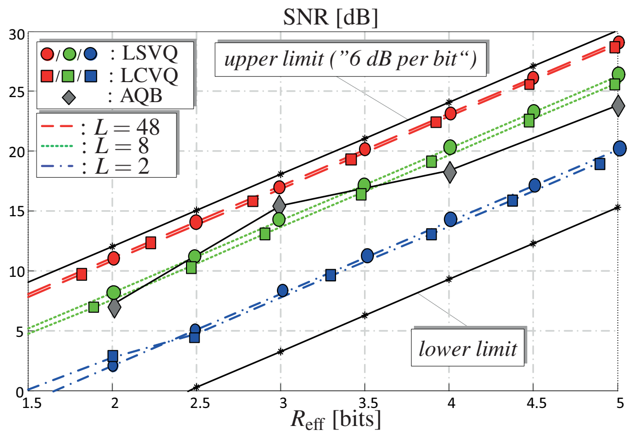


Figure 5: SNR for uniformly distributed input vectors $\mathbf{x}$.

same SNR as LSVQ for the uniform input-vector distribution for all input-vector dimensions L. LCVQ shows better results than AQB for high bit rates and above vector dimension $L = 8$.

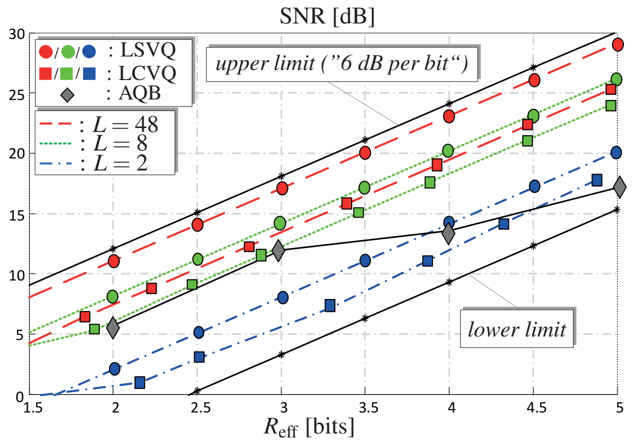


Figure 6: SNR for input vectors $\mathbf{x}$ following a normal distribution with mean $\mu_x = 0$ and variance $\sigma_x^2 = 1$.

For normally distributed input vectors, LSVQ outperforms LCVQ for all vector dimensions L, whereby the difference between the LSVQ and LCVQ SNR increases with L. For vector dimension $L = 8$ and lower bit rates (≤ 3 bits per sample), LCVQ provides a slightly higher performance than AQB. For higher bit rates, the performance of the LCVQ follows a 6-dB-per-bit gradient whereas the increase of quantization SNR over bit rate is much lower in case of the AQB.
The measured results are bounded by the *lower limit* which is the performance of LCVQ and LSVQ for a vector dimension $L = 1$ (LSQ, refer to Section 2.1) and the *upper limit* which is the asymptotic performance of the LSVQ for infinite vector dimension, $L \to \infty$.
In the comparison of results for uniform and normal distribution of vectors to be quantized, it is obvious that the grouping of codevectors on shells of hypercubes in the LCVQ approach is not optimal for spherically distributed sources such as the normal distribution [1]. Nevertheless, the LCVQ is a good alternative to the LSVQ since it can be realized with significantly lower complexity and memory consumption. Also, it is not feasible to operate the LSVQ above a vector dimension of $L = 16$ and a bit rate of more than 2 bits per sample. Compared to this, the LCVQ offers a significantly higher flexibility as it can be used for very large dimensions and also for bit rates higher than 3 bits per sample.

5 Conclusions

In this paper, a novel gain-shape vector quantization technique, "Logarithmic Cubic Vector Quantization", was proposed. In comparison to other approaches for gain-shape vector quantization, the LCVQ employs the uniform norm to find a gain factor and normalized representations of the vectors to be quantized which are located on the surface of a hypercube. As a result, the LCVQ can be realized with low computational complexity and memory consumption and is also very flexible in terms of the adaptation to different bit rates.
In evaluations of different quantization schemes it was shown that the recently proposed LSVQ which is significantly more complex than LCVQ, shows only slightly better SNR results than LCVQ for uniformly distributed signals. This, however, can be compensated by an increase of the input-vector dimension of the LCVQ, which is not possible for LSVQ. Also, the LCVQ outperforms the well-known AQB already for moderate dimensions and for higher bit rates.

References

[1] H. Krüger, *Low Delay Audio Coding Based on Logarithmic Spherical Vector Quantization*, vol. 25 of *Aachener Beiträge zu digitalen Nachrichtensystemen (Dissertation)*. Wissenschaftsverlag Mainz, 2010.
[2] N. S. Jayant and P. Noll, *Digital Coding of Waveforms, Principles and Applications to Speech and Video*. Englewood Cliffs NJ, USA: Prentice-Hall, 1984.
[3] ITU-T, Rec. G.726, "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)," Dec. 1990.
[4] C. Laflamme, J.-P. Adoul, H. Y. Su, and S. Morissette, "On reducing computational complexity of codebook search in CELP coder through the use of algebraic codes," in *Proceedings of ICASSP*, pp. 177–180, 1990.
[5] H. Krüger, R. Schreiber, B. Geiser, and P. Vary, "On Logarithmic Spherical Vector Quantization," in *Proceedings of ISITA*, 2008.
[6] J. Hamkins, *Design and Analysis of Spherical Codes*. PhD thesis, University of Illinois, 1996.
[7] J. B. Huber and B. Matschkal, "Spherical logarithmic Quantization and its Application for DPCM," in *5th International ITG Conference on Source and Channel Coding(SCC)*, (Erlangen, Germany), pp. 349–356, January 2004.
[8] H. Krüger, B. Geiser, P. Vary, H. T. Li, and D. Zhang, "Gosset lattice spherical vector quantization with low complexity," in *Proceedings of ICASSP*, pp. 485 –488, may 2011.
[9] H. Krüger and P. Vary, "SCELP: Low Delay Audio Coding with Noise Shaping Based on Spherical Vector Quantization," in *Proceedings of European Signal Processing Conference (EUSIPCO), EURASIP 2006*, 2006.
[10] M. Sabin and R. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Trans. on Acoust., Speech and Signal Proc.*, vol. 32, pp. 474 – 488, jun 1984.
[11] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding And Error Concealment*. John Wiley & Sons, 2006.
[12] I. Bronstein and K. Semendjajew, *Taschenbuch der Mathematik, 25.* BG Teubner Verlagsgesellschaft, Stuttgart Leipzig und Verlag Nauka, Moskau, 1991.