Spherical Logarithmic Quantization

Bernd Matschkal and Johannes B. Huber, *Fellow, IEEE.*

***Abstract*—Spherical logarithmic quantization (SLQ) is a vector quantization method for efficiently digitizing analog signals at a high dynamic range and with very low distortion while preserving the original waveform as closely as possible. SLQ is able to operate at a low data rate of e.g. 2 bits per sample and at a very low signal delay of about 10 samples, this corresponds to approximately 200 μs for high quality audio signals. The technique of SLQ is universally applicable (i.e. not restricted to e.g. audio signals) and achieves an efficient digital representation of waveforms with high longterm as well as high segmental Signal-to-Noise Ratios.**

The aim of this paper is to give a detailed description of the SLQ algorithm and to present simulation results on the performance of this new quantization scheme that combines several advantages. After a review of some important basic principles concerning quantization, linear prediction and multidimensional spheres, the SLQ encoder is described. To short vectors of signal samples which are represented in sphere coordinates, logarithmic quantization is applied to the radius and uniform quantization is applied to the angles. This results in the advantage of a constant Signal-to-Noise Ratio over a very high dynamic range at a small loss with respect to the rate-distortion theory. In order to increase the Signal-to-Noise Ratio by exploitation of correlations within the source signal, a solution for the problem of combining this vector quantization scheme with scalar adaptive differential pulse code modulation (ADPCM), i.e. ADPCM with sample by sample backward recursion is presented. Furthermore, an indexing scheme for the quantization cells covering the surface of a multidimensional unit sphere is presented and simulation results using different source signals are given.

***Index Terms*—Dynamic range, logarithmic quantization, low delay audio coding, vector quantization.**

I. Introduction

FOR efficient and robust transmission, processing or storage of analog waveforms, a digital representation offers lots of advantages. Usually, the analog signal is quantized and digitized and thereafter data compression methods are applied exploiting redundancy within the source signal and irrelevance due to special properties of the consumer of the signal. The numerous different methods for digitizing analog signals may be subdivided into two basic categories:

1) Methods where the reconstructed waveform approximates the original one closely, i.e. no exploitation of irrelevance.

2) Non-waveform conserving methods. These approaches are very important in audio and video signal compression, as long as only the subjective quality of the signal at the receiver output is relevant (e.g. exploitation of psychoacoustic masking). Usually, signal processing for exploitation of irrelevance produces a reconstructed waveform rather far away from the original one and additionally introduces a high signal delay (e.g. due to spectral transforms or equivalent block based operations). The quality of these waveform coding methods cannot be expressed by a Signal-to-Noise Ratio (SNR) in the classical sense; it has to be determined in complex performance tests by well-trained human listeners or viewers.

In many applications, such as processing of measurement data, recording of waveforms for further signal processing and real time signal transmission using digital modulation schemes without tolerance of a noticeable signal delay as e.g. for wireless digital stage microphones, non-waveform conserving signal coding schemes are completely useless. Moreover in some realtime applications such as wireless microphones and wireless in-ear stage monitoring systems, an overall delay greater than 8 milliseconds cannot be tolerated since it could distract the performers.

In this paper a new waveform conserving method, the so-called spherical logarithmic quantization (SLQ), is developed in order to meet the following requirements:

- low data rate by exploitation of favorite packing properties of multidimensional lattices (vector quantization)
- extremely high dynamic range, i.e. preservation of a constantly high measurable SNR for variations of 60 dB and more of the short time variance of the analog source signal; this very important e.g. for almost all kinds of microphones
- high segmental SNR for short segments of waveform samples
- insensitivity to special signal parameters such as segmental probability density function etc., and most important
- introduction of an only extremely low signal delay in the order of a few (up to 10) samples.

The paper is organized as follows. Section II recalls some basic aspects concerning quantization, linear prediction and multidimensional spheres as they are essential for the understanding of SLQ. Moreover, related work is quoted and the differences and similarities of SLQ are discussed. Section III introduces SLQ, discusses the arrangement of the quantization cells and presents theoretical results on the quantization noise and the SNR that can be achieved. A detailed explanation of the SLQ algorithm, its extension by means of ADPCM as well as the indexing problem is discussed in section IV followed by simulation results in section V. Conclusions are given in section VI.

Manuscript received July 15, 2008; revised April 23, 2009. First published June 02, 2009; current version published October 23, 2009. This work was presented in part at the 4th and 5th International ITG Conference on Source and Channel Coding (SCC), Erlangen and Munich, Germany, respectively.

B. Matschkal was with the Institute for Information Transmission, 91058 Erlangen, Germany. He is now with the Siemens Audiology, 91058 Erlangen, Germany (e-mail: matschkal@lnt.de).

J. B. Huber is with the Institute for Information Transmission, 91058 Erlangen, Germany (e-mail: jbhuber@lnt.de).

II. Preliminaries and Related Work

If an analog signal (e.g. an audio signal) is transmitted by means of digital transmission, usually pulse code modulation (PCM) is applied. The analog source signal $q(t)$ is sampled and the resulting discrete-time signal $q[k]$ is quantized [1]. The quantized values $\widehat{q}[k]$ are mapped to digital symbols which are transmitted over the digital channel.

A. Scalar Quantization

Scalar quantization at a rate R (average number of bits per sample) divides the considered amplitude range $[-1,+1]$ into

$$M = 2^R \tag{1}$$

usually not equally sized quantization intervals per sample. In the special case of a uniform quantization and assuming small quantization intervals as well as no overloading of the quantizer, the resulting Signal-to-Noise Ratio (SNR) yields

$$\mathrm{SNR} = \frac{\sigma_q^2}{N_\mathrm{Q}} = 3\sigma_q^2 \cdot M^2 \tag{2}$$

$$10\log_{10}(\mathrm{SNR}) = 4.77\ \mathrm{dB} + 10\log_{10}(\sigma_q^2) + R \cdot 6.02\ \mathrm{dB} \tag{3}$$

with σ_q^2 denoting the variance of q and N_Q the average quantization noise power. Equation (3) is also well known as the "6 dB-per-bit rule" of quantization theory.

Uniform quantization is not in general the most effective way to achieve a good performance. Usually, depending on the optimization criterion, one of the various non-uniform quantization schemes is used. The so-called logarithmic quantization is characterized by a high dynamic range i.e. a wide range of the signal level where the SNR is constant. An effective technique, used by Bennet [2], for studying non-uniform quantization is to model the quantizer as a memoryless nonlinearity $k_\mathrm{K}(q)$ ("Compressor") followed by a uniform quantizer (i.e. the ADC) and an inverse "Expander" $k_\mathrm{K}^{-1}(q)$ after the digital-to-analog converter (DAC). The characteristic of the compressor for well-known logarithmic quantization according to the $\mathcal{A}$-law, which has been in use for many years in telecommunications [3], consists of a linear area and two logarithmic areas and yields

$$k_\mathrm{K}(q) = \begin{cases} q \cdot \frac{\mathcal{A}}{1+\ln(\mathcal{A})} & \text{for} \quad 0 \le q < \frac{1}{\mathcal{A}} \\ \left(1 + \frac{\ln(q)}{1+\ln(\mathcal{A})}\right) & \text{for} \quad \frac{1}{\mathcal{A}} \le q \le 1 \\ -k_\mathrm{k}(-q) & \text{for} \quad q < 0 \end{cases} \tag{4}$$

where $\mathcal{A}$ is the usual parameter of this logarithmic quantization, c.f. European PCM Standard (ITU G.711, [4]).

Regarding a medium signal level and $R \gg 1$, the SNR can be expressed by [3]

$$10\log_{10}(\mathrm{SNR}) = R \cdot 6.02\,\mathrm{dB} + 10\log_{10}\left(\frac{3}{\left(1+\ln(\mathcal{A})\right)^2}\right) . \tag{5}$$

The logarithmic quantization provides a constant relative maximum quantization error for signals with power levels greater than

$$-B_1 := 20\log_{10}(1/\mathcal{A}) \quad [\mathrm{dB}] , \tag{6}$$

which are quantized and reconstructed at a SNR given by (5). The parameter $\mathcal{A}$ specifies the quotient of the maximum to the minimum width of quantization intervals and thus indicates the extent of the logarithmic quantization. Equivalently, this parameter is also expressed by the increase of resolution Δn in bit/sample for very small values of q w.r.t. a uniform quantization at the same rate, cf. Table I. $\mathcal{A} = 1$ represents a uniform quantization without any logarithmic aspect, the larger $\mathcal{A}$ is chosen, the larger is the dynamic range. The dynamic range of scalar logarithmic quantization is characterized by its "left edge" $-B_1$ and its "right edge" $-E_1$ for overload, the latter depends on the probability density function (pdf) of the signal, cf. the special case $D = 1$ of Fig. 7.

Within the dynamic range of $B_1 - E_1$ [dB], the SNR is independent of the pdf of the signal which provides an universal applicability of this method. Comparing (5) to the result for a uniformly distributed source signal and using a uniform quantizer or to the rate distortion function for independent, identically distributed (i.i.d.) Gaussian random variables, the term

$$10\log_{10}\left(\frac{3}{\left(1+\ln(\mathcal{A})\right)^2}\right) \tag{7}$$

represents a "SNR-loss by companding", which obviously has to be paid for a wide dynamic range. Hence, an increasing $\mathcal{A}$ decreases the maximum achievable SNR. For instance, selecting $\mathcal{A} = 87.6$ provides an increased resolution of $\Delta n = 4$ bits for small values of q and results in a SNR-loss of $10.0\,\mathrm{dB}$, whereas $\mathcal{A} = 968877.2$ results in $\Delta n = 16$ and a SNR-loss of $18.6\,\mathrm{dB}$.

B. Vector Quantization

The combined quantization and reconstruction of blocks of D samples ("D dimensional vector quantization") gives a higher degree of freedom for choosing the reconstruction values and the decision regions compared to scalar quantization. Moreover, combined processing of several samples gives the possibility of exploiting statistical dependencies among samples. The price for this increased performance is an increase in computational complexity in both rate and number of dimensions.

Lattice quantization was proposed by [5] and can be viewed as a vector generalization of uniform scalar quantization [6]. It provides near optimum performance for uniformly distributed sources and due to its highly structured nature, low complexity algorithms have been found. However, for fixed rate coding with rate R of sources with infinite support such as i.i.d. Gaussian, special attention has to be paid to the overload problem and there exist apparently no low complexity methods to date [7]. The application of a lattice quantization scheme for quantizing filter coefficients of linear predictive coding is presented in [8] and [9]. As we are interested in a very high dynamic range within the scope of this paper, an approach by lattice quantization is not the first choice.

Among various different classes of quantizers, the so-called product quantizer belongs to the quantizers with structured

codebooks and uses a reproduction codebook that is the Cartesian product of lower dimensional reproduction codebooks [7]. This ordinarily results in less arithmetic and storage complexity and in addition to that, the product structure makes searching easier. A shape-gain vector quantizer (see [10], [11]) is an example of a product quantizer: the product reproduction codebook consists of a gain codebook and a shape codebook. The gain codebook consists of positive scalars whereas the shape codebook comprises unit norm D dimensional vectors. The overall reproduction vector is defined by the product of both codebook entries, which yields scaled vectors of the shape codebook.

Polar quantizers belong to the group of shape-gain quantizers where a set of D samples is quantized in terms of magnitude and phase. Strictly polar quantizers (SPQ) assign a fixed number of phase positions at each magnitude level, whereas with unrestricted polar quantizers (UPQ), complete freedom is allowed in assigning the available number of phase positions at different magnitude levels. In [12] SPQs for complex (i.e. $D = 2$) Gaussian random variables are numerically optimized, but this approach suffers from the lack of flexibility needed for factoring the number of overall quantization levels into levels for the magnitude and the phase. Since these have to be integer numbers, a solution close to the optimum is not possible for all rates. In [13] and [14] UPQs for pairs of Gaussian random variables with unit variance are proposed and numerically optimized. The encoder uses lookup-tables for the magnitude and the associated phase levels which are numerically optimized by a gradient correction procedure. An analytical solution for the optimum number of levels by an asymptotic analysis of optimum uniform polar quantization ($D = 2$) is given by [15]. Another approach using radial compander technique is introduced in [16]. Reference [17] proposes multivariate block polar quantization (MBPQ) where the amplitudes and phases of a set of L complex variables are quantized by $2L$ scalar quantizers in a manner that the overall phase bit budget in a block is distributed according to the amplitudes observed in that particular block. The quantizer is signal-adaptive in the sense that a training signal can be used for optimizing the quantizer but this information has to be forwarded to the decoder, i.e. there is no signal-adaption without transmission of side information. In [18] a polar quantizer for vectors of Fourier coefficients is iteratively designed which benefits in terms of bit allocation from the fact that the human ear appears to be more sensitive to magnitude than to phase. Hence, this method belongs to the non-waveform conserving methods. Generally, coding methods which make use of spectral transforms or equivalent block based operations introduce a high signal delay which is incompatible with our demand for an extremely low signal delay of about 10 samples.

The main focus of this paper, the so-called spherical logarithmic quantization (SLQ), belongs to the group of shape-gain quantizers, to be more precisely SLQ resembles a multidimensional polar quantizer which incorporates ADPCM. SLQ can be described as operating with two codebooks, although none of these two codebooks has to be stored directly. Moreover, encoder and decoder are able to cope with very small lookup-tables and a few calculations. Each source signal vector is represented in D-dimensional spherical coordinates, the first codebook is used for a logarithmic quantization of the magnitude and the second codebook consists of vectors uniformly covering the surface of a D-dimensional unit sphere.

Especially in the recent past, spherical quantization and related topics were successfully worked on. In [19], SLQ was introduced for the first time as a new method for efficiently digitizing analog signals while preserving the original waveform as close as possible with respect to the relative quantization error and an extremely low delay of the signal. Reference [20] discusses the construction of a spherical code in a similar way and also deals with the problem of uniform distributing quantization cells on a unit sphere's surface. A similar algorithm for partitioning the surface of a multidimensional sphere into equal areas can be found in [21] and [22] as well as several analysis and proofs on equal area partitioning from a mathematical point of view. The construction of a spherical code similar to peeling an apple in three dimensions and by simulated annealing is presented in [23]. Unfortunately, besides computational effort and possible convergence problems of the algorithm, no efficient scheme for indexing, coding and decoding is given. Wrapped and laminated spherical codes were introduced in [24] and [25]. These approaches construct a spherical code by wrapping and projecting, respectively, an underlying lattice onto a unit sphere. The resulting spherical code may achieve asymptotically a maximum density, but only Gaussian sources with unit variance are considered [26]. In order to cope with non-Gaussian sources, the author of [26] proposes transform coding of the source. On the one hand, this introduces a high signal delay and on the other hand, correlations within the source signal cannot be used for a better performance of the quantizer. This is one of the reasons why SLQ does not allow spectral transforms or equivalent block based operations. Nevertheless, SLQ achieves both an extremely low delay and a remarkable performance gain when applied to correlated sources. The index assignment proposed in [26] assumes that there is an efficient method for assigning indices to the underlying lattice, while the indexing scheme presented in our paper is especially designed to incorporate ADPCM.

C. Differential Pulse Code Modulation

Differential pulse code modulation (DPCM) encodes pulse code modulation (PCM) samples $x[k]$ as differences between the current sample $q[k]$ of the source signal and a predicted sample, cf. Fig. 1 (discrete time: $k \in \mathbb{Z}$). The predicted sample is calculated by filtering the preceding *quantized*[1] source signal samples with a linear prediction filter $H_{\mathrm{P}}(z)$ of order P that is able to exploit statistical dependencies (i.e. redundancy) within the stream of source samples.

The power ratio (in dB) of the source signal and the prediction error signal is called prediction gain:

$$G_{\mathrm{P}} \;=\; 10\log_{10}\left(\frac{\sigma_q^2}{\sigma_x^2}\right) \;. \tag{8}$$

[1]We call this a *backward* prediction. In a forward prediction, the initial (i.e. unquantized) source signal samples are fed into the prediction filter.

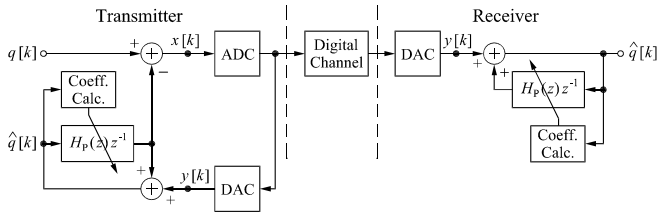

Fig. 1. DPCM system using backward adaptive backward prediction.

Usually, the predictor is optimized in the sense of the minimum mean-squared prediction error ([3], [28], [29]), i.e. the maximum prediction gain. In order to track the statistics of a nonstationary source signal, the filter coefficients are recalculated after a certain time period which leads to the well known adaptive prediction (ADPCM). In this paper, a method similar to the adaptive prediction scheme of CELP audio coding is applied to improve performance, see [30]. Optimum linear prediction (i.e. for $P \to \infty$) completely decorrelates the source signal. Due to the central limit theorem, the pdf often approximates a Gaussian pdf which is likely to result in a reduction of the peak-to-average ratio and therefore in a more convenient distribution for the subsequent quantization. The purpose of DPCM is to produce a difference signal $x[k]$ with both a smaller variance and a smaller maximum value in order to reduce the number of quantization intervals without an increase of the interval width, hence without loss of quality. Due to the transmitter's feedback-structure ("backward prediction") the effective quantization noise at the receiver side is the same as the real quantization noise at the output of the DAC, i.e. white noise, assuming that the number of quantization levels is sufficiently large. Moreover, no amplification of the quantization noise takes place resulting in the full benefit from the prediction gain G_{P}.

D. Spheres in D Dimensions

A point in the real-valued D-dimensional space $\mathbb{R}^D$ is simply a string of D real-valued coordinates $\boldsymbol{x} := (x_1, ..., x_D)$. A D-dimensional sphere in $\mathbb{R}^D$ with its center at the origin and radius r consists of all the points $\boldsymbol{x} := (x_1, ..., x_D)$ satisfying

$$x_1^2 + x_2^2 + \cdots + x_D^2 \;=\; r^2 \;. \tag{9}$$

The volume $V_{\mathrm{sph},D}(r)$ of a D-dimensional sphere [31] is

$$V_{\mathrm{sph},D}(r) = \frac{\pi^{D/2}}{\Gamma(D/2+1)} \cdot r^D \;, \tag{10}$$

where $\Gamma(\cdot)$ denotes the usual Gamma-Function. The surface area $S_{\mathrm{sph},D}(r)$ can be acquired by the derivative of the volume $V_{\mathrm{sph},D}(r)$ in radial direction [31]

$$S_{\mathrm{sph},D}(r) \;=\; D \cdot \frac{\pi^{D/2}}{\Gamma(D/2+1)} \cdot r^{D-1} \;. \tag{11}$$

A cut through this sphere, without loss of generality orthogonal to the x_D-axis, can be interpreted as a $D-1$-dimensional sphere with radius $\sqrt{r^2 - h^2}$ within the $D-1$-dimensional space. This is immediately obvious, if the x_D coordinate in (9) is assigned to a fixed value, for example $x_D := h$ where h should be within $[-r, +r]$, otherwise the plane will not cut the sphere. Hence, we can conclude that the cutting area of a D-dimensional sphere resembles a $D-1$-dimensional sphere, e.g. cutting a 3-dimensional sphere results in a circle.

Spherical Coordinates: A vector $\boldsymbol{x} = (x_1, ..., x_D)$ of D samples in Cartesian coordinates can be expressed by spherical coordinates $\boldsymbol{u} := (\varphi_1, ..., \varphi_{D-1}, r)$. Notice that throughout this paper, the radius will be addressed by the last component of the vector $\boldsymbol{u}$ in spherical coordinates. In the D dimensional case, the spherical representation consists of a radius r and $D-1$ angles, i.e. one longitude φ_1, considered modulo 2π, and $D-2$ colatitudes $\varphi_i,\ i \in \{2, ..., D-1\}$. In this paper, we will use the following equations for the transformation into the spherical domain, Fig. 2 provides an example for $D = 3$

$$\varphi_1 \;=\; \arg\left(x_D + \mathrm{j}\; x_{D-1}\right) \qquad \in [-\pi, +\pi) \tag{12}$$

$$\varphi_i \;=\; \arg\left(\sqrt{\sum_{l=D-i+1}^{D} x_l^2} + \mathrm{j}\; x_{D-i}\right) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right), \quad i \in \{2, ..., D-1\} \tag{13}$$

$$r^2 \;=\; \sum_{l=1}^{D} x_l^2 \;. \tag{14}$$

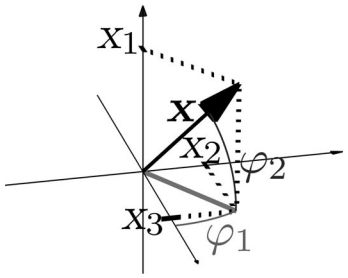

Fig. 2. Spherical coordinates (Example: $D = 3$).

Note that the samples of the vector $\boldsymbol{x}$ are used in reverse order compared to usual literature, the necessity of this will become clear soon.

The reconstruction of the Cartesian components from the vector $\boldsymbol{u}$ yields:

$$x_i \;=\; r \cdot b_{D-i} \cdot \sin\left(\varphi_{D-i}\right), \quad i \in \{1, ..., D-1\} \tag{15}$$

$$x_D \;=\; r \cdot b_1 \cdot \cos\left(\varphi_1\right) = r \cdot b_0 \tag{16}$$

as well as the radii b_i of the circles of colatitude of the unit sphere ($r = 1$):

$$b_{D-1} \;=\; 1 \tag{17}$$

$$b_{D-i} \;=\; \prod_{l=D-i+1}^{D-1} \cos\left(\varphi_l\right), \quad i \in \{2, ..., D\} \;. \tag{18}$$

III. Spherical Logarithmic Quantization

Applying SLQ, a vector $\boldsymbol{x}$ of D samples in Cartesian coordinates at first is expressed by spherical coordinates $\boldsymbol{u}$, according to (12)–(14). In order to preserve the properties of logarithmic quantization, i.e. the independence of the SNR of the signal variance and its special (short time) pdf, usual logarithmic quantization according to the $\mathcal{A}$-law is applied for

the radius (magnitude). For the angle variables φ_i we simply apply individual uniform quantization. However, the feature of logarithmic quantization (i.e. the proportionality of the width of a quantization interval to the signal value) is already present for all D dimensions, because of the proportionality of a circular arc segment to the radius, see Fig. 3 for a simple example of three dimensions ($D = 3$). Thus, the loss for the SNR due to companding (cf. (7)) has to be paid only for logarithmic quantization for one out of D dimensions. This is the essential source of the gains due to SLQ.

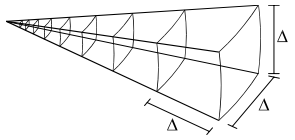

Fig. 3. Proportionality of the size of the quantization cells to the radius.

A. Arrangement of the Quantization Cells

In order to achieve a uniform distribution of the quantization cells on the surface of the sphere as shown in Fig. 4, the quantization intervals of the angle φ_i have to be functions of the quantized angle variables $\widehat{\varphi}_l$, $l \in \{i+1, i+2, ..., D-1\}$ of higher orders. The individual uniform quantization of the

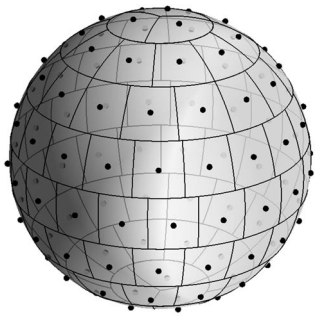

Fig. 4. Example for $D = 3$, $R = 4\,\mathrm{bit/sample}$. Quantization cells covering the surface of the 3-dimensional unit sphere and the corresponding reconstruction vectors ($\bullet$).

angles corresponds to a quantization of the surface of a D-dimensional sphere roughly into $(D-1)$-dimensional (hyper-) cubes (e.g. ordinary squares for $D = 3$) as long as the number of quantization cells is very large.

Since a D dimensional vector quantization combines D samples for a joint quantization, a number of $M^D := 2^{R \cdot D}$ levels can be spent on the quantization of the entire vector. In the following, we will discuss how to split this M^D quantization levels into quantization intervals for the radius r and the surface of a D-dimensional unit (hyper-)sphere in an optimum way.

B. Radius Quantization

Let r_0 denote a scale factor representing the maximum radius to be quantized. (We will fix r_0 soon). In the logarithmic area $\frac{r_0}{\mathcal{A}} \leq r \leq r_0$ of $\mathcal{A}$-Law companding the following compression function is applied for the radius as usual, cp. (4):

$$k_{\mathrm{K}}(r) = r_0 \cdot \left(1 + c \cdot \ln \frac{r}{r_0}\right) \quad \text{with} \quad c := \frac{1}{1 + \ln \mathcal{A}} \,. \tag{19}$$

If we assume M_D quantization intervals for the radius (D^{th} component of the vector $\boldsymbol{u}$), the width $\Delta r(r)$ of a quantization cell in radial direction reads

$$\Delta r(r) \approx \frac{r_0}{M_D \cdot k'_{\mathrm{K}}(r)} = \frac{1}{M_D \cdot c} \cdot r \;, \tag{20}$$

with the derivative $k'_{\mathrm{K}}(r)$ of the compression function. It is obvious that $\Delta r(r)$ is proportional to the radius r, i.e. proportional to the signal magnitude and therefore satisfies the claim due to logarithmic quantization. Notice that $\Delta r(r)$ is independent of r_0 in the considered logarithmic area.

In order to achieve a similar performance of the quantization in spherical coordinates as in Cartesian coordinates (quantization range $x_i \in [-1; 1]$ in every dimension) with respect to overloading the quantizer, we normalize the maximum value r_0 of the radius such that the volume of the corresponding D-dimensional sphere equals the volume of a D-dimensional cube with edge length 2:

$$V_{\mathrm{sph},D}(r_0) \;\stackrel{!}{=}\; V_{\mathrm{cube},D} \;=\; 2^D \;. \tag{21}$$

Inserting (10) leads to

$$r_0 = \frac{2}{\left(V_{\mathrm{sph},D}(1)\right)^{1/D}} \quad \text{with} \quad r_0 \geq 1 \quad \forall D \in \mathbb{N} \;. \tag{22}$$

C. Surface Quantization

Since the radius (i.e. the gain) is encoded separately, we can simplify the calculation of the SNR by choosing $r = 1$ in the following, i.e. we consider the quantization cells covering the surface of a unit sphere. Hence, a spherical representation and quantization, respectively, of the shape is the favorable approach compared to e.g. a quantization in the Cartesian space. Moreover, choosing $r = 1$ is possible without loss of generality because the SNR does not depend on the radius within $\frac{r_0}{\mathcal{A}} \leq r \leq r_0$ due to its logarithmic quantization and the natural scaling of arc segments with the radius. Each quantization cell is represented by a reconstruction vector pointing to the cell's center, Fig. 4 shows an example for $D = 3$ and $R = 4\,\mathrm{bit/sample}$. Please note that the arrangement of the quantization cells located at the surface of the unit sphere is independent of the radius. The width of the quantization cells reads:

$$\Delta := \Delta r(r = 1) \;=\; \frac{1}{M_D \cdot c} \;. \tag{23}$$

For a fair partitioning of the number of M^D quantization intervals into intervals for the radius and intervals for the surface of the unit sphere, following considerations hold: Regarding the approximation of cubic quantization cells, i.e. enforcing an approximately equal width of quantization cells in all dimensions (radius and circular arc segments), the surface of the unit sphere is uniformly covered with $\lfloor M_\varphi \rfloor$ equal $(D-1)$-dimensional hypercubes[2]. Each hypercube contributes an area (a "$(D-1)$-dimensional volume") of Δ^{D-1}

[2] $\lfloor x \rfloor \in \mathbb{Z}$: largest integer $\leq x$ with $x \in \mathbb{R}$.

to the whole surface area $S_{\mathrm{sph},D}(1)$, cf. (11), thus we can calculate the number of quantization cells that are available for the quantization of the surface:

$$S_{\mathrm{sph},D}(1) \stackrel{!}{=} M_\varphi \cdot \Delta^{D-1} \ . \tag{24}$$

As M^D intervals per quantization step are available, additionally

$$M_\varphi \cdot M_D \stackrel{!}{=} M^D \tag{25}$$

has to be claimed. With (23) and (25) the optimum width Δ of the quantization cell can easily be calculated:

$$\Delta = \frac{1}{M} \cdot \left(\frac{S_{\mathrm{sph},D}(1)}{c} \right)^{\frac{1}{D}} \ . \tag{26}$$

The constant c is determined by the selected dynamic range, see (6) and (19). Using (23) and (25), this result immediately leads to the maximum number $\lfloor M_D \rfloor$ of quantization cells that have to be devoted to the radius

$$M_D = M \cdot \left(\frac{c^{1-D}}{S_{\mathrm{sph},D}(1)} \right)^{\frac{1}{D}} \tag{27}$$

and the maximum number $\lfloor M_\varphi \rfloor$ of quantization cells or reconstruction vectors, respectively, on the surface of the unit sphere

$$M_\varphi = M^{D-1} \cdot \left(c^{D-1} \cdot S_{\mathrm{sph},D}(1) \right)^{\frac{1}{D}} \ . \tag{28}$$

Equations (27) and (28) represent the optimal bit allocation between gain and shape for the multidimensional logarithmic approach. Note that M_D and M_φ usually are non-integer numbers and have to be rounded down to provide an integer number of quantization cells for an implementation which does not matter as long as RD is high enough. This refers to the factorization problem of usual two-dimensional polar quantization (cf. [12]) which vanishes with increasing dimensionality.

In the following, we will compare the allocation of bits between gain and shape of SLQ with the method proposed in [27] for wrapped spherical codes. The latter represents a fixed-rate shape-gain quantizer for the memoryless Gaussian source while using a shape quantizer which is constructed from wrapped spherical codes. In the general case, the rate allocation is numerically optimized and for special high rate assumptions an analytical solution is provided. This solution corresponds roughly to the experimental observations and devotes approximately M codevectors to the gain and M^{D-1} codevectors to the shape. Fig. 5 shows the normalized number of quantization cells devoted to the radius for both SLQ and the method discussed in [27] for wrapped spherical codes. It is visible that the number of intervals that SLQ devotes to the radius is up to a factor of 40 larger than with a scalar quantization with an equivalent resolution of very small signal values. Thus, we expect a gain of 32 dB (which corresponds to more than 5 bit/sample) for SLQ in this area! Please note that these two coding schemes refer to different optimization problems: With the wrapped spherical codes, the bit allocation is optimized for a given distribution of the source, i.e. Gaussian. In contrast to this, SLQ solves the rate allocation problem in an optimum way w.r.t. the logarithmic aspect.

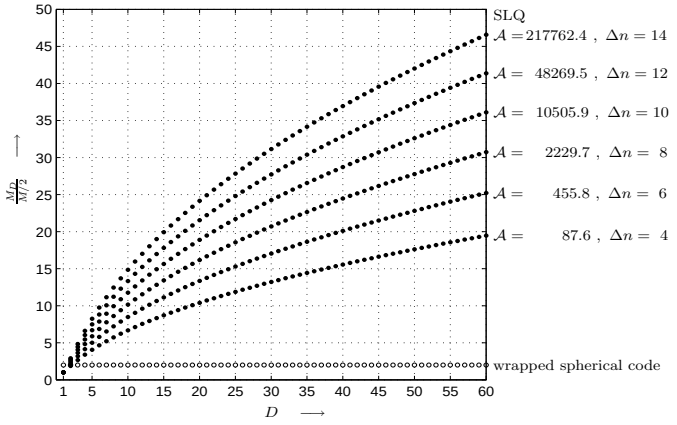

Fig. 5. Number of quantization cells devoted to the radius normalized to the number $M/2$ of cells per absolute value of a sample in scalar quantization. • SLQ, ◦ wrapped spherical code.

D. Quantization Noise and SNR

As long as there is a sufficiently high number of quantization cells in D dimensions, the usual approximation of a uniformly distributed quantization error within these cubic cells, each of them represented by its center point, may be applied. Thus, the noise variance for a D-dimensional signal vector with radius r is:

$$N_{\mathrm{Q}}(r) \ = \ \frac{\Delta^2(r)}{12} \cdot D \ . \tag{29}$$

SLQ enforces $\Delta(r) = \Delta \cdot r$ in all dimensions within the logarithmic area of the compressor characteristic ($\frac{r_0}{\mathcal{A}} \le r \le r_0$), hence, the average quantization noise power can be written as

$$N_{\mathrm{Q}} = \mathcal{E}\{N_{\mathrm{Q}}(r)\} = \mathcal{E}\left\{ \frac{\Delta^2 \cdot r^2}{12} \cdot D \right\} = D \cdot \frac{\Delta^2}{12} \cdot \sigma_q^2 \ . \tag{30}$$

The resulting Signal-to-Noise Ratio can now be calculated

$$\mathrm{SNR} \ = \ \frac{\sigma_q^2}{N_{\mathrm{Q}}} \ = \ \frac{12}{\Delta^2 \cdot D} \ = \ L(D,\mathcal{A}) \cdot M^2 \tag{31}$$

with

$$L(D,\mathcal{A}) \ := \ \frac{12}{M^2 \cdot \Delta^2 \cdot D} \ = \ \frac{12}{\pi} \cdot \frac{\left[c \cdot \Gamma\left(\frac{D}{2}+1\right) \right]^{\frac{2}{D}}}{D^{\frac{D+2}{D}}} \ , \tag{32}$$

see (26) and (11). As designed, the SNR is independent of the variance of the signal in this area.

Looking at (31), the term $1/L(D,\mathcal{A})$ may be considered as the loss with regard to the rate-distortion-bound for i.i.d. Gaussian random variables ("6 dB-per bit-rule"). $L(D,\mathcal{A})$ is presented in Fig. 6 for different values of $\mathcal{A}$ and D.

The loss increases, if $\mathcal{A}$ and the dynamic range, respectively, is increased. For example, let us consider a scalar logarithmic quantization ($D = 1$) and choose $\mathcal{A} = 87.6$ (which corresponds to the set of parameters of the ITU G.711 recommendation [4] for voice transmission at 8 bits per sample and a sampling frequency of 8 kHz), this results in a loss of 10.0 dB as mentioned earlier. While increasing dimensionality D this loss is vanishing up to a gap of $1.53\,\mathrm{dB}$. The higher $\mathcal{A}$, i.e. the dynamic range for constant SNR, is, the more pronounced this effect becomes.

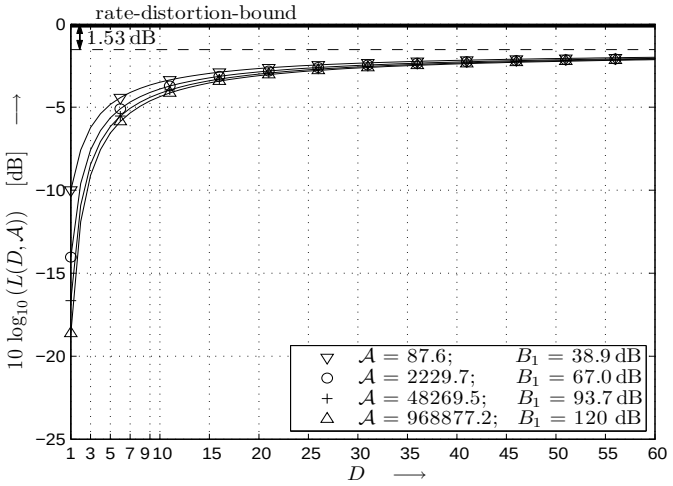

Fig. 6. SNR-loss with respect to $R \cdot 6.02\,\mathrm{dB}$ versus number of dimensions for different values of $\mathcal{A}$.

Using Stirling's approximation for factorial or Γ-function we may write

$$L(D,\mathcal{A}) \approx \frac{6}{\pi \mathrm{e}} \cdot \frac{c^{\frac{2}{D}} \cdot \pi^{\frac{1}{D}}}{D^{\frac{1}{D}}} \quad , \quad D \gg 1 \; . \tag{33}$$

The limit for an infinite number of dimensions

$$\lim_{D\to\infty} L(D,\mathcal{A}) = \frac{6}{\pi \mathrm{e}} \quad \widehat{=} \quad -1.53\,\mathrm{dB} \tag{34}$$

does not depend on the variable c, i.e. the parameter $\mathcal{A}$. This means that $\mathcal{A}$ and the selected dynamic range, respectively, has no influence on the gap between the achieved SNR and the rate-distortion-bound for a large number of dimensions D.

In other words: By using SLQ, it is possible to compensate the loss due to companding described in (7) up to a residual margin of $1.53\,\mathrm{dB}$. Therefore, it is possible in principle to choose the parameters $\mathcal{A}$ and D in a way that an infinite dynamic range is achieved without having to accept a noticeable loss in the maximum attainable SNR. Neither any knowledge about the source signal nor the transmission of side information for amplitude scaling etc. is necessary as long as overload is avoided, which is no problem because of the extremely high dynamic range of SLQ.

The loss of $1.53\,\mathrm{dB}$ is caused by the suboptimal cubical quantization cells instead of spheres and corresponds to a rate loss of 0.25 bit/sample which we accept for two main reasons: First, the advantage of an extreme reduction in complexity and second, the possibility of incorporating ADPCM with backward recursion into the quantization algorithm. Since the latter is one of the main achievements of this work, we will present more details on this issue in the following section. Moreover, the rate-distortion-function for Gaussian i.i.d. random variables is a lower bound on the minimum achievable rate (distortion, respectively) at a given distortion (rate, respectively), cf. Berger's upper bound of the rate-distortion-function, [33]. Therefore, further improvements beyond these 1.53 dB are not possible at all under the given constraints and requirements. Because of lack of knowledge of the pdf of the source signal, a (worst case) Gaussian pdf would have to be taken into account and coding better than the rate-distortion-function of the Gaussian variable using special probability density functions can be excluded in principle.

The distance of 1.53 dB towards the rate-distortion-bound is identical to the so called "ultimate shaping gain". This is the upper limit of the shaping gain, i.e. the gain in reducing average energy compared to a signal with uniform distribution. The ultimate shaping gain is achieved for a continuous Gaussian probability density function, see [34] for further details.

Furthermore the dynamic range changes by increasing dimensionality at a constant $\mathcal{A}$ due to two effects. First, the scaling factor r_0 increases which causes a slight shift of the dynamic range towards higher signal levels. Second, the limitations of the logarithmic compression only apply to one single dimension (the radius) whereas for the other $D-1$ dimensions the proportionality of the width of the quantization cells to the signal value is maintained even for smallest signal values. Thus, the left edge of the dynamic range (cf. Fig. 7) expands in D dimensions approximately to

$$B_D \approx B_1 + 10\log_{10}(D) - 20\log_{10}(r_0) \;\; [\mathrm{dB}] \quad (D \geq 2) \; . \tag{35}$$

Since r_0 only causes a shift, the width of the dynamic range merely depends on B_1 (and $\mathcal{A}$, respectively), D and the overload characteristic which is determined by the actual pdf of the signal x that is being quantized. Table I gives the width of the dynamic range (with no respect to overload) for several combinations of $\mathcal{A}$ and D. Moreover, because of the averaging effect of combining D samples, the robustness to overload increases which results in a further expansion of the dynamic range. In the limit $D \to \infty$, an infinite dynamic range is offered for any value of $\mathcal{A}$. As visible from Table I, the ITU G.711 recommendation [4] ($\mathcal{A} = 87.6$, $D = 1$, $R = 8$ bits per sample) implies a left edge of the dynamic range of $-B_1 = -38.8\,\mathrm{dB}$. If e.g. Gaussian signals are assumed, the right edge yields $-E_1 = -10.2\,\mathrm{dB}$ which offers only a very small dynamic range.

TABLE I
Term "$B_1 + 10\log_{10}(D)$" (in dB), characterizing the width of the dynamic range (with no respect to overload)

Δn	$\mathcal{A}$ \ D	1	2	3	5	10	20	60
2	14.8	23.4	26.4	28.2	30.4	33.4	36.4	41.2
4	87.6	38.8	41.9	43.6	45.8	48.8	51.9	56.6
6	455.8	53.2	56.2	57.9	60.2	63.2	66.2	71.0
8	2229.7	67.0	70.0	71.7	74.0	77.0	80.0	84.7
10	10505.9	80.4	83.4	85.2	87.4	90.4	93.4	98.2
12	48269.5	93.7	96.7	98.4	100.7	103.7	106.7	111.5
14	217762.4	106.8	109.8	111.5	113.7	116.8	119.8	124.5
16	968877.2	119.7	122.7	124.5	126.7	129.7	132.7	137.5

Note that the structural delay of spherical logarithmic quantization is, as for any vector quantization method, exactly D samples. As to be seen from Fig. 6 and Fig. 12, the major portion of the possible gain is already achieved at very small values of D (up to 10).

E. SLQ and Linear Prediction

For the benefit of a further increase in SNR by exploiting correlations within the source signal, we use ADPCM as

shown in Fig. 1 without introducing any signal delay. This provides the very important possibility to adapt to an unknown source signal. Notice that a SNR gain, expressed by a decrease of the mean squared error between original and reconstructed samples, is only possible for application of backward prediction of section II-C. Using logarithmic quantization, the SNR for the prediction error signal $x[k]$ is independent of its variance or equivalently the power of the quantization noise is proportional to the signal power. This is the reason why the prediction gain is directly transformed into an SNR gain. Thus, logarithmic quantization is a favorite choice for DPCM [35], [36]. Fig. 7 gives an illustration: Notice the plateau of constant

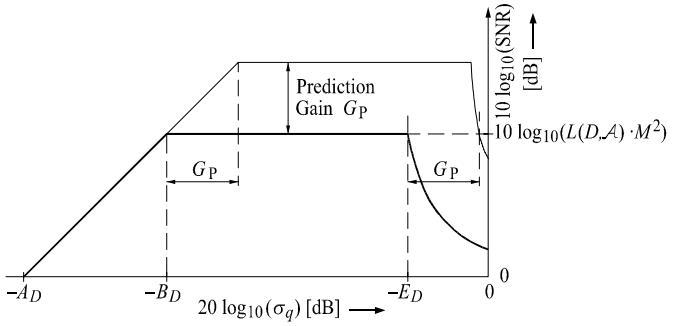

Fig. 7. SNR of a D dimensional logarithmic quantizer with (upper curve) and without (lower curve) prediction gain G_{P}.

SNR with and without the use of prediction. The SNR level that is achieved by multidimensional SLQ, cf. (31), is simply increased by the prediction gain G_{P}, but, at the same time, the dynamic range is reduced by G_{P} at its left edge. Signals at a signal level smaller than $-B_D + G_{\mathrm{P}}$ are not able to achieve the maximum constant SNR of

$$10\log_{10}(\mathrm{SNR}) \;=\; 10\log_{10}\left(L(D,\mathcal{A})\cdot M^2\right) + G_{\mathrm{P}}\ . \quad (36)$$

For this reason, the width of the dynamic range should be selected as large a possible taking into consideration the prediction gain as well. If we now recall the example of the ITU G.711 recommendation [4] and its enhancement by means of ADPCM (see [36]), it is immediately obvious that due to the small dynamic range, the prediction gain can hardly increase the SNR. With SLQ, the necessary increase in $\mathcal{A}$ and the corresponding loss in SNR can easily be retrieved by increasing the number of dimensions D, cf. e.g. Fig. 6.

On the other hand, if the variance of the signal is decreased by means of prediction at the encoder side, overload does not take place until the decreased signal level (prediction error signal) becomes too high. Hence, if prediction is involved, the signal level of the original source signal is allowed to be higher (by G_{P}) than without prediction. Therefore, the dynamic range is expanded by G_{P} at its right edge. Thus, the dynamic range is not lowered due to prediction but simply shifted towards higher signal levels of the original signal so that the prediction error signal fits to the original dynamic range. Moreover, if the mean value of the source signal is not zero (DC-offset), this is automatically handled by the differential encoding and the performance is not decreased. This supports the universal applicability of SLQ.

Additionally, as already mentioned above and what is of very high importance in our context, no further signal delay is introduced by ADPCM w.r.t. PCM, because an optimum prediction error filter for maximum prediction gain is causal and strictly minimum phase in principle [37] and therefore invertible without any structural delay.

F. Vector Quantization versus ADPCM with backward recursion

Generally, the process of quantization can be optimized if a joint quantization of a block of many samples is performed, i.e. if vector quantization is applied. Increasing the number of dimensions provides a higher gain and a higher SNR, respectively.

The process of linear prediction is optimized, if the DPCM encoder is able to completely decorrelate the source signal. Since usually a major part of this correlation is contained within directly successive samples, the optimum predictor has to operate sample by sample, i.e. to predict samples instead of vectors of several samples. Moreover this prediction has to be performed by backward recursion since this is the only way of increasing the measurable SNR at the decoder output by the prediction gain. Backward recursion implies that the *quantized* samples are used for the calculation of the prediction error. Hence, usually a sample by sample (i.e. a scalar) quantization scheme is favorable for maximum prediction gain. (The problem of combination of vector quantization and backward prediction may be solved by means of a multidimensional optimization process, see section IV-A).

The use of forward prediction instead of backward prediction decorrelates the signal as well (and can easily be combined with any vector quantization scheme by means of a "preprocessing"), but the the inverse prediction error filter at the decoder side will cause an amplification of the quantization noise. This is the reason why the prediction gain does not increase the measurable SNR at the decoder output.

Well known relevant approaches in literature can roughly be divided into these two groups namely vector quantization and scalar ADPCM with backward recursion. To our best knowledge, these two methods have not been combined so far. SLQ seems to be the first approach of combining vector quantization on the one hand and ADPCM with sample by sample backward recursion on the other hand. Hence, both quantization and prediction are optimized and the increase in quality at the decoder output is measurable by SNR and not only as a subjective impression.

IV. Practical Implementation of Spherical Logarithmic Quantization combined with ADPCM

A. Encoding

At this point, we have to note that the encoder of Fig. 1 involves a basic contradiction: On the one hand, for the calculation of the actual prediction error sample $x[k]$ all previous reconstructed samples $\widehat{q}[k-i]\ ,\ i = 1, 2, \ldots$ have to be available, i.e. there is a mutual dependence of the samples due to the feedback loop. On the other hand, SLQ is a vector quantization scheme which means that a joint quantization

of blocks of length D of *unquantized* samples is performed. At first glance, it seems that the two claims "DPCM with backward prediction" and "vector quantization" cannot be simultaneously fulfilled.

For solutions to this problem, a method very closely related to the principle "analysis-by-synthesis" well known from CELP waveform coding methods [38], [39] might be used in combination with a simple discrete step gradient descent algorithm minimizing the squared Euclidean distance between the unquantized vector of samples

$$\boldsymbol{q}[l] = (q[l \cdot D], q[l \cdot D + 1], ..., q[l \cdot D + D - 1]) \qquad (37)$$

and a corresponding reconstruction vector $\widehat{\boldsymbol{q}}[l]$, $l = \lfloor k/D \rfloor$ ("discrete vector time"). First, an entire (forward predicted) signal vector is quantized and afterwards a better reconstruction vector is iteratively searched for, see [19] for details.

In this paper, a much more efficient encoding approach will be introduced. As proposed in [40], SLQ is combined with DPCM by successively processing the D dimensions with respect to quantization and prediction filtering. For better readability, the encoding and decoding algorithm is outlined in Fig. 8 and Fig. 9, respectively. In the following, we will give a more detailed description.

```
Estimate signal magnitude r by forward prediction
Parallelize for different r̃ {
  Quantize r̃ according to the A-Law → r̂
  Select radius index N_D
  for (i = 1, ..., D − 2) {
    Calculate number of layers M_{D−i} on a D-dim. sphere with radius b̃_{D−i}
    Calculate the numbers K_{D−i}^{(ν_{D−i})} of cells on each layer
    Calculate φ_{D−i} by conversion of x_i into spherical coordinates
    Quantize φ_{D−i} → φ̂_{D−i}
    Select layer index ν_{D−i}
    Reconstruct y_i
    Feed y_i into the backward prediction loop → x_{i+1}
    Cut D − i + 1-dim sphere → D − i-dim sphere with radius b̃_{D−i−1}
    }
  i = D − 1
  Quantize φ_1 → φ̂_1   (final layer)
  Reconstruct y_{D−1}, x_D, y_D
  }
Select r̃ which leads to minimum distortion
Calculate the surface index N_φ
Calculate the overall index N
```

Fig. 8. SLQ-Encoding.

Radius Estimation: First, the algorithm needs some knowledge about the signal magnitude r of the vector $\boldsymbol{x}$ i.e. the radius r according to (14). Unfortunately, at the beginning, only $\boldsymbol{q}$ is given and the prediction gain is not initially known due to the backward predictive structure. But as the prediction gains for forward and backward prediction do not differ very much, a good estimate of G_{P} and therefore of r may be calculated by the use of forward prediction, i.e. by disabling the chain ADC – DAC at the transmitter side (Fig. 1). In many cases, this estimate is not sufficiently exact and has to be improved afterwards, we will refer to this later on in this section. The quantized radius $\widehat{r}$ is provided by means of scalar logarithmic quantization according to the $\mathcal{A}$-Law of the radius r.

```
Decompose N → N_D, N_φ
  for (i = 1, ..., D − 1) {
    Calculate number of layers M_{D−i} on a D-dim. sphere with radius b̃_{D−i}
    Calculate the numbers K_{D−i}^{(ν_{D−i})} of cells on each layer
    Calculate the remaining surface index Ñ_{D−i}
    Reconstruct layer index ν_{D−i}
    Calculate the corresponding reconstruction angle φ̂_{D−i}
    Reconstruct the Cartesian coordinate y_i
    }
Reconstruct the Cartesian coordinate y_D
Feed y_1, ..., y_D into the inverse prediction error filter → q_1, ..., q_D
```

Fig. 9. SLQ-Decoding.

Step-by-step Quantization: The vector $\boldsymbol{q} := (q_1, ..., q_i, ..., q_D)$ is now quantized according to Fig. 1 sample by sample in their chronological correct order starting with $i = 1$. x_i is now used to calculate the corresponding unquantized angle φ_{D-i} by the use of (15), whereas b_{D-i} are given by (17)–(18):

$$\varphi_{D-i} = \arcsin\left(\frac{x_i}{r \cdot b_{D-i}}\right) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right) \ , \ i \in \{1, ..., D-2\}. \quad (38)$$

Afterwards, φ_{D-i} is fed into the step-by-step quantizer which works on each dimension (angle) individually: Assume the number M_{D-i} of quantization intervals to be used for quantizing this angle is given (we will discuss this point in detail later for better readability), the unit sphere is divided into M_{D-i} layers as it is shown in Fig. 10. This can also be visualized by dividing a semicircle arc (i.e. half of the "meridian") with radius 1 (arclength = π) into M_{D-i} equivalent angular intervals. Each possible reconstruction angle $\widehat{\varphi}_{D-i}$ of φ_{D-i} corresponds to one layer and is labeled by a layer index ν_{D-i}.

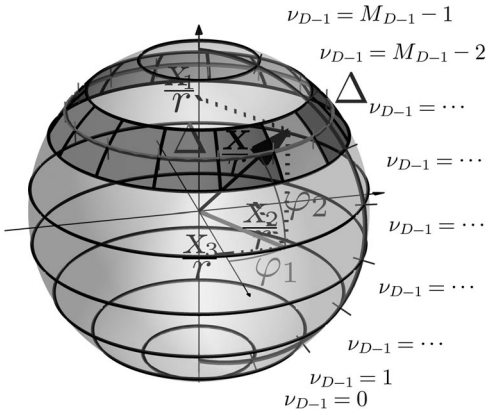

Fig. 10. Example: $D = 3$, $R = 4$ bit/sample, $\mathcal{A} = 48269.5$, i.e. $\lfloor M_\varphi \rfloor = 117$: Quantization of φ_2 (elevation) and φ_1 (azimuth).
1^{st} stage: $M_2 = 9$ layers ($\nu_2 = 0, ..., 8$), selected layer: $\nu_2 = 6$.
2^{nd}stage: $M_1 = K_2^{(6)} = 15$ cells within the 6^{th} layer, selected cell: $\nu_1 = 12$.

Now, the layer that comprises the unquantized angle φ_{D-i} of (38) is selected and addressed by the layer index ν_{D-i}. The corresponding reconstruction angle $\widehat{\varphi}_{D-i}$ reads

$$\widehat{\varphi}_{D-i} = (\nu_{D-i} + 0.5) \cdot \frac{\pi}{M_{D-i}} - \frac{\pi}{2} \ , \qquad (39)$$

and points to the middle of the layer which is selected.

We now use the *quantized* radius $\widehat{b}_{D-i}$ of the selected layer's "circle of colatitude" and the *quantized* radius $\widehat{r}$ to

reconstruct the i^{th} Cartesian component of $\boldsymbol{y}$

$$y_i = \widehat{r} \cdot \widehat{b}_{D-i} \cdot \sin\left(\widehat{\varphi}_{D-i}\right), \quad i \in \{1, ..., D-2\} \tag{40}$$

which is further processed according to Fig. 1.

Increasing i by 1 leads to the next sample of $\boldsymbol{x}$, i.e. x_i, thus a successive quantization of the angles is accomplished.

Within this succession, several quantized radii $\widehat{b}_{D-i}$ of the circles of colatitude are necessary, they are obtained similarly to their unquantized equivalents b_{D-i}, cf. (17) and (18).

The Final Layer Problem: The succession given above only holds for $i \leq D-2$, the final layer ($i = D-1$) has to be processed in a special way because the last remaining angle φ_1 is determined by x_{D-1} and x_D simultaneously which contradicts the successive procedure shown above. As a solution to this problem, i.e. in order to quantize x_{D-1} independently of x_D, we have to rotate the quantization cells within the (x_{D-1}, x_D)-plane by $\frac{\pi}{2}$ counterclockwise such that a symmetric quantization of φ_1 with respect to the x_{D-1}-axis is achieved. This provides the remaining Cartesian components y_{D-1} and y_D as well as the index ν_1 describing the selected quantization cell on the last layer, for details see [32]. Another solution to this problem takes into account all possible reconstruction angles of $\widehat{\varphi}_1$ and selects the one which leads to the best SNR, i.e. minimizes

$$\left(\widehat{q}_{D-1} - q_{D-1}\right)^2 + \left(\widehat{q}_D - q_D\right)^2 \ . \tag{41}$$

Since this method proved to perform best, it was used for the simulations which are presented in this paper. Using this method, the CPU time of the encoder is increased by 30% compared with the first method.

Discussion of the Radius: The exact signal magnitude r (*unquantized* radius) of the signal vector $\boldsymbol{x}$ is essential, because all D *unquantized* components of the vector $\boldsymbol{x}$ are intimately connected with r by the transformation formulas (12)–(18). Due to their mutual dependence on each other by the backward prediction loop and the involvement of the *unquantized* radius r in (38) the encoder ought to know (or to estimate, respectively) r at the beginning of each succession. If r does not fit accurately to all $x_i,\ i \in \{1, ..., D\}$, (38) and (18) will provide a wrong result for φ_{D-i} and b_{D-i} at the next encoding stage. The errors accumulate very fast and finally lead to a completely worthless result. Interestingly, the decoder does not need to have this knowledge, since signal *reconstruction* is always performed by the use of the *quantized* radius $\widehat{r}$, at the encoder as well as at the decoder side, cf. (40). The decoder never needs to evaluate (38) where the *unquantized* radius is necessary. Since the initial approximation of r by means of forward prediction proved not to be sufficiently accurate, the algorithm above has to be parallelized η times for different discrete *unquantized* values $\widetilde{r}$ of the radius. Several simulations showed that the search within the restricted interval $\widetilde{r} \in [r - \Delta r(r), r + \Delta r(r)]$ is a sufficient choice where $\Delta(r)$ denotes the width of the quantization cell, cf. (20) and (26). The aim of this search is to find a value of $\widetilde{r}$ for which the quantization noise is minimized. Simulations (see [32]) have shown, that a reasonable number of parallelizations (between $\eta = 10$ and $\eta = 50$), reduces the remaining loss in SNR compared with the maximum according to (36) down to some 0.1 dB. Certainly, this parallelization seems like a drastic increase in computational effort, but on the one hand, this search does not necessarily involve an iterative procedure and therefore can easily be parallelized in a hardware implementation. On the other hand, the encoder can evaluate some different values for the radius until a certain stopping criterion (e.g. computation time or SNR) is fulfilled.

Of course, if SLQ is applied without linear prediction (i.e. $\boldsymbol{x} \equiv \boldsymbol{q}$, e.g. with Gaussian sources) or even only with forward prediction instead of backward prediction, there is no need for an estimation of r because the power of the vector $\boldsymbol{x}$ and therefore r can be calculated exactly. In this case, no parallelization is necessary (i.e. $\eta = 1$) and therefore, the computational complexity is not increased.

This new step-by-step encoding procedure is a fundamental improvement over the mentioned gradient descent algorithm, since the search for the optimum quantization cell does not need to be performed within the D-dimensional spherical domain and the computational complexity does not significantly increase with the number of dimensions D. Thus, vector quantization at a high dimensionality can here be combined with ADPCM with backward recursion, a feature that has not been known so far in the field of vector quantization. The search for the optimum quantization cell is to a certain extent transferred from the D-dimensional space towards the one-dimensional space and only a reasonable number of η "cells" have to be investigated.

B. Indexing

Radius Index Selection: The radius is quantized by scalar logarithmic quantization according to the $\mathcal{A}$-Law, we denote the selected quantization level of the considered radius by a radius index $N_D \in \{0, 1, ..., \lfloor M_D \rfloor - 1\}$, cf. (27).

Index Distribution among the Layers: As the assignment N_D is straightforward and independent of an index $N_\varphi \in \{0, 1, ..., \lfloor M_\varphi \rfloor - 1\}$ of the selected cell on the unit sphere, the problem can be mainly concentrated towards the calculation of N_φ.

During the process of encoding, the surface of the D-dimensional unit sphere has to be successively split up into

$$M_{D-i} \quad = \quad \left\lfloor \frac{\pi \cdot \widetilde{b}_{D-i}}{\Delta} \right\rfloor \tag{42}$$

layers with the reconstructed radii $\widetilde{b}_{D-i}$ of the circles of colatitude. Fig. 10 gives an example for $D = 3$. Each layer comprises a possible reconstruction angle cf. (39) which we will denote with $\widehat{\varphi}_{D-i}^{(\nu_{D-i})}$ and holds a number $K_{D-i}^{(\nu_{D-i})}$ of quantization cells (i.e. $D-i$-dimensional hypercubes). The superscript index (ν_{D-i}) in both variables marks which layer number the variable belongs to. For optimum surface quantization, an uniform distribution of the quantization cells among the surface of the sphere is required. Therefore, $K_{D-i}^{(\nu_{D-i})}$ has to be proportional to its corresponding layer's (layer index ν_{D-i}) quantized surface which can be calculated by subtraction of two partial sphere surfaces.

Let $S_{D-i+1}(\varphi_{D-i})$ denote the partial surface of a $D-i+1$-dimensional unit sphere between the sphere's bottom (i.e. $\varphi_{D-i}=-\frac{\pi}{2}$) and the "elevation" angle φ_{D-i}.

We can now write

$$S_{D-i+1}(\varphi_{D-i}) = \int_{-\frac{\pi}{2}}^{\varphi_{D-i}} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cdots \int_{0}^{2\pi} \rho \, \mathrm{d}\varphi'_1 \cdots \mathrm{d}\varphi'_{D-i-1} \mathrm{d}\varphi'_{D-i} \; , \tag{43}$$

with the element of surface

$$\rho = \cos^{D-i-1}(\varphi_{D-i}) \cdot \cos^{D-i-2}(\varphi_{D-i-1}) \cdot \; \cdots \; \cdot \cos(\varphi_2). \tag{44}$$

Now, let the radii of the considered $D-i+1$-dimensional spheres be $\widetilde{b}_{D-i}$. The partial surface between the bottom of such a non-unit sphere and the "elevation" angle φ_{D-i} yields

$$\left(\widetilde{b}_{D-i}\right)^{D-i} \cdot S_{D-i+1}(\varphi_{D-i}) \; . \tag{45}$$

At the 1^{st} stage of the encoding process ($i=1$), we have to use the reconstructed radius $\widetilde{b}_{D-1}=1$.

In order to achieve a unique index representation despite rounding we have to quantize the two surfaces to be subtracted towards multiples of the quantization cell volume Δ^{D-i}. We can now calculate the number $K_{D-i}^{(\nu_{D-i})}$ of quantization cells that is available for being spent on each layer

$$K_{D-i}^{(\nu_{D-i})} = \left\lfloor \left(\frac{\widetilde{b}_{D-i}}{\Delta}\right)^{D-i} \cdot S_{D-i+1}\left(\widehat{\varphi}_{D-i}^{(\nu_{D-i})} + \frac{\pi}{2M_{D-i}}\right) \right\rfloor - \left\lfloor \left(\frac{\widetilde{b}_{D-i}}{\Delta}\right)^{D-i} \cdot S_{D-i+1}\left(\widehat{\varphi}_{D-i}^{(\nu_{D-i})} - \frac{\pi}{2M_{D-i}}\right) \right\rfloor , \quad i \le D-2 \; . \tag{46}$$

The selection of a layer or the fixing of the reconstruction angle $\widehat{\varphi}_{D-i}$, respectively, by the use of (39) in the i^{th} stage of the SLQ algorithm resembles a cut through a $D-i+1$-dimensional sphere. The intersecting area of this cut yields a $D-i$-dimensional sphere (cf. section II-D) on the surface of which we can now continue distributing quantization cells. The interesting aspect of this consideration is that the algorithm can now be recursively executed: We start at encoding stage $i=1$ with a $D-i+1$-dimensional sphere, and end up at stage $i=D-1$ with a 2-dimensional sphere, i.e. a circle. This circle represents the final layer and holds the quantization cells on its circumference. The "last" reconstruction angle $\widehat{\varphi}_1$ finally points to one of them.

To maintain recursivity, the radius $\widetilde{b}_{D-i-1}$ of the next encoding stage is now interpreted as the radius of a $D-i$-dimensional sphere holding the same surface as the (quantized) surface of the selected layer of the $D-i+1$-dimensional sphere divided by Δ and thus can be obtained by:

$$\left(\widetilde{b}_{D-i-1}\right)^{D-i-1} \cdot S_{\mathrm{sph},D-i}(1) \stackrel{!}{=} \frac{K_{D-i}^{(\nu_{D-i})} \cdot \Delta^{D-i}}{\Delta} \; . \tag{47}$$

This division by Δ equals the calculation of an "average" circumference of the accessed circle of colatitude. Now we move to the next stage i.e. we increase i by 1 and, as long as $i \le D-2$ is still valid, we recursively continue in dividing the actual layer into M_{D-i} sublayers according to (42). At $i=D-1$ we have reached the last layer which consists of

$$M_1 = \left\lfloor \frac{2\pi \cdot \widetilde{b}_1}{\Delta} \right\rfloor = K_2^{(\nu_2)} \tag{48}$$

quantization cells.

During the process of encoding, an analytic integration in (43) is not necessary for every signal vector. Since the number of possible quantized angles per unit sphere is restricted, a storage of precalculated values within some lookup-tables is possible and does not need huge amounts of memory. Reference [32] provides further details as well as analytic results of (43) for several numbers of dimensions D.

Surface Index Selection: The index N_φ can now easily be calculated by summing up the numbers of quantization cells spent on the $D-1$ layers or subspheres, respectively, between their "bottoms" and the particular selected layer ν_{D-i} on each subsphere:

$$N_\varphi = \left(\sum_{i=1}^{D-2} \left(\sum_{m=0}^{\nu_{D-i}-1} K_{D-i}^{(m)} \right) \right) + \nu_1 \; . \tag{49}$$

Fig. 11 shows the distribution of the quantization cells within the $M_2=9$ layers according to the example ($D=3$, $R=4$ bit/sample, $\mathcal{A}=48269.5$) and illustrates the position of the quantization cells on the sphere's surface. In this example, the maximum number of quantization cells on the surface according to (28) yields $\lfloor M_\varphi \rfloor = 117$ of which 114 cells may be used due to rounding effects caused by (46).

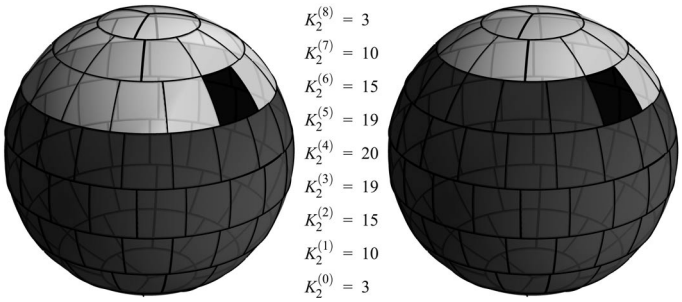

Fig. 11. Example: $D=3$, $R=4$ bit/sample, $\mathcal{A}=48269.5$, $\lfloor M_\varphi \rfloor = 117$, $\nu_2=6$, $\nu_1=12$: Summing up the number of quantization cells in each layer.
Only along the elevation (left): $\sum_{m=0}^{\nu_2-1} K_2^{(m)} = 86$
Along elevation and azimuth (right): $N_\varphi = \sum_{m=0}^{\nu_2-1} K_2^{(m)} + \nu_1 = 86+12 = 98$
■: Surface cells (indices, respectively) to be accumulated, cf. (49)
■: Surface cell to be transmitted (cell index N_φ).

The surface index N_φ is now combined with the index N_D of the (scalar) quantization of the radius towards the overall index

$$N = N_\varphi \cdot \lfloor M_D \rfloor + N_D \tag{50}$$

that is transmitted.

C. De-Indexing and Decoding

At the receiver side, the decomposition of the transmitted index N into the surface index N_φ and the radius index N_D

can easily be achieved by a modulo operation

$$N_D = N \bmod \lfloor M_D \rfloor \quad (51)$$

$$N_\varphi = \frac{N - N_D}{\lfloor M_D \rfloor} , \quad (52)$$

with the modulo operator $\bmod u$, $u \in \mathbb{N}$, mapping an integer into the interval $[0, ..., u-1]$.

The recovery of ν_{D-i}, $i \in \{1, ..., D-1\}$ requires some calculations similar to those used while encoding: The decoding algorithm successively processes the stages for $i = 1, ..., D-1$ and provides at each stage a "remaining surface index" $\widetilde{N}_{D-i}$, starting with $\widetilde{N}_{D-1} = N_\varphi$. At the 1st stage of the decoding process (i.e. $i = 1$), the $K_{D-i}^{(\nu_{D-i})}$ are calculated for all possible reconstruction angles $\widehat{\varphi}_{D-i}^{(\nu_{D-i})}$ of the current layer using (46), i.e. for $\nu_{D-i} = 0, ..., M_{D-i} - 1$. The reconstruction index ν_{D-i} is now fixed by fulfilling

$$\sum_{n=1}^{i} \sum_{m=0}^{\nu_{D-n}-1} K_{D-n}^{(m)} \;\le\; \widetilde{N}_{D-i} \;<\; \sum_{n=1}^{i} \sum_{m=0}^{\nu_{D-n}} K_{D-n}^{(m)} \quad (53)$$

and the corresponding reconstruction angle $\widehat{\varphi}_{D-i}$ can be calculated by applying (42) and (39). The remaining surface index for the next stage of decoding yields

$$\widetilde{N}_{D-i-1} = \widetilde{N}_{D-i} - \sum_{n=1}^{i} \left(\sum_{m=0}^{\nu_{D-n}-1} K_{D-n}^{(m)} \right) . \quad (54)$$

Now i is increased by 1 and as long as $i \le D-2$ is valid, we continue with calculating ν_{D-i}, cf. (53). Reaching $i = D-1$, leads to

$$\nu_1 = \widetilde{N}_1 \quad (55)$$

$$\text{and} \qquad \widehat{\varphi}_1 = (\nu_{D-i} + 0.5) \cdot \frac{2\pi}{M_1} + \frac{\pi}{2} . \quad (56)$$

The reconstruction of the Cartesian components of the signal is performed by applying (15) and (16) to the reconstructed angles.

V. Simulation Results

In this section we will present the results of two simulation setups: The first simulation is intended to verify the theoretical results of section III concerning the performance of SLQ according to dynamic range and the rate-distortion-bound. Further simulations provide typical examples of low delay audio coding by means of SLQ as well as a comparison of SLQ with the ITU G.722 codec [42].

Fig. 12 shows the distance between the SNR and the rate-distortion-bound $R \cdot 6.02\,\mathrm{dB}$ depending on the average signal level $20\log_{10}(\sigma_q)$ of i.i.d. Gaussian random variables for the example choosing $\mathcal{A} = 48269.5$ for different numbers of dimension D. Prediction makes no sense in this example and is therefore disabled. Hence, the radius r can be calculated exactly and no parallelization is necessary, i.e. $\eta = 1$. These simulation results meet exactly the theoretical analysis according to (31)–(34) and Fig. 6, respectively. Further simulation results for some test sets, e.g. signals with a priori known correlations, are to be found in [32].

The following simulation gives an impression of the performance of SLQ using audio assessment material from the well known SQAM compact disc [41]. Since SLQ is intended to use neither psychoacoustic effects nor inaudibilities due to stereophonic effects, the following results refer to single channel audio signals (mono) only. Therefore the stereo tracks of the SQAM compact disc had been converted into two mono signals which only the left channel was used from for the simulations. This original signal was encoded by means of SLQ into a binary data stream and decoded afterwards. Signal quality is now evaluated by means of the *segmental* SNR denoted with $\mathrm{SNR}_{\mathrm{seg}}$ (short time SNRs which are averaged over l_{seg} samples at a time) and the *overall* $\overline{\mathrm{SNR}}$ between the entire source signal and the reconstructed signal. $\sigma^2_{q,\mathrm{seg}}$ and $\sigma^2_{x,\mathrm{seg}}$, respectively, denote the segmental variances (cf. Fig. 1), i.e. the signal variances averaged over l_{seg} samples each, and $G_{\mathrm{P,seg}}$ represents the segmental prediction gain (in dB). $\mathrm{SNR}_{y,\mathrm{seg}}$ resembles the segmental SNR of the signal $y[k]$ (with reference to $x[k]$) and thus provides the SNR "directly" between the input and output of the quantizer (the chain ADC – DAC, respectively, cf. Fig. 1). Fig. 13 shows simulation results for the track "`63_Soloists_(Verdi)`" (Soloists, Giuseppe Verdi, left stereo channel) at $R = 4\,\mathrm{bit/sample}$, $D = 12$, $\mathcal{A} = 48269.5$. The figure reveals intense variations in segmental signal level of the source signal (graph A) and at the input of the quantizer (graph C). Since the segmental quantizer input level never drops below the left edge of the dynamic range $-B_{12} = -98.7\,\mathrm{dB}$, $\mathrm{SNR}_{\mathrm{y,seg}}$ in graph D shows a constant level of $20.0\,\mathrm{dB}$ according to (31) due to the logarithmic quantization[3]. As expected from theory in (36), graph E shows that the resulting segmental SNR is the segmental SNR of graph D increased by the segmental prediction gain (graph B). Please note, that the overall algorithmic delay of this example is only D (here: 12) signal samples, i.e. only $272\,\mu\mathrm{s}$ at a sampling frequency of $44.1\,\mathrm{kHz}$ which corresponds to an acoustic propagation delay for a distance of $9\,\mathrm{cm}$. It is obvious that SLQ performs best, if a high prediction gain is achieved. With real world signals this is usually the case, if the sampling frequency is high enough, e.g. $44.1\,\mathrm{kHz}$ for high quality audio

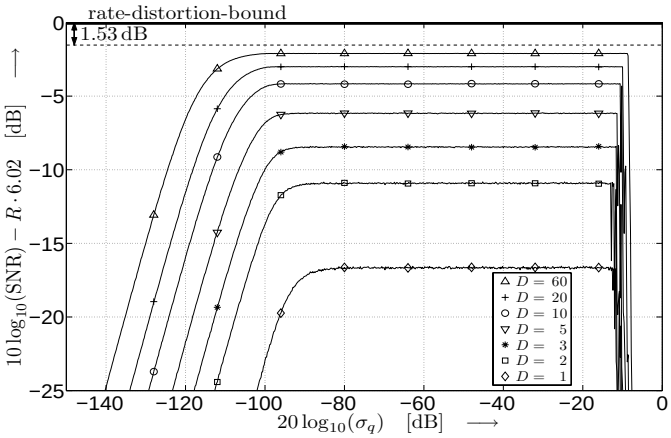

Fig. 12. SNR of i.i.d. Gaussian distributed signal values using spherical logarithmic quantization ($\mathcal{A} = 48269.5$, B_1 = 93.7 dB, $R = 12\,\mathrm{bit/sample}$).

[3]Since (32) does not take into account the prediction gain, σ_q^2 has to be identified with σ_x^2.

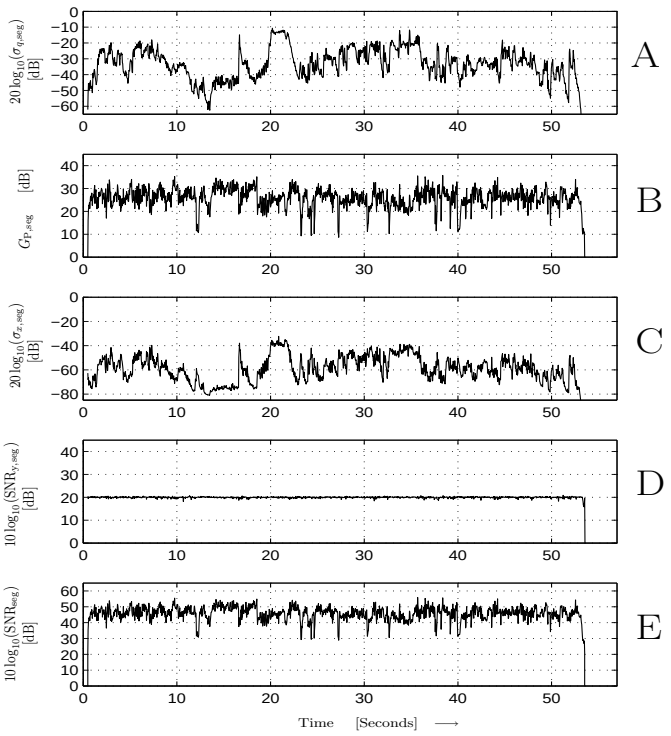

Fig. 13. Segmental signal level (A), segmental prediction gain (B), segmental quantizer input level (C), segmental SNR due to quantization (D) and segmental SNR (E) over time. Quantization at $R = 4$ bit/sample, $D = 12$, $\mathcal{A} = 48269.5$, $\Delta n = 12$ bit/sample, adaptive prediction, $l_{\mathrm{seg}} = 1536$, $\eta = 41$, average $G_{\mathrm{P}} = 24.6\,\mathrm{dB}$, average $10\log_{10}\left(\overline{\mathrm{SNR}}\right) = 44.6\,\mathrm{dB}$.

signals. The application of SLQ at very low sampling frequencies e.g. $8\,\mathrm{kHz}$ as it is often used with speech processing is not recommended. Unfortunately, many well known quantizers are optimized for speech processing at a sampling frequency of $8\,\mathrm{kHz}$ (e.g. the ITU G.726 recommendation [36]) which is not an advantageous field of application for SLQ.

For the sake of completeness, we used the ITU G.722 recommendation [42] for a comparison with SLQ. ITU G.722 describes a sub-band ADPCM system with an adaptive logarithmic quantizer and a maximum bit rate of 64 kbit/s at a sampling frequency of $16\,\mathrm{kHz}$ and is therefore suited for a variety of higher quality speech and music applications. Moreover, prediction with backward recursion is applied which allows to evaluate the achieved quality by measuring the SNR at the decoder output. To provide a fair comparison, we converted the audio assessment material from the SQAM compact disc [41] from $44.1\,\mathrm{kHz}$ to a sampling frequency of $16\,\mathrm{kHz}$ and used it for SLQ as well as for ITU G.722 in each case at a bit rate of 64 kbit/s or 4 bit/sample, respectively. Table II shows the measured SNR at the decoder output for a representative selection of audio files. Roughly speaking, SLQ outperforms the ITU G.722 codec by approximately $3.5\,\mathrm{dB}$ averaged over the whole SQAM CD. In this example, the overall algorithmic delay of SLQ is $0.75\,\mathrm{ms}$ compared with $1.5\,\mathrm{ms}$ of the ITU G.722 codec.

Despite this remarkable performance of SLQ, this example is only of moderate interest for us due to the following reason. The average signal level $20\log_{10}(\sigma_q) = -24.0\,\mathrm{dB}$ of the audio material on the SQAM CD is pretty high and therefore lacks relevance for practical applications such as e.g. digital waveform coding for microphones. Since SLQ is designed to achieve a very high dynamic range, we did a second comparison of these two coding schemes using a by $30\,\mathrm{dB}$ attenuated version of the same audio files, Table III presents the corresponding results. While the SNR of SLQ remains the same (except for small deviations due to a slightly different prediction gain), the SNR of the ITU G.722 breaks down. On the average, the achieved SNR of SLQ is approximately $12\,\mathrm{dB}$ higher than the SNR of ITU G.722. Since we used again $\mathcal{A} = 48269.5$ in this example, the left edge of the dynamic range of SLQ is $-B_{12} = -98.7\,\mathrm{dB}$ which means, that the signal level could be reduced even further without noticing a decrease in the achieved SNR. Concerning the ITU G.722 recommendation, which is designed to operate on 16 bit uniformly quantized source signals, this would lead to a unfair comparison: if the source signal level becomes too low, the quantization noise of this uniform quantization would also have to be taken into account.

TABLE II
Audio material from SQAM CD, resampled to 16 kHz, average signal level $20\log_{10}(\sigma_q) = -24.0\,\mathrm{dB}$. Quantization at $R = 4$ bit/sample. SLQ: $D = 12$, $\mathcal{A} = 48269.5$, $\Delta n = 12$ bit/sample

Track	Section	Signal	$10\log_{10}(\overline{\mathrm{SNR}})$ [dB],SLQ	$10\log_{10}(\overline{\mathrm{SNR}})$ [dB],G.722
11	1	Double-bass, arpeggio	52.6	51.0
14	1	Oboe, arpeggio	37.1	34.7
18	2	Bassoon, melodious phrase	47.2	47.5
24	1	Tuba, arpeggio	53.7	57.9
26	2	Claves, single tone, rhythm	33.3	27.0
27	1	Castanets, single tone, rhythm	27.0	13.9
31	2	Cymbal - hard stick, single tone	21.5	13.1
32	2	Triangle, roll	27.6	16.5
35	1	Glockenspiel, arpeggio	32.3	18.1
41	3	Celesta, melodious phrase	46.7	47.6
44	1	Soprano	42.7	39.9
48	1	Quartet	38.6	35.3
53	1	Female speech, German	29.0	26.0
54	1	Male speech, German	28.2	25.4
60	1	Piano, Schubert	39.1	37.4
69	1	Abba	28.8	26.0

TABLE III
Audio material from SQAM CD, resampled to 16 kHz, attenuated by 30 dB, average signal level $20\log_{10}(\sigma_q) = -54.0\,\mathrm{dB}$. Quantization at $R = 4$ bit/sample. SLQ: $D = 12$, $\mathcal{A} = 48269.5$, $\Delta n = 12$ bit/sample

Track	Section	Signal	$10\log_{10}(\overline{\mathrm{SNR}})$ [dB],SLQ	$10\log_{10}(\overline{\mathrm{SNR}})$ [dB],G.722
11	1	Double-bass, arpeggio	52.2	33.1
14	1	Oboe, arpeggio	37.0	29.6
18	2	Bassoon, melodious phrase	47.1	32.7
24	1	Tuba, arpeggio	54.0	33.9
26	2	Claves, single tone, rhythm	32.2	23.2
27	1	Castanets, single tone, rhythm	26.8	13.8
31	2	Cymbal - hard stick, single tone	21.3	12.1
32	2	Triangle, roll	27.6	13.2
35	1	Glockenspiel, arpeggio	32.5	16.6
41	3	Celesta, melodious phrase	46.6	33.5
44	1	Soprano	42.1	28.5
48	1	Quartet	38.3	28.8
53	1	Female speech, German	28.8	23.2
54	1	Male speech, German	28.2	23.5
60	1	Piano, Schubert	38.5	22.7
69	1	Abba	28.7	21.7

We like to note, that it is not the purpose of this paper to present a prediction scheme which achieves the theoretical maximum prediction gain. In fact, this work may be considered as proof of concept for the possibility of combining prediction schemes like DPCM with the vector quantization method of SLQ but there is no doubt that further gains are possible by applying sophisticated state-of-the-art prediction schemes.

A fair comparison of SLQ with other quantization schemes such as shape-gain lattice quantization is difficult for several reasons. Usual vector quantization schemes with prediction only apply forward instead of backward prediction and therefore do not provide a measurable quality in the sense of SNR for correlated sources. So far, quantizers that apply backward prediction are scalar quantizers and they are usually not designed for a large dynamic range (e.g. 60 dB).

If prediction is not considered and e.g. a Gaussian source is used, the performance benchmark for SLQ is rate-distortion bound, as well as for all other relevant approaches in literature. As discussed in section III-D and verified by simulation (see Fig. 12) SLQ is able to reach this bound up to a gap of $1.53\,\mathrm{dB}$. While some vector quantization methods outperform SLQ in this case, it is the unique strength of SLQ that if the source signal is correlated the achieved prediction gain directly increases the objective (measurable) SNR at the decoder output.

Reference [32] provides further simulation results and shows that erroneous data transmission using SLQ does not cause more noise at the dequantizer output as all usual PCM- and DPCM-schemes. Moreover an optional feature of including a perceptional weighting filter into the SLQ encoder (similar to the CELP approach, cf. [38], [43]) is presented in [32]. The reader is invited to visit the author's website [44] where many audio examples are available for download and evaluation.

The simulation programs for SLQ are implemented in MATLAB [45] without the use of additional C routines (MEX-files) and were run on an Quad-Core AMD Opteron(tm) Processor, Model 2354 with 1100 MHz using only one CPU core. The CPU time spent on a typical run is about 15 times slower than realtime for encoding, depending on the number of parallelizations η. The decoder is capable of running in realtime. Since the source code is not speed optimized and includes hard disk access as well as debugging output, this can only be considered as a rough estimation of computational complexity. To our experience, an implementation being capable for realtime processing is possible in a straightforward way using an optimized code and a modern DSP.

VI. Conclusion

In this paper, a waveform-conserving digitizing scheme for analog source signals is presented which combines gains from multidimensional and logarithmic quantization at a tolerable complexity for implementation. ADPCM with backward recursion is merged with vector quantization at a high dimensionality and further gains in SNR are achieved by the direct transformation of the prediction gain into a SNR gain due to the logarithmic quantization of the prediction error signal. Although the purpose of SLQ is similar to well known scalar methods like e.g. the ITU G.726 recommendation [36] which applies logarithmic quantization and ADPCM, SLQ offers more benefits. An extremely high dynamic range is provided, but the great loss due to companding that has to be taken into account for scalar quantization is here drastically reduced, i.e. the logarithmic characteristic of a quantization scheme is here achieved almost for free. This paper shows how SLQ is able to compensate this loss up to a residual margin of $1.53\,\mathrm{dB}$ by exploiting packing properties in D dimensions. SLQ is therefore able to asymptotically reach the rate-distortion-bound for Gaussian i.i.d. random variables within this margin. Besides an advantageous tradeoff between rate and distortion this method is particularly characterized by a very low algorithmic signal delay of only D signal samples. In addition to that, SLQ provides a reconstructed waveform which approximates the original very closely, and therefore the achieved quality can be measured by means of objective scales like the SNR. Since no exploitation of irrelevance takes place, a reconstructed SLQ signal can be post-processed by audio engineering tools without the risk that coding artifacts become audible (or visible). Moreover, the application of SLQ is not restricted to audio signals, in fact any kind of signal can be used. Although the involvement of ADPCM is one of the main achievements of SLQ, it is an optional feature and can be disabled, e.g. for Gaussian source signals.

Future work should be done to check whether similar algorithms can be applied to other vector quantization schemes such as lattice quantization. The aim is to find other successful combinations of vector quantization on the one hand and ADPCM with backward recursion on the other hand. In this context, special attention has to be paid to the indexing scheme since a complex full search should be avoided.

Acknowledgment

The authors want to express their deep thanks to the anonymous reviewers for many valuable comments.

References

[1] A. Gersho. Quantization. *IEEE Commun. Society*, vol. 15, no. 5, pp. 16-29, Sept. 1977.
[2] W. R. Bennett, Spectrum of Quantized Signals. *Bell Syst. Technical J.*, vol. 27, pp. 446-472, July 1948.
[3] N. S. Jayant, P. Noll. Digital Coding of Waveforms. Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
[4] —. ITU-T G.711 Pulse Code Modulation (PCM) of Voice Frequencies. Int. Telecommun. Union, Recommendation G.711, Nov. 1988.
[5] A. Gersho. Asymptotically optimal block quantization. *IEEE Trans. Inf. Theory*, vol. 25, pp. 373-380, July 1979.
[6] A. Gersho, R. M. Gray. Vector Quantization and Signal Compression. Kluwer Academic Publishers, 1992. ISBN 978-0-7923-9181-4.
[7] R. M. Gray, David L. Neuhoff. Quantization. *IEEE Trans. Inf. Theory*, vol. 44, pp. 2325-2383, Oct. 1998.
[8] A. Vasilache, M. Vasilache, I. Tabus. Predictive multiple-scale lattice VQ for LSF quantization. *Int. Conference on Acoustics, Speech, and Signal Process. (ICASSP)*, vol. 2, pp. 657-660, Phoenix, Arizona, USA, 1999.
[9] A. Vasilache, B. Dumitrescu, I. Tabus. Multiple-scale leader-lattice VQ with application to LSF quantization. *Signal Process.*, vol. 82, pp. 563-586, April 2002.
[10] M. J. Sabin and R. M. Gray. Product Code Vector Quantizers for Speech Waveform Coding. *Conf. Rec. GLOBECOM*, pp. 1087-1091, Dec. 1982.

[11] M. J. Sabin and R. M. Gray. Product Code Vector Quantizers for Waveform and Voice Coding. *IEEE Trans. on Acoustics, Speech, and Signal Process.*, vol. ASSP-32, pp. 474-488, June 1984.
[12] W. A. Pearlman. Polar Quantization of a Complex Gaussian Random Variable. *IEEE Trans. on Commun.*, vol. 27, pp. 892-899, Jun. 1979.
[13] S. G. Wilson. Magnitude/Phase Quantization of Independent Gaussian Variates. *IEEE Trans. on Commun.*, vol. 28, pp. 1924-1929, Jan. 1980.
[14] D. L. Neuhoff. Polar quantization revisited. *IEEE Int. Symposium on Inform. Theory*, p. 60, July 1997.
[15] Z. H. Peric, M. C. Stefanovic. Asymptotical Analysis of Optimal Uniform Polar Quantization. *AEU – Int. Journal of Electron. and Commun.*, vol. 56, pp. 345-347, 2002.
[16] Z. H. Peric, O. D. Milanovic, A. Z. Jovanovic. Optimal companding vector quantization for circularly symmetric sources. *Information Sciences*, vol. 178, pp. 4375-4381, Nov. 2008.
[17] H. Pobloth, R. Vafin, W. B. Kleijn. Multivariate block polar quantization. *IEEE Trans. on Commun.*, vol. 53, pp. 2043 - 2053, Dec. 2005.
[18] P-Chi Chang, R. Gray, J. May. Fourier Transform Vector Quantization for Speech Coding. *IEEE Trans. on Commun.*, vol. 35, pp. 1059-1068, Oct. 1987.
[19] J. B. Huber, B. Matschkal. Spherical Logarithmic Quantization and its Application for DPCM. *5th Int. ITG Conference on Source and Channel Coding (SCC)*, pp. 349-356, Erlangen, Germany, Jan. 2004.
[20] Z. Utkovski, A. Utkovski, T. Eriksson.High-Dimensional Spherical Quantization of Gaussian Sources. *9th Canadian Workshop on Inform. Theory (CWIT)*, pp. 211-214, Montreal, Quebec, June 2005.
[21] P. Leopardi. A Partition of the Unit Sphere into Regions of Equal Area and Small Diameter. *Electron. Trans. on Numerical Anal.*, vol. 25, pp. 309-327, 2006.
[22] P. Leopardi. Distributing Points on the Sphere: Partitions, Separation, Quadrature and Energy. Ph.D. dissertation. Dept. of Applied Math., School of Math. and Statistics, University of New South Wales, 2007.
[23] A. E. Gamal, L. Hemachandra, I. Shperling, V. Wei. Using Simulated Annealing to Design Good Codes. *IEEE Trans. Inf. Theory*, vol. 33, pp. 116-123, Jan. 1987.
[24] J. Hamkins, K. Zeger. Asymptotically Dense Spherical Codes - Part I: Wrapped Spherical Codes. *IEEE Trans. Inf. Theory*, vol. 43, pp. 1774-1785, Nov. 1997.
[25] J. Hamkins, K. Zeger. Asymptotically Dense Spherical Codes - Part II: Laminated Spherical Codes. *IEEE Trans. Inf. Theory*, vol. 43, pp. 1786-1798, Nov. 1997.
[26] J. Hamkins. Design and Analysis of Spherical Codes. Ph.D. dissertation. Univ. of Illinois, Urbana-Champaign, Sept. 1996.
[27] J. Hamkins, K. Zeger. Gaussian Source Coding With Spherical Codes. *IEEE Trans. Inf. Theory*, vol. 48, pp. 2980-2989, Nov. 2002.
[28] J. Durbin. The Fitting of Time-series Models. *Review of the Int. Statistical Instit.*, vol. 28, no. 3, pp. 233-243, 1960.
[29] N. Levinson. The Wiener RMS Error Criterion in Filter Design and Prediction. *J. of Mathematical Physics*, vol. 25, pp. 261-278, 1947.
[30] J.-H. Chen. High-quality 16 kb/s Speech Coding with a One-Way Delay less than 2ms. *Int. Conference on Acoustics, Speech, and Signal Process. (ICASSP)*, Albuquerque, NM, USA, 1990.
[31] J. H. Conway, N. J. A. Sloane. Sphere Packings, Lattices, and Groups. Springer-Verlag, 3rd edition, Jan. 1999.
[32] B. Matschkal. Spherical Logarithmic Quantization. Ph.D. dissertation. University of Erlangen–Nuremberg, Shaker Verlag, Aachen, Germany, March 2008. ISBN 978-3-8322-6989-0.
[33] T. Berger. Lossy Source Coding. *IEEE Trans. Inf. Theory*, pp. 2693-2723, Oct. 1998.
[34] G. D. Forney, L.-F. Wei. Multidimensional Constellations - Part I. Introduction, Figures of Merit, and Generalized Cross Constellations. *IEEE J. on Select. Areas in Commun.*, JSAC-7, pp. 877-892, Aug. 1989.
[35] R. W. Stroh, M. Paez. A Comparison of Optimum and Logarithmic Quantization for Speech PCM and DPCM Systems. *IEEE Trans. on Commun.*, vol. 21, pp. 752-757, June 1973.
[36] —. ITU-T G.726 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM). Int. Telecommun. Union, Recommendation G.726, Dec. 1990.
[37] L. Pakula and S. Kay. Simple Proofs of the Minimum Phase Property of the Prediction Error Filter. *IEEE Trans. on Acoustics, Speech, and Signal Process.*, vol. 31, 1983.
[38] —. CCITT G.728 Coding of Speech at 16 kbps using Low-Delay Code-Excited Linear Prediction. Int. Telecommun. Union, Recommendation G.728, Sept. 1992.
[39] M. R. Schroeder and B. S. Atal. Code-Excited Linear Prediction (CELP): High Quality at very Low Bit Rates. *IEEE Int. Conference on Acoustics, Speech, and Signal Process.*, pp. 937-940, March 1985.
[40] B. Matschkal, F. Bergner, J. B. Huber. Joint Signal Processing for Spherical Logarithmic Quantization and DPCM. *6th Int. ITG Conference on Source and Channel Coding (SCC)*, Munich, Germany, April 2006.
[41] European Broadcast. Union (EBU). Demonstr. CD, SQAM (Sound Quality Assessment Material). Recordings for Subjective Tests, April 1988.
[42] —. ITU-T G.722 7 kHz audio-coding within 64 kbit/s. Int. Telecommun. Union, Recommendation G.722, Nov. 1988.
[43] N. Görtz. Aufwandsarme Qualitätsverbesserungen bei der gestörten Übertragung codierter Sprachsignale. Ph.D. dissertation. Christian Albrechts Universität, Kiel, 1998. ISBN 3-8265-4601-6.
[44] SLQ - Audio Examples available at: `http://www.LNT.de/~berndmat/SLQ_examples.html`
[45] The MathWorks, MATLAB 7.3.0, 2006. `http://www.mathworks.com`

Bernd Matschkal was born in Fürth, Germany, in 1975. He received the Dipl.-Ing. degree in electrical engineering in 2001 from the University of Erlangen–Nuremberg, Germany. From 2001 to 2008, he was research assistant at the Institute for Information Transmission of the University of Erlangen–Nuremberg from which he received the Dr.-Ing. degree for a thesis on Spherical Logarithmic Quantization. In 2008, he was an Akademischer Rat at the University of Erlangen–Nuremberg. Presently, he is with Siemens Audiology in Erlangen.

Johannes B. Huber received the Dipl.-Ing. degree in electrical engineering from the Technische Universität, München in 1977. From 1977 to 1982 he was research assistant at the Lehrstuhl für Nachrichtentechnik of the Universität der Bundeswehr, München, from which he received the Dr.-Ing. degree with a thesis on coding for channels with memory. From 1982 to 1990, he was an Akademischer Oberrat at the Universität der Bundeswehr, München and received the Dr. habil. degree with a thesis on trellis coded modulation. In 1991 he joined the IBM Research Laboratory, Zurich, Switzerland. Since autumn 1991, he is Professor at the University of Erlangen–Nuremberg, Germany. His research interests are information and coding theory, modulation schemes, algorithms for signal detection and adaptive equalization for channels with severe intersymbol interference, signalling, detection and equalization for multiple-input multiple-output (MIMO) channels and concatenated coding together with iterative decoding. Johannes B. Huber is Fellow of the IEEE, Corresponding Fellow of the Royal Society of Edinburgh, and a regular member of the Bavarian Academy of Sciences. He was a member of the board of governors of the IEEE Information Theory Society from January 1999 to December 2001 and again from January 2004 to December 2006. From 1994 to 2004, Johannes Huber was a member of the Editorial Board of the "International Journal on Electronics and Communications (AEÜ)" and between 1997 and 2002 he was Editor-in-Chief of this journal. From 1996 to 1999 he was an Associate Editor of the IEEE Transactions on Communications. He has also been Chairman and member of the program committees of several international conferences on information theory and coding. Johannes Huber is author or co-author of a textbook on trellis coding and about 250 papers in international journals or conference records. Papers authored or co-authored by him were awarded by the Best Paper Award of the German Society of Information Technology (ITG) in 1988, 2000, and 2006. He also received the Vodafone award for innovations in mobile communications 2004.