On Entropy-Constrained Vector Quantization Using Gaussian Mixture Models

David Y. Zhao, *Student Member, IEEE*, Jonas Samuelsson, Mattias Nilsson

***Abstract*—A flexible and low-complexity entropy-constrained vector quantizer (ECVQ) scheme based on Gaussian mixture models (GMMs), lattice quantization, and arithmetic coding is presented. The source is assumed to have a probability density function of a GMM. An input vector is first classified to one of the mixture components, and the Karhunen-Loève transform of the selected mixture component is applied to the vector, followed by quantization using a lattice structured codebook. Finally, the scalar elements of the quantized vector are entropy coded sequentially using a specially designed arithmetic coder. The computational complexity of the proposed scheme is low, and independent of the coding rate in both the encoder and the decoder. Therefore, the proposed scheme serves as a lower complexity alternative to the GMM based ECVQ proposed by Gardner, Subramaniam and Rao [1]. The performance of the proposed scheme is analyzed under a high-rate assumption, and quantified for a given GMM. The practical performance of the scheme was evaluated through simulations on both synthetic and speech line spectral frequency (LSF) vectors. For LSF quantization, the proposed scheme has a comparable performance to [1] at rates relevant for speech coding (20-28 bits per vector) with lower computational complexity.**

***Index Terms*—Entropy constrained, vector quantization, VQ, lattice, Gaussian mixture model, GMM, arithmetic coding**

I. Introduction

It is well-known that an entropy-constrained vector quantizer (ECVQ) achieves better rate-distortion performance compared to a resolution-constrained vector quantizer (RCVQ). This is due to the flexibility of assigning bit sequences of different lengths to different code vectors according to the probability of their appearance. However, classical codebook-training based ECVQ has been limited to low-rate vector quantizers (VQ), e.g. [2], because of the exponentially increasing computational complexity and memory requirements for higher rates.

Gaussian mixture modeling has been successfully applied to RCVQ, e.g., [3]–[7]. A Gaussian mixture model (GMM) has the form of a weighted sum of Gaussian density functions, and is suitable for modeling of real-world sources with an unknown probability density function (PDF). The GMM parameters can be optimized for a source using a training database [8]. Applied to RCVQ, the method of [4] achieves good rate-distortion performance, with a low computational complexity. In particular, the computational complexity of the method is independent of the coding rate, making vector quantization at a high rate feasible. The method of [4] is additionally rate scalable, and works over a large range of rates without the need of retraining the codebooks. For communication channels with errors, such a method facilitates rate-adaptation according to the channel condition.

Rate-independent complexity and rate-scalability are attractive features also for an ECVQ. It has been shown that, at a high rate, the quantization point density of an optimal ECVQ is uniform for a source with finite differential entropy [9], [10]. This theoretical result in combination with Gersho's conjecture [11] form the main motivation for the usage of a lattice structured codebook for vector quantization followed by entropy coding using, e.g., an arithmetic code [12], [13]. The regular structure of a lattice facilitates closest-point search algorithms (quantization) with low computational complexity, e.g, [14] for classical lattices and [15] for arbitrary lattices. For a tutorial on the "best known" lattices for quantization, see [16].

While the lattice structure simplifies the codebook design, the subsequent step of generating the variable-length bit sequence, is a non-trivial task for an ECVQ. A traditional arithmetic code is based on the probability mass function (PMF) of the code vectors, e.g., obtained through a histogram, and the cumulative distribution function (CDF) according to a particular ordering of the vectors. Two critical difficulties appear in such a design at a high rate [17]. First, the PMF storage and the CDF computation become infeasible as the number of code vectors increases exponentially with rate. Second, as the volumes of the quantization cells shrink, the available data may not provide a PMF estimate with enough accuracy. Popat and Picard proposed a solution, mainly for the second problem, using a GMM for describing the source PDF [17], [18]. Their solution is designed for quantization using a Z lattice and a GMM with diagonal covariance matrices. Since the CDF is computed by summation over the PMF (computed from the GMM), the computational complexity of [17], [18] grows exponentially with rate. Another solution is proposed by Gardner, Subramaniam and Rao [1], and we refer to their ECVQ as GSR-ECVQ. In GSR-ECVQ, a scalar conditional approximation is proposed, for which the PMF and CDF are computed scalar-wise, and conditioned on past scalar elements of the quantized vector. The GSR-ECVQ has a computational complexity that is independent of rate in the encoder and linear in rate in the decoder. The GSR-ECVQ also generalizes the method of [17], [18] by allowing for lattices other than the Z lattice, and a GMM with full covariance matrices.

An alternative ECVQ design is based on a classified VQ where an input vector is first classified to one of the mixture components. The input is quantized using a component-wise lattice quantizer and then coded using an entropy code. The

D. Y. Zhao and M. Nilsson are with the School of Electrical Engineering, (KTH) Royal Institute of Technology, Stockholm, Sweden. E-mail: david.zhao@ee.kth.se; mattias.nilsson@ee.kth.se.

J. Samuelsson is with Coding Technologies, Stockholm, Sweden. Email: js@codingtechnologies.com.

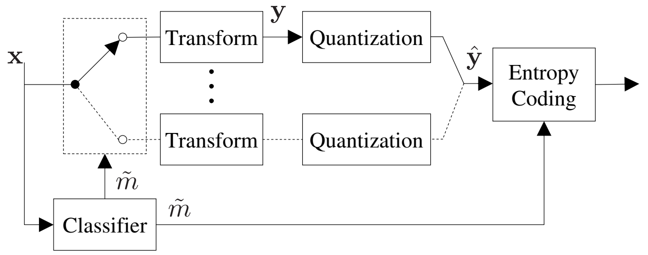

Fig. 1. Schematic diagram of a GMM-based classified ECVQ.

schematic diagram of such a scheme is shown in Fig. 1. The idea of a GMM-based classified ECVQ has appeared in the theoretical works of [19], [20]. However, a crucial component of such a design, a practically feasible entropy code, has not yet been reported.

In this paper, we propose a low-complexity ECVQ (LC-ECVQ) based on a classified VQ [19], [20] that enables the use of a, compared to [1], more practical arithmetic code. The proposed LC-ECVQ has a further reduced complexity than GSR-ECVQ at the cost of lower performance at high rates. In particular, the LC-ECVQ has a computational complexity that is independent of rate both in the encoder and the decoder. The reduced complexity is mainly due to the simplified arithmetic code design, which has the additional advantage of reduced sensitivity to numerical errors in the CDF computation. Furthermore, the proposed LC-ECVQ has the possibility of applying different lattices for different component quantizers depending on the local behavior of the source. This is useful for sources that are likely to contain vanishing dimensions [21]–[23] (the high-rate assumption is violated in these dimensions). We propose a lattice adaptation scheme for LC-ECVQ that adapt the lattice of each mixture component depending on the coding rate. We show experimentally for a speech derived source that LC-ECVQ with lattice adaptation achieves better performance at low rates than GSR-ECVQ with a fixed lattice.

II. Preliminaries

In this section, the notation and the problem formulation are first introduced. We then review the classical result of the optimal ECVQ at a high rate. Finally, the signal model using a GMM is presented.

A. Problem formulation

Let $\mathbf{x} = [x_1, \ldots, x_K]^T$ denote a K-dimensional vector drawn from a vector stationary random source, and $\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_K]^T$ denote the corresponding vector after quantization. Let $f(\mathbf{x})$ denote the PDF of $\mathbf{x}$, and $p(\hat{\mathbf{x}})$ denote the PMF of $\hat{\mathbf{x}}$.

In this paper, we consider the mean square error (MSE) as the average distortion, D, between the original and quantized variable,

$$D = \frac{1}{K} \int f(\mathbf{x}) ||\mathbf{x} - \hat{\mathbf{x}}||^2 d\mathbf{x}. \tag{1}$$

The quantized vectors are coded using a variable-rate code. The average rate of the generated bit stream is $\hat{H} = \sum_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}}) \ell(\hat{\mathbf{x}})$, where $\ell(\hat{\mathbf{x}})$ denotes the codeword length of $\hat{\mathbf{x}}$. The optimal variable-rate code has an average rate approaching the entropy of $\hat{\mathbf{x}}$,

$$H_{\text{lower}} = -\sum_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}}) \log_2 p(\hat{\mathbf{x}}). \tag{2}$$

Using an entropy code such as the arithmetic code [12], [13], this rate can be approached arbitrarily closely by increasing the sequence length.

The optimal ECVQ has a quantization point density function $g_c(\mathbf{x})$ that minimizes the average distortion D, under the constraint that the entropy H_{lower} equals a desired target rate R. The optimal $g_c(\mathbf{x})$ minimizes the extended criterion

$$\eta \quad = \quad D + \lambda \left(H_{\text{lower}} - R\right), \tag{3}$$

where λ is the Lagrange multiplier for the entropy constraint. The goal of this work is to design a flexible and low complexity ECVQ that performs closely to a system with the theoretically optimal $g_c(\mathbf{x})$.

B. High-rate ECVQ

The optimization problem (3) can be solved under a high-rate assumption, which implies that the data PDF is considered uniform within the boundary of a quantization cell. Assuming that the source has a finite differential entropy, denoted by h, it can be shown that the optimal $g_c(\mathbf{x})$ is a constant [11], [9],

$$g_c = 2^{R-h}. \tag{4}$$

The theoretical result implies that, at a high rate, uniformly distributed quantization points are optimal if the point indices are subsequently coded using an entropy code. This result is the main motivation for using a lattice structured codebook in an entropy-constrained quantizer. A high-rate ECVQ for a non-difference distortion measure is discussed in [24], [25].

Using a lattice quantizer, the MSE distortion D at a high rate is approximately [11]

$$D \quad \approx \quad \mathcal{C}_K g_c^{-\frac{2}{K}}, \tag{5}$$

where $\mathcal{C}_K$ is the normalized moment of inertia of the quantization cell shape, defined by

$$\mathcal{C}_K = \frac{1}{K} g_c^{\frac{K+2}{K}} \int_{S(\hat{\mathbf{x}})} ||\mathbf{x} - \hat{\mathbf{x}}||^2 d\mathbf{x}, \tag{6}$$

for the quantization cell shape $S(\hat{\mathbf{x}})$ of a lattice point $\hat{\mathbf{x}}$.

C. Gaussian mixture modeling

Throughout the remainder of this paper we assume that $f(\mathbf{x})$ is a GMM density with M mixture components,

$$f(\mathbf{x}) = \sum_{m=1}^{M} \rho_m f_m(\mathbf{x}), \tag{7}$$

where m denotes the component index, ρ_m denotes the mixture weight of the m'th component, and $f_m(\mathbf{x})$ is a component Gaussian PDF with mean $\boldsymbol{\mu}_m$ and covariance matrix $\mathbf{D}_m$. Let $\mathbf{D}_m = \mathbf{V}_m \boldsymbol{\Sigma}_m^2 \mathbf{V}_m^T$ be the eigenvalue decomposition, where $\boldsymbol{\Sigma}_m^2 = \text{diag}[\sigma_1^2, \ldots, \sigma_K^2]$ is a diagonal matrix consisting of the eigenvalues of $\mathbf{D}_m$. $\mathbf{V}_m^T$ is a decorrelating transform, also known as the Karhunen-Loève transform (KLT), of the mth mixture component.

III. LC-ECVQ ALGORITHM

The proposed LC-ECVQ encoding is a two step procedure. First, the input vector $\mathbf{x}$ is classified to one mixture component with index $\tilde{m}$. Next, $\mathbf{x}$ is quantized to $\hat{\mathbf{x}}$ using the $\tilde{m}$th per-component quantizer. Both the component index $\tilde{m}$ and the quantized vector $\hat{\mathbf{x}}$ are to be coded using an arithmetic code.

A. Classification

In LC-ECVQ, a maximum a posteriori (MAP) classifier is used to determine the index of the mixture component, $\tilde{m}$, to which the data vector $\mathbf{x}$ most likely belongs to,

$$\tilde{m} = \arg\max_{m} p(m|\mathbf{x}) = \arg\max_{m} \rho_m f_m(\mathbf{x}). \tag{8}$$

The MAP classifier selects the per-component quantizer for quantization of the data vector. An alternative classifier is to minimize a weighted sum of rate and distortion in a Lagrange formulation [19], [20]. The MAP classifier is used in LC-ECVQ for two main reasons. First, the Lagrange formulation would require quantization using all component quantizers (for distortion computation), which increases the computational complexity. Second, if the mixture components are sufficiently separated, the additional distortion term mainly affects the code vectors near the class boundaries, and their contribution to the overall performance is expected to be low. We further show in Section IV that the MAP classifier based solution has a bounded worst-case performance, which can be used to determine the expected performance of LC-ECVQ.

B. Transform and quantization

We transform $\mathbf{x}$ by subtracting the mean, and applying the KLT of the $\tilde{m}$th mixture component before lattice quantization,

$$\mathbf{y} = \mathbf{V}_{\tilde{m}}^T(\mathbf{x} - \boldsymbol{\mu}_{\tilde{m}}). \tag{9}$$

The transformed variable $\mathbf{y}$ is then quantized to the closest code vector neighbor in a lattice structured codebook.

A lattice Λ is generated through a scaled generating matrix $\mathbf{G}$:

$$\Lambda = \{c\ \mathbf{G}^T\mathbf{u} : \mathbf{u} \in \mathbb{Z}^k\}, \tag{10}$$

where c is a scaling factor, and $\mathbf{G}$ is the generating matrix of the form (without losing generality [15])

$$\mathbf{G} = \begin{bmatrix} g_{1,1} & 0 & \cdots & 0 \\ g_{2,1} & g_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{K,1} & g_{K,2} & \cdots & g_{K,K} \end{bmatrix}, \tag{11}$$

with positive diagonals $g_{k,k}$. We further assume that $\mathbf{G}$ is normalized such that the determinant is one.

The quantized vector using the $\tilde{m}$th lattice codebook is given by $\hat{\mathbf{y}} = c\ \mathbf{G}_{\tilde{m}}^T\hat{\mathbf{u}}$, for a particular $\hat{\mathbf{u}}$. We further normalize $\mathbf{y}$ by the square root of the eigenvalues,

$$\mathbf{y}' = \boldsymbol{\Sigma}_{\tilde{m}}^{-1}\mathbf{y}, \tag{12}$$

and the quantized vector is scaled similarly,

$$\hat{\mathbf{y}}' = c\ \boldsymbol{\Sigma}_{\tilde{m}}^{-1}\mathbf{G}_{\tilde{m}}^T\hat{\mathbf{u}}. \tag{13}$$

The classification (8), transform (9) and scalings (12-13) enable a simplified design of the arithmetic code. For a GMM with sufficiently separated mixture components, a Gaussian PDF with zero mean and identity covariance matrix approximates the PDF of $\mathbf{y}'$ well, and this is exploited in the entropy coding of the quantized and scaled $\hat{\mathbf{y}}'$ vector. Details regarding the encoding and decoding of the arithmetic code are given in Section V.

C. Lattice selection

In LC-ECVQ, different lattices can be used in different component quantizers. At a high rate and for a fixed dimension, the optimal lattice for quantization has the lowest normalized moment of inertia among all lattices. The best known lattices for dimensions up to ten are listed in [16]. However, those lattices are not necessarily the best at low rates. In particular, many sources derived from speech and images are likely to contain vanishing dimensions [21]–[23], and a GMM of such a source may have mixture components with some near-zero principal components. When an eigenvalue corresponding to a principal component is small compared to the quantizer step size, the assumption of high-rate does not apply to this dimension, and the best known lattice from [16] may not be the best choice for this rate.

In LC-ECVQ, we propose a heuristic lattice adaptation scheme for a given rate. For each mixture component, the eigenvalues are compared to a constant experimentally set to $2D$, where the high-rate distortion D is computed using (5) using the best-known K-dimensional lattice from [16]. The ϵ dimensions that have eigenvalues less than the constant are quantized using the Z lattice. The remaining $K - \epsilon$ dimensions are quantized using the best-known lattice in $K-\epsilon$ dimensions (the generating matrices are normalized to have the determinants equal to one) [16].

Fig. 2 shows a low-rate quantization example for a 2-dimensional source with one near-zero principle component. For this example, the LC-ECVQ with the Z_2 lattice performs better than with the A_2 (hexagonal) lattice, with both a lower distortion and a lower rate. The lower distortion is because that the code vectors using the Z_2 lattice are denser on the diagonal. The lower rate is because that the code vectors off the diagonal (using the A_2 lattice) are coded using much lower probabilities. The example provides an intuitive explanation of using the Z lattice in dimensions with small eigenvalues.

In the K-dimensional space, the lattice adaptation scheme corresponds to quantization using a hybrid lattice, for which the generating matrix are merged from the $\epsilon \times \epsilon$ dimensional identity matrix and the $(K - \epsilon) \times (K - \epsilon)$ dimensional lattice generating matrix. The hybrid generating matrices can be computed off-line. Therefore, the lattice adaptation scheme is not associated with any additional computations compared to using a traditional lattice. We show experimentally in Section VIII-A that the lattice adaptation scheme significantly improves the LC-ECVQ performance at low rates.

The high-rate optimal quantization point density (4) motivates the use of lattices with same cell volumes in all mixture component (a proof is given in [19], [20]). Therefore, c is

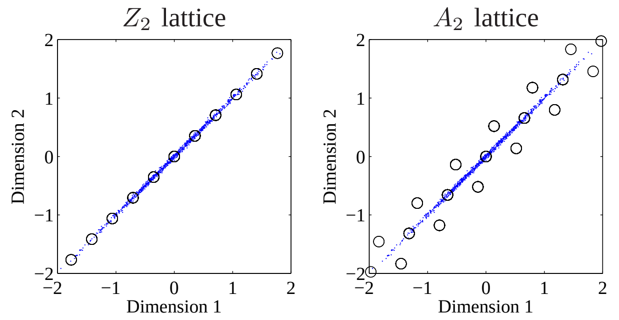

Fig. 2. Examples of quantization of a source with vanishing dimensions using LC-ECVQ based on the Z_2 lattice and the A_2 lattice (hexagonal lattice). The circles are the quantized vectors using the two lattices.

considered as a constant for a given R, and the same c is used in all component quantizers.

IV. Theoretical analysis

The performance of the proposed LC-ECVQ is analyzed theoretically under a high-rate assumption. Neglecting contributions from quantization cells located on the classification boundaries, the MSE distortion is given by (5) at a high rate. The rate of LC-ECVQ and its lower and upper bounds are derived in this section. We show that the performance loss due to classification can be predicted, and the loss approaches zero for a GMM with well-separated mixture components. Finally, we discuss the optimization of the quantizer for different target rates. The analysis of this section does not apply to the lattice adaptation scheme (which is a low-rate optimization), and assumes that the same lattice is used in all mixture components.

A. Rate performance

In the proposed LC-ECVQ, both the component index $\tilde{m}$ and the quantized vector $\hat{\mathbf{x}}$ are entropy coded. The average rate is given by

$$\hat{H} \;=\; -\sum_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}})(\log_2 \rho_{\tilde{m}} + \log_2 p_{\tilde{m}}(\hat{\mathbf{x}})), \tag{14}$$

where $\rho_{\tilde{m}}$ is the weight of the $\tilde{m}$'th mixture component and $p_{\tilde{m}}(\hat{\mathbf{x}}) = \int_{S(\hat{\mathbf{x}})} f_m(\mathbf{x})d\mathbf{x}$ is the PMF of $\hat{\mathbf{x}}$ using the $\tilde{m}$'th component PDF. Under a high-rate assumption, $\hat{H}$ can be approximated by, e.g., [26],

$$\hat{H} \;\approx\; -\int f(\mathbf{x}) \log_2 \left(\rho_{\tilde{m}} f_{\tilde{m}}(\mathbf{x})\right) d\mathbf{x} + \log_2 g_c, \tag{15}$$

where $g_c = \mathrm{vol}(\hat{\mathbf{x}})^{-1}$ denotes the inverse volume of the quantization cell for $\hat{\mathbf{x}}$. The integral of (15) can be solved through numerical integration. The integral is independent of the coding rate, and can be evaluated once for a given GMM density.

1) Lower bound: The lower performance bound is straightforward. Since $f(\mathbf{x}) \geq \rho_m f_m(\mathbf{x})$ for all m and using (4), we have

$$\begin{aligned} \hat{H} &\geq -\int f(\mathbf{x}) \log_2 f(\mathbf{x}) d\mathbf{x} + \log_2 g_c \\ &= h + \log_2 g_c \approx H_{\text{lower}}, \end{aligned} \tag{16}$$

which is the entropy of $\hat{\mathbf{x}}$ under a high-rate assumption.

2) Upper bound: The upper performance bound can be derived for the MAP classifier, where $\rho_{\tilde{m}} f_{\tilde{m}}(\mathbf{x}) \geq \rho_m f_m(\mathbf{x})$ for all m. We then have

$$\begin{aligned} \hat{H} &\approx -\int \sum_{m=1}^{M} \rho_m f_m(\mathbf{x}) \log_2 \left(\rho_{\tilde{m}} f_{\tilde{m}}(\mathbf{x})\right) d\mathbf{x} + \log_2 g_c \\ &\leq -\int \sum_{m=1}^{M} \rho_m f_m(\mathbf{x}) \log_2 \left(\rho_m f_m(\mathbf{x})\right) d\mathbf{x} + \log_2 g_c \\ &= H_M + \sum_{m=1}^{M} \rho_m h_m + \log_2 g_c \triangleq H_{\text{upper}}, \end{aligned} \tag{17}$$

where

$$H_M \;=\; -\sum_{m=1}^{M} \rho_m \log_2 \rho_m, \tag{18}$$

is the entropy of the component index, and

$$\begin{aligned} h_m &= -\int f_m(\mathbf{x}) \log_2 f_m(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{2} \log_2 \left((2\pi \mathrm{e})^K |\mathbf{D}_m| \right), \end{aligned} \tag{19}$$

is the differential entropy of the mth Gaussian component, with $|\cdot|$ denoting the determinant.

3) Performance prediction: Since the rate of the proposed LC-ECVQ is bounded, the maximum theoretical loss for a GMM density is the difference between the bounds,

$$H_{\text{diff}} \;=\; H_M + \sum_{m=1}^{M} \rho_m h_m - h. \tag{20}$$

Interestingly, H_{diff} is independent of the coding rate. Hence, once the GMM of a particular source is given, the maximum theoretical performance loss of using the proposed LC-ECVQ can be predicted.

By rearranging the terms, H_{diff} can be written as

$$H_{\text{diff}} \;=\; H_M - \sum_{m=1}^{M} \rho_m h(f_m || f), \tag{21}$$

where

$$h(f_m||f) \;=\; \int f_m(\mathbf{x}) \log_2 \frac{f_m(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x} \tag{22}$$

denotes the Kullback-Leibler distance between the component PDF and the mixture PDF.

It is interesting to quantify the extrema of H_{diff} over all possible GMM densities with the same number of mixture components. The maximum of H_{diff} can be obtained by simultaneously maximizing H_M and minimizing $\sum_{m=1}^{M} \rho_m h(f_m||f)$. We note that $h(f_m||f) \geq 0$ and equals zero if and only if $f_m(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x}$. The other term H_M is maximized when all component weights are equal. Therefore, the maximum H_{diff} occurs when all mixture components have the same PDF and the same weight, and

$$\max H_{\text{diff}} = \log_2 M. \tag{23}$$

In this case, the source has a Gaussian distribution, and using a GMM is unnecessary.

We consider now the other extreme scenario when the mixture components are located far from each other. In the limit, the mixture model can be considered as a partition of the space, for which the component PDF is non-zero only within its own partition. The Bayesian classification error of the MAP classifier then approaches zero. The Kullback-Leibler distance for such a model can be approximated as

$$h'(f_m||f) \approx \int f_m(\mathbf{x}) \log_2 \frac{f_m(\mathbf{x})}{\rho_m f_m(\mathbf{x})} d\mathbf{x} = -\log_2 \rho_m, \tag{24}$$

and

$$H'_{\text{diff}} \approx H_M - \sum_{m=1}^{M} \rho_m(-\log_2 \rho_m) = 0. \tag{25}$$

Therefore, for a GMM with well-separated mixture components, H_{diff} approaches zero, and the proposed LC-ECVQ approaches the theoretically optimal performance. In the experiments we observed a decreasing H_{diff} with increasing separation of the GMM mixture components (see section VIII-A).

B. Rate adjustment

Similar to GSR-ECVQ, the LC-ECVQ is rate scalable. The target rate R can be changed to allow seamless adaptation to, e.g., varying channel conditions. The rate scalability is achieved by modifying the lattice scaling factor, c, so that different rate-distortion operating points are obtained.

The lattice scaling factor, c, is related to the point density function of the lattice codebook,

$$g'_c(\mathbf{x}) = |c\mathbf{G}|^{-1}. \tag{26}$$

Setting the rate $\hat{H}$ to be equal to R, we get

$$c = \left(|\mathbf{G}| 2^{R + \int f(\mathbf{x}) \log_2(\rho_{\tilde{m}} f_{\tilde{m}}(\mathbf{x})) d\mathbf{x}} \right)^{-\frac{1}{K}}. \tag{27}$$

The integral can be solved through numerical integration. Again, the integral is independent of the coding rate, and can be evaluated once for a given GMM density.

V. Implementation

In this section, we discuss implementational aspects of the proposed LC-ECVQ, and in particular the entropy coding. The implementation of the lattice quantizer follows standard procedures [14], [15] (apart from truncation, cf. Section V-F). We assume that the input data vector has been classified and quantized according to Section III.

A. Arithmetic coding

The component index $\tilde{m}$ and the scaled quantized vector $\hat{\mathbf{y}}'$ are to be entropy coded. The proposed LC-ECVQ uses a specially designed arithmetic code to generate the bit sequence. After classification, KLT, quantization and scaling, we encode a sequence consisting of the component index and the scalar indices of the scaled quantized vector, $[\tilde{m}, \hat{y}'_K, \ldots, \hat{y}'_1]$. Note that the scalar indices of $\hat{\mathbf{y}}'$ are encoded in the reverse order for practical reasons (cf. Section V-E). In [1], the PMF of each scalar element is computed from a conditional GMM conditioned on the previous quantized scalar elements. In LC-ECVQ, the scalar elements are coded independently (only conditioned on the mixture component index), and the PMF is computed using a Gaussian density function with zero mean and unit variance. Therefore, the arithmetic encoder of LC-ECVQ is expected to have a reduced computational complexity compared to GSR-ECVQ. We show in Section VII that the proposed arithmetic code has advantages of reduced complexity also in the decoder and facilitates an implementation that is less sensitive to numerical errors in the CDF computation.

The component index is coded according to the component weights. An implementation for the Z lattice is presented in Sections V-B and V-C, and the extension to an arbitrary lattice is discussed in Sections V-D and V-E. The arithmetic code for an arbitrary lattice is based on a similar idea proposed in [1]. Since the presentation in [1] is incomplete (e.g., the decoder details are missing), we present the implementation also for an arbitrary lattice with detailed steps.

B. Encoding for the Z lattice

The Z lattice has the generating matrix $\mathbf{G} = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. The scalar elements of $\hat{\mathbf{y}}'$ are encoded sequentially (the reverse ordering is not required for the Z lattice case, cf. Section V-E), and the encoding of the kth element is described here. The kth scalar element, $\hat{y}'_k$, is located on a grid with points at integer multiples of Δ_k, where

$$\Delta_k = \frac{c}{\sigma_k}. \tag{28}$$

We use a Gaussian density function zero-mean and unit-variance for all k. Therefore, the same CDF, $\Phi(\cdot)$, applies for all dimensions,

$$\Phi(y'_k) = \int_{-\infty}^{y'_k} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}z^2} dz = \frac{1}{2}\mathrm{erf}(\frac{1}{\sqrt{2}} y'_k) + \frac{1}{2}, \tag{29}$$

where $\mathrm{erf}(\cdot)$ is the Gauss error function. Thus, the interval $\left[\Phi(\hat{y}'_k - \frac{1}{2}\Delta_k), \Phi(\hat{y}'_k + \frac{1}{2}\Delta_k)\right)$ is used as the probability interval for the arithmetic encoder. The $\mathrm{erf}(\cdot)$ function has no closed-form expression and needs to be evaluated using numerical approximations. See Section VII-B for a discussion on the necessary conditions for the approximated $\Phi(\cdot)$ function. The practical implementation of the encoder additionally requires rounding of the probability interval to numbers with some predefined finite precision (see [12], [13], [27]).

C. Decoding for the Z lattice

Decoding of the bit sequence is performed for each dimension k. The bit sequence is translated to a floating point number, ξ, between the rounded probability interval of $\hat{y}'_k$. Since the point $\hat{y}'_k$ is located on the grid consisting of integer multiples of Δ_k, it is possible to search among the grid points for the one with the probability interval containing ξ. This idea is used in the decoder of GSR-ECVQ.

The use of the $\Phi(\cdot)$ function in the encoder simplifies the decoder in the proposed LC-ECVQ. Since $\Phi(\cdot)$ is a monotonically increasing function, an inverse function of $\Phi(\cdot)$ exists. The inverse of the $\Phi(\cdot)$ function can be used to map ξ back to a point that is located near $\hat{y}'_k$. If the probability interval of $\hat{y}'_k$ is not rounded, the nearest grid point of $\Phi^{-1}(\xi)$ can be used to obtain the decoded $\hat{y}'_k$. In a practical implementation, when the probability interval is rounded using finite-precision numbers, additional neighboring points should be checked due to the rounding errors. We propose a truncation bound in Section V-F that allows us to restrict the search to the two nearest grid points of $\Phi^{-1}(\xi)$. The decoded $\hat{y}'_k$ are selected to be the candidate that has the probability interval $\left[\Phi(\hat{y}'_k - \frac{1}{2}\Delta_k), \Phi(\hat{y}'_k + \frac{1}{2}\Delta_k)\right)$ containing ξ. Compared to GSR-ECVQ, which requires on the order of R/K times CDF computations, the decoder of LC-ECVQ has a significantly lower computational complexity.

D. Encoding for an arbitrary lattice

The arithmetic encoding and decoding procedures for the Z lattice can be generalized to arbitrary lattices. The difference is the approximation of the Voronoi region by hyper-rectangular regions of the same volume to facilitate the PMF and CDF calculations [1]. For the arithmetic encoding, the hyper-rectangular region has length $cg_{k,k}$ in dimension k, and is centered at the quantized lattice vector from the lattice codebook. This approximation is a consequence of encoding the quantized vector as a sequence of scalar elements.

Unlike the Z lattice, the kth scalar value $\hat{y}'_k$ is located on a *sliding* one-dimensional equally-spaced grid determined by a step size Δ_k, and a dimension dependent offset o_k that must be obtained for each dimension. The grid step size Δ_k is determined according to

$$\Delta_k = \frac{cg_{k,k}}{\sigma_k}. \tag{30}$$

In the arithmetic encoder the offset is already given, implicitly, by the vector $\hat{y}_k$ that the lattice encoder provides. Therefore, the encoder is similar to the Z lattice encoder, and the interval $\left[\Phi(\hat{y}'_k - \frac{1}{2}\Delta_k), \Phi(\hat{y}'_k + \frac{1}{2}\Delta_k)\right)$ is used in the arithmetic encoder. However, in the decoder, the offset o_k must be handled explicitly (cf. Section V-E).

E. Decoding for an arbitrary lattice

The decoder for an arbitrary lattice is similar to the one for the Z lattice. The main difference is that the grid is determined by a step size Δ_k and an additional sliding offset o_k. Therefore, it is essential to determine the offset o_k in the decoder.

In [1], implementational details of the decoder for an arbitrary lattice are not presented. We here propose one solution based on a layered decomposition of a lattice [15]. We show that the offset o_k depends on the decoded scalar values from $k+1$th to Kth dimensions, and can be obtained recursively, starting from the Kth dimension. This dependency motivates the reverse ordering of the scalar sequence in the encoder.

We now present a recursive algorithm to obtain the offset vector $\mathbf{o} = [o_1, \ldots, o_K]^T$. The Kth scalar value is located on a grid with $o_K = 0$, due to the lower triangular form of $\mathbf{G}$ (11). Furthermore, let $\mathbf{P}_{K-1}$ be a matrix containing the upper $K-1$ rows of $\mathbf{G}_{\tilde{m}}$ and $\mathbf{g}_K^T$ the last row of $\mathbf{G}_{\tilde{m}}$. Let $\hat{\mathbf{u}}_{K-1}$ be the upper $K-1$ elements of $\hat{\mathbf{u}}$ and $\hat{u}_K$ the last element of $\hat{\mathbf{u}}$. Then $\hat{\mathbf{y}}'$ can be written as

$$\begin{aligned} \hat{\mathbf{y}}' &= c\, \mathbf{\Sigma}_m^{-1} \begin{bmatrix} \mathbf{P}_{K-1}^T & \mathbf{g}_K \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_{K-1} \\ \hat{u}_K \end{bmatrix} \\ &= c\, \mathbf{\Sigma}_m^{-1} \mathbf{P}_{K-1}^T \hat{\mathbf{u}}_{K-1} + c\, \mathbf{\Sigma}_m^{-1} \mathbf{g}_K \hat{u}_K. \end{aligned} \tag{31}$$

Using this lattice decomposition, $c\, \mathbf{\Sigma}_m^{-1} \mathbf{P}_{K-1}$ is a generating matrix for a lattice lying in a $K-1$ dimensional subspace, and the vector $\hat{\mathbf{y}}' - c\, \mathbf{\Sigma}_m^{-1} \mathbf{g}_K \hat{u}_K$ is a lattice vector. The offset for dimension $K-1$ can be obtained by letting $\mathbf{o} = c\, \mathbf{\Sigma}_m^{-1} \mathbf{g}_K \hat{u}_K$. The offset is then given by o_{K-1}. For other dimensions, the generating matrix is decomposed similarly, and the offset vector $\mathbf{o}$ is accumulated recursively. The process continues until all dimensions of $\hat{\mathbf{y}}'$ have been decoded.

F. Lattice truncation

The PMF of each code-vector must be greater than a minimum allowed probability to allow for a practical implementation on a finite-precision computer. If a 31 bit integer is used, this threshold is $\delta = 2^{-29}$ [27]. Additionally, the decoding algorithm (Section V-C) requires a bounded rounding error in a finite-precision implementation of $\Phi(\cdot)$. Therefore, for the kth dimension, we ensure that all $\hat{y}'_k$ are within a truncation interval such that

$$\frac{1}{2}(\Phi(\hat{y}'_k + \frac{1}{2}\Delta_k) - \Phi(\hat{y}'_k - \frac{1}{2}\Delta_k)) \geq \delta, \tag{32}$$

is fulfilled. If Φ and its inverse are implemented using a numerical approximation, the truncation should be applied such that the approximated Φ fulfills the condition (32) for all $\hat{y}'_k$ within the truncation interval.

To avoid the integration in solving the truncation interval (32), we derive a sufficient condition of (32) based on an approximation using the lower Riemann sum. Due the symmetry of the Gaussian PDF, we consider only the bound for $\hat{y}'_k > 0$, and the sufficient condition can be formulated as

$$\frac{1}{2\pi} \mathrm{e}^{-\frac{1}{2}(\hat{y}'_k + \frac{\Delta_k}{2})^2} \cdot \Delta_k \geq 2\delta. \tag{33}$$

The sufficient condition (33) can be relaxed for large Δ_k. For sufficiently large Δ_k, the truncation interval contains at least one Δ_k. Combining the two conditions, we get a sufficient truncation interval of $\hat{y}'_k$ that satisfies

$$|\hat{y}'_k| \quad \leq \quad \max\left(\sqrt{\log\frac{\Delta_k^2}{8\delta^2\pi}} - \frac{1}{2}\Delta_k, \frac{1}{2}\Delta_k\right). \tag{34}$$

All $\hat{y}'_k$ within the sufficient truncation interval (34) also fulfills (32). The sufficient truncation interval (34) is used in our implementation of LC-ECVQ.

VI. The algorithm

In this section, the complete algorithm for the proposed entropy-constrained lattice vector quantizer is given. Some details related to arithmetic coding are neglected and can be found in, e.g., [12], [13], [27].

The encoding of $\mathbf{x}$ at a given average rate R follows:

1) Solve for c using (27).
2) If the lattice adaptation scheme is used, construct all lattices according to Section III.
3) Classify $\mathbf{x}$ to the $\tilde{m}$th component using (8).
4) Encode the index $\tilde{m}$.
5) Set $\mathbf{y} = \mathbf{V}_{\tilde{m}}^T(\mathbf{x} - \boldsymbol{\mu}_{\tilde{m}})$.
6) Quantize $\mathbf{y}$ to the nearest lattice point, $\hat{\mathbf{y}} = c\mathbf{G}_{\tilde{m}}^T\hat{\mathbf{u}}$.
7) Set $\hat{\mathbf{y}}' = \boldsymbol{\Sigma}_m^{-1}\hat{\mathbf{y}}$.
8) Set $\mathbf{o}$ to a K-dimensional zero vector.
9) For $k = K$ to 1 with step of -1,
    a) Get the step size Δ_k using (30).
    b) Determine the equally-spaced grid with step size Δ_k and offset o_k.
    c) Verify that $\hat{y}'_k$ is within the truncation interval (34). If not, round to the nearest grid point in the interval.
    d) Encode $\hat{y}'_k$ using the probability interval $\left[\Phi(\hat{y}'_k - \frac{1}{2}\Delta_k), \Phi(\hat{y}'_k + \frac{1}{2}\Delta_k)\right)$.
    e) Set $\mathbf{o} = \mathbf{o} + c\ \boldsymbol{\Sigma}_{\tilde{m}}^{-1}\hat{u}_k\mathbf{g}_k$.

The decoding follows:

1) Decode $\tilde{m}$.
2) Set $\mathbf{o}$ to a K-dimensional zero vector.
3) for $k = K$ to 1 with step of -1,
    a) Get the step size Δ_k using (30).
    b) Determine the equally-spaced grid with step size Δ_k and offset o_k.
    c) Decode the bit sequence to a floating point number ξ between zero and one.
    d) Compute $\Phi^{-1}(\xi)$ using the inverse function of (29), and find its two nearest points on the grid.
    e) Set $\hat{y}'_k$ to one of the two grid points that has the probability interval $\{\Phi(\hat{y}'_k - \frac{1}{2}\Delta_k), \Phi(\hat{y}'_k + \frac{1}{2}\Delta_k)\}$ containing ξ.
    f) Determine $\hat{u}_k = (\hat{y}'_k - o_k)/\Delta_k$.
    g) Set $\mathbf{o} = \mathbf{o} + c\ \boldsymbol{\Sigma}_{\tilde{m}}^{-1}\hat{u}_k\mathbf{g}_k$.
4) Set $\hat{\mathbf{x}} = \mathbf{V}_{\tilde{m}}\boldsymbol{\Sigma}_{\tilde{m}}\hat{\mathbf{y}}' + \boldsymbol{\mu}_{\tilde{m}}$.

VII. Discussions

In this section, we analyze the computational complexity of the proposed LC-ECVQ. We also discuss the necessary conditions for numerical approximation of the CDF computation (29).

A. Computational complexity

The computational complexity of the LC-ECVQ algorithm (Section VI) is analyzed in this section. The algorithm of the encoder can be divided into classification, transformation, quantization and arithmetic coding. The classification step computes (8) for each mixture component, and the computational complexity of (8) is on the order of $O(K^2)$ due to the matrix multiplication. The transformation step (9) also has a matrix multiplication and its complexity is on the order of $O(K^2)$. The quadratically increasing complexity applies to a GMM with full-covariance matrices. If more structured covariance matrices are used, the complexity in the classification and transformation steps can be reduced. The complexity associated with the quantization depends on the lattice type. For a Z lattice, the computational complexity is $O(K)$. For an arbitrary lattice, the general search algorithm of [15] can be used. The algorithm of [15] relies on the geometrical structure of the lattice and its particular form of generating matrix. It has a bounded computational complexity that is exponential with respect to dimension $O(c^K)$. The arithmetic coding step has a complexity on the order of $O(K)$. In all encoding steps, the computational complexity is independent of the coding rate.

For the decoder, the arithmetic coding steps have a computational complexity that is on the order of $O(K)$. The final transformation step requires a matrix multiplication. Therefore, the overall complexity of the decoder is on the order of $O(K^2)$.

Compared to GSR-ECVQ, LC-ECVQ has lower computational complexity in both the encoder and the decoder. For instance, the arithmetic encoder of GSR-ECVQ requires $2M$ times computation of erf$(\cdot)$ for each dimension, M times more than LC-ECVQ. In the decoder, the GSR-ECVQ requires a search algorithm on the order of $O(R)$, where each search step involves at least one computation of erf$(\cdot)$. Therefore, the decoder of GSR-ECVQ is more complex than LC-ECVQ. A comparison in code execution time for a speech derived source is presented in Section VIII-B.

B. Numerical approximation of the CDF computation (29)

Computing the CDF is a key step in the proposed LC-ECVQ as well as in GSR-ECVQ. A problem is however that the erf$(\cdot)$ function has no closed-form solution and has to be evaluated using numerical approximations. The arithmetic code in GSR-ECVQ is based on a CDF consisting of a weighted sum of erf$(\cdot)$ functions. Special care is needed to handle to the accumulated approximation error of each erf$(\cdot)$ function that may lead to additionally reduced performance [1, pp. 93].

The arithmetic code of LC-ECVQ is based on a Gaussian CDF (29), $\Phi(\cdot)$. Compared to GSR-ECVQ, LC-ECVQ is less sensitive to approximation errors in the CDF computation. Proper encoding and decoding in LC-ECVQ requires 1) the approximated $\Phi(\cdot)$ is a monotonically increasing function between zero and one, 2) the same approximation is applied in both the encoder and the decoder, such that $a = \Phi^{-1}(\Phi(a))$ for all a, and 3) The PMF of each cell is large enough such that the condition (32) is fulfilled for all $\hat{y}'_k$ within the truncation interval. In a practical system, the $\Phi(\cdot)$ function and its inverse can be implemented using a table lookup and linear interpolation between table entries.

VIII. Experiments and results

In this section, we evaluate the performance of the proposed LC-ECVQ and present the experimental results. The algorithm was implemented according to Section VI in the Matlab environment. The built-in Matlab function for erf$(\cdot)$ and its inverse were used in our implementation. Experiments were performed for two different types of sources: artificial sources and one speech derived source. In each evaluation, the rates were evaluated by counting the number of bits per source vector using the encoded bit sequence.

A. Artificial source

Evaluation of the proposed LC-ECVQ was first performed using artificially generated sources in a fully controlled manner. For comparison, the training based ECVQ [2] was implemented and evaluated as the reference system. In this experiment, two dimensional sources were used to allow visualization of the source. In two dimensions, the hexagonal lattice is optimal for quantization (MSE criterion), and was used in LC-ECVQ.

The GMMs used for generating the source vectors have four mixture components and uniform component weights. Each mixture component has the identity matrix as the covariance matrix. The mean vectors are located on the corners of a cube centered on the origin. A number of GMMs are generated as a function of the distance between the mean vectors and the origin. The distance, denoted as r, was considered as a parameter specifying the degree of component separability. Two GMM models with $r = 1$ and $r = 3$ are visualized in Fig. 3. The GMM model with $r = 1$ represent the non-ideal quantization condition as the mixture components are located close to each other. The GMM model with $r = 3$ represent a more ideal quantization condition as the mixture components are located far apart. An evaluation data set and a training data set (for codebook training of the reference system) are generated for each GMM, and each data set consists of 10^6 vectors.

For each evaluation data set, the vectors were quantized, encoded, and decoded for various target rates. The MSE distortion as a function of rate (per vector) is plotted in Fig. 4 for $r = 1$ and $r = 3$. For both test cases, the performance of the reference system, the training based ECVQ, is very close to the optimal performance predicted by the high-rate theory (16). The proposed LC-ECVQ approaches the predicted performance (15) for rates above around seven bits per vector. The performance gap between the proposed LC-ECVQ and the reference system is about one bit for $r = 1$ case, indicating that the sub-optimality of the method for a GMM with heavily overlapped components. However, if the components are well separated, such as $r = 3$, the two bounds coincide and the proposed LC-ECVQ approaches the theoretically optimal performance at a high rate.

Next, the maximum theoretical loss of using the proposed LC-ECVQ for a model (20) is evaluated for artificial models with different r. The loss (H_{diff}) as a function of r is shown in Fig. 5. The results show that the maximum loss occurs when the distance is zero. In this case, the mixture components completely overlap, and the loss is two bits (=$\log_2 4$ as four mixture components were used). The performance loss decreases rapidly for increased separation of the mixture components. For $r \geq 3$, the theoretical performance loss is near zero, and the high-rate performance of LC-ECVQ approaches the theoretical optimum.

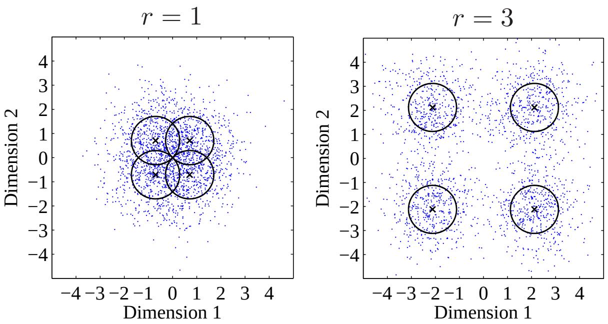

Fig. 3. Examples of a non-ideal GMM (left) and an ideal GMM (right). The circles represent the covariance matrices and crosses the mean vectors of GMM components. The distance r was measured between the mean vectors and the origin. The dots are the scatter plot of the data generated from the models.

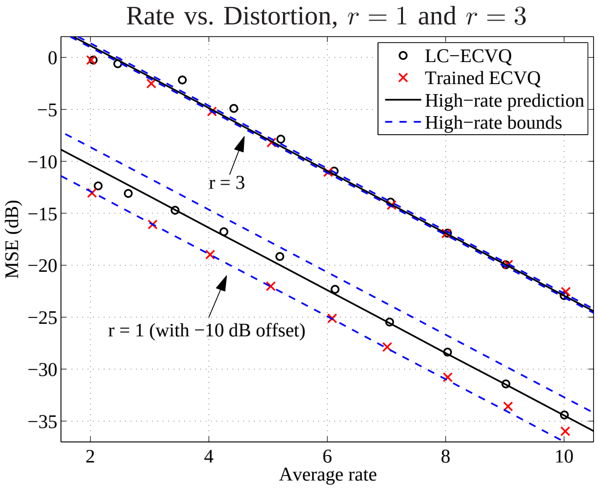

Fig. 4. MSE distortion as a function of rate (bits per vector) for the artificial sources.

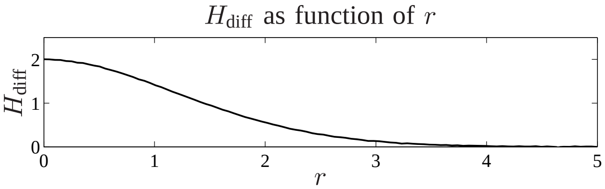

Fig. 5. Maximum theoretical loss as a function of distance to the origin.

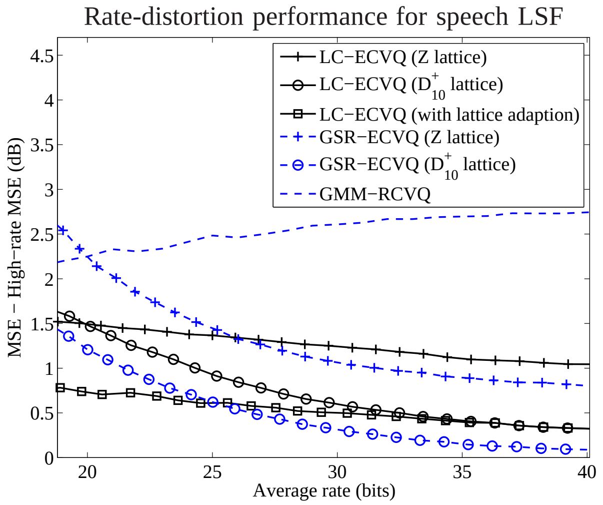

Fig. 6. Relative MSE distortion (compared to the high-rate distortion for a D_{10}^+ lattice quantizer) as a function of rate for the LSF source.

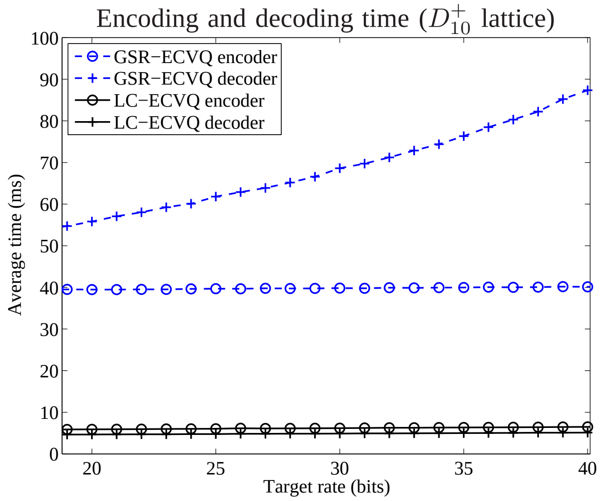

Fig. 7. Average Matlab execution time (per vector) using the proposed LC-ECVQ and GSR-ECVQ on one test computer.

B. Speech derived source

In this section, the proposed LC-ECVQ is evaluated using a ten-dimensional speech line spectral frequency (LSF) source, extracted from the TIMIT speech database using the Adaptive Multi-Rate (AMR) speech codec [28]. The speech signals were downsampled to 8kHz, and LSF vectors were extracted before quantization. The training set consists of about 700000 vectors and the evaluation set consists of 50000 vectors.

A GMM with $M = 16$ mixture components with full covariance matrices was trained over the training set using the expectation-maximization (EM) algorithm [29]. For this model, the maximum theoretical performance loss (20) was evaluated to be around 0.65 bits per vector. For lattice quantization, we have considered the Z lattice and the D_{10}^+ lattice [16]. The D_{10}^+ lattice is the best known lattice for quantization in ten dimensions and its normalized moment of inertia is 0.0708, compared to 0.0833 for the Z lattice [16]. Therefore, the theoretical MSE difference of the two lattice quantizers at a high rate is about 0.71 dB.

The reference systems in this experiment are the GSR-ECVQ [4] and the resolution-constrained GMM vector quantizer (GMM-RCVQ) [4] adapted to use the MSE distortion criterion. The GMM-RCVQ is based on scalar quantization in the transformed domain, equivalent to using a Z lattice for quantization. For a fair comparison, the same GMM was used in all three methods. The methods were implemented using shared codes and similar parameter settings when possible to minimize irrelevant performance mismatch. The training based ECVQ [2] is excluded in this experiment due to the unmanageable complexity and memory requirement for the tested rates.

The MSE distortion of the quantizers was evaluated over the evaluation set for different target rates. For clarity, the MSE distortion was subtracted by the high-rate MSE distortion (5) for a D_{10}^+ lattice quantizer, and the differences as a function of rate (per vector) are shown in Fig. 6.

GMM-RCVQ performs more than 2 dB above the high-rate MSE of the theoretically optimal ECVQ. At rates above 20 bits, ECVQ methods have a higher performance also in practice. Comparing the ECVQ methods, LC-ECVQ performs worse than GSR-ECVQ for rates higher than 26 bits for the Z lattice. The performance difference at a high rate is less than 0.3 dB MSE. At rates lower than 26 bits, LC-ECVQ performs better than GSR-ECVQ, as the Z lattice vectors of LC-ECVQ are aligned along the principal axes of each mixture component. At 20 bits, LC-ECVQ has an MSE that is about 0.7 dB lower than ECVQ. For the D_{10}^+ lattice, LC-ECVQ performs consistently worse than GSR-ECVQ, with about 0.3 dB higher MSE. Comparing the two lattices, the MSE differences at a high rate are about 0.7 dB for both methods, consistent with the theory. At low rates, the MSE difference between the two lattices decreases for the LC-ECVQ method, which demonstrates that the D_{10}^+ lattice is not necessary the best lattice at low rates. Finally, using the lattice adaptation scheme, the performance of LC-ECVQ is improved significantly at low rates, and the MSE distortion is lower than GSR-ECVQ with the D_{10}^+ lattice for rates below 25 bits. The MSE difference between the Z lattice and adaptive lattices is almost constant, indicating the space-filling advantage of vector quantization when the high-rate optimal (best known) lattices are used only in those dimensions with large principal components.

Next, the computational complexity of the two ECVQ methods are evaluated by measuring the average code execution time (per vector) on one test computer (a Pentium 4 at 3.0 GHz running Linux). The measurements as a function of the rate are shown in Fig. 7 for the D_{10}^+ lattice case. The code execution time of LC-ECVQ is almost constant for both the encoder and the decoder. For GSR-ECVQ, it is constant for only the encoder and increases with rate for the decoder. The code execution time of LC-ECVQ is about 6 times lower than GSR-ECVQ for the encoder, and 10-18 times lower for the decoder depending on the rate.

IX. Summary and conclusion

We have proposed a new flexible and low-complexity entropy-constrained vector quantizer, LC-ECVQ, using lattice codebooks, Gaussian mixture modeling, and arithmetic coding for the MSE distortion measure. The proposed LC-ECVQ is a low complexity alternative to the GSR-ECVQ proposed by Gardner, Subramaniam and Rao [1]. Compared to GSR-ECVQ, LC-ECVQ is sub-optimal due to the component classification and the arithmetic coding based on a Gaussian density function. The performance bound by using the proposed LC-ECVQ is derived under a high-rate assumption, and the theoretically loss is quantified for a given GMM density. We observed in our experiments that the loss decreases with increased separation of the GMM mixture components. For a ten-dimensional speech LSF source, the maximum theoretical loss is 0.65 bits per vector. This suboptimality was demonstrated experimentally for the same source, where GSR-ECVQ achieved up to 0.3 dB lower MSE distortion than the proposed LC-ECVQ for rates above 25 bits per vector.

The advantages of the proposed LC-ECVQ over GSR-ECVQ are three-fold. First, the proposed LC-ECVQ has a computational complexity that is independent of the coding rate in both the encoder and the decoder (the decoder complexity of GSR-ECVQ increases linearly with rate). In the LSF quantization experiment, the code execution time of LC-ECVQ is 6 times lower than GSR-ECVQ in encoding, and 10 to 18 times lower in decoding depending on the rate. The reduced complexity is due to the simplified arithmetic coding design based on a Gaussian density function. Second, the proposed LC-ECVQ facilitates a better performance at low rates, when the source exhibits vanishing dimensions (the high-rate assumption is violated in these dimensions). The proposed LC-ECVQ has the possibility of applying different lattices for different component quantizers depending on the local behavior of the source. We have proposed a heuristic lattice adaptation scheme in which the near-vanishing dimensions (e.g., low eigenvalues compared to the MSE) are quantized using the Z lattice. The remaining dimensions are quantized using the best-known lattice in that dimension [16]. Applied to the LSF source, the LC-ECVQ using lattice adaptation achieves better performance than GSR-ECVQ using the D_{10}^{+} lattice [16] for rates below 25 bits. Finally, the proposed LC-ECVQ is less sensitive to approximation errors in the CDF computation, which simplifies the implementation on a finite-precision computer.

We conclude that, for the LSF source, the performance of the proposed LC-ECVQ is at least comparable to GSR-ECVQ at rates relevant for speech coding (20-28 bits per vector), whereas the computational complexity is much reduced. The proposed LC-ECVQ can be applied to other sources by adapting the GMM to the new data. An evaluation of the maximum theoretical loss (21) can give an indication of the performance. The performance will be competitive if the mixture components are sufficiently separated.

References

[1] A. D. Subramaniam, "Gaussian mixtures models in compression and communication," Ph.D. dissertation, University of California, San Diego (UCSD), 2004.

[2] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 1, pp. 31–42, Jan. 1989.

[3] M. Tasto and P. Wintz, "Image coding by adaptive block quantization," *IEEE Trans. Communications*, vol. 19, no. 6, pp. 957–972, 1971.

[4] A. D. Subramaniam and B. D. Rao, "PDF optimized parametric vector quantization of speech line spectral frequencies," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 2, pp. 130–142, Mar. 2003.

[5] P. Hedelin and J. Skoglund, "Vector quantization based on Gaussian mixture models," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 4, pp. 385–401, Jul. 2000.

[6] J. Samuelsson and P. Hedelin, "Recursive coding of spectrum parameters," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 5, pp. 492–503, Jul. 2001.

[7] A. D. Subramaniam, W. R. Gardner, and B. D. Rao, "Low-complexity source coding using Gaussian mixture models, lattice vector quantization, and recursive coding with application to speech spectrum quantization," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 524–532, Mar. 2006.

[8] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[9] T. Linder and K. Zeger, "Asymptotic entropy-constrained performance of tessellating and universal randomized lattice quantization," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 575–579, Mar. 1994.

[10] A. György, T. Linder, P. Chou, and B. Betts, "Do optimal entropy-constrained quantizers have a finite or infinite number of codewords?" *IEEE Trans. Inform. Theory*, vol. 49, no. 11, pp. 3031–3037, Nov. 2003.

[11] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. 25, no. 4, pp. 373–380, Oct. 1979.

[12] R. Pasco, "Source coding algorithms for fast data compression," Ph.D. dissertation, Stanford Univ., 1976.

[13] J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM Journal of Research and Development*, vol. 20, p. 198, 1976.

[14] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 227–232, Mar. 1982.

[15] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[16] E. Agrell and T. Eriksson, "Optimization of lattices for quantization," *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 1814–1828, Sep. 1998.

[17] K. Popat and R. Picard, "Cluster-based probability model applied to image restoration and compression," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 5, Apr. 1994, pp. 381–384.

[18] ——, "Cluster-based probability model and its application to image and texture processing," *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 268–284, Feb. 1997.

[19] R. Gray, "Gauss mixture vector quantization," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 3, 2001, pp. 1769–1772.

[20] W. B. Kleijn, "A basis for source coding," Jan. 2003, lecture notes, KTH, Stockholm.

[21] W. Klein, R. Plomp, and L. C. W. Pols, "Vowel spectra, vowel spaces, and vowel identification," *J. Acoust. Soc. Am.*, vol. 48, no. 4, pp. 999–1009, 1970.

[22] R. Togneri, M. D. Alder, and Y. Attikiouzel, "Dimension and structure of the speech space," in *IEE Proc.-I, Communications, Speech and Vision*, vol. 139, no. 2, 1992, pp. 123–127.

[23] M. Nilsson and W. B. Kleijn, "On the estimation of differential entropy from data located on embedded manifolds," *IEEE Trans. Inform. Theory*, 2007, to appear.

[24] T. Linder and R. Zamir, "High-resolution source coding for non-difference distortion measures: the rate-distortion function," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 533–547, Mar. 1999.

[25] T. Linder, R. Zamir, and K. Zeger, "High-resolution source coding for non-difference distortion measures: multidimensional companding," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 548–561, Mar. 1999.

[26] R. M. Gray, *Source coding theory*. Kluwer Academic Publisher, 1990.

[27] E. Bodden, M. Clasen, and J. Kneis, "Arithmetic Coding in revealed - a guided tour from theory to praxis," in *Proseminar Datenkompression 2001*. RWTH Aachen University.

[28] "AMR speech codec; transcoding functions," 3GPP TS 26.090, 1999.

[29] A. P. Dempster, N. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.