Prediction Transform GMM Vector Quantization for Wideband LSFs

Miguel Arjona Ramírez

***Abstract*—Split vector quantizers are specialized over clusters defined by a low-order Gaussian Mixture Model (GMM). A prediction-based lower-triangular transform is adapted for the enhancement of vector quantization (VQ) in each cluster. This transform is generalized to be used in generic vector spaces, where component shifts are used instead of time shifts. Optimal quantizer banks are designed in minimum noise structures whose codebooks are used for the proposed Cartesian split, which improves their coding gain. A novel minimum noise structure is proposed for split VQ. This kind of split VQ is tested for line spectral frequency (LSF) quantization of wideband speech spectra, revealing a comparable average performance to the Karhunen-Loève transform at lower rates with reduced outlier generation and computational complexity.**

Index Terms prediction transform, vector quantization, Gaussian mixture models, line spectral frequencies, speech analysis, speech coding.

I. Introduction

VECTOR quantization (VQ) is more efficient than scalar quantization (SQ) but generally its search complexity is much higher and grows exponentially with dimension when full search is applied [1]. A successful approach factors the space into a Cartesian product of lower-dimensional subspaces in what is known as split VQ (SVQ). Another approach involves transform coding. Both approaches lead to lower computational complexity at a reduced performance penalty if properly applied. Indeed, for a broad range of applications, SVQ proper [2] or enhanced versions such as [3], [4] are good enough.

Linear transform coding of a vector source leads to a vector space where the components are less correlated. This makes quantization under weighted square distortion more efficient for jointly Gaussian sources. Eventually, if the source vectors can be rendered completely independent, the scalar quantization of the components of the transformed vector is very efficient and flexible [5], even though vector quantization still holds the space-filling advantage [6]. Such an optimal transform is the source-specific Karhunen-Loève transform (KLT) for jointly Gaussian sources.

By modeling an arbitrary source as a Gaussian mixture, each cluster can be viewed as a jointly Gaussian source. Thus the KLT can be considered optimal as long as each cluster is assigned its own KLT and the clusters are sufficiently far apart.

This work is supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under Grant no. 307633/2011-0 and by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) under Grant no. 2012/24789-0.

The current state-of-art transform domain quantization is GMM-based classified SVQ in the KLT domain for the line spectral frequency (LSF) representation of speech spectral parameters [7], building on previous results for GMM-based classified SVQ [8], [9].

However, there are other decorrelating transforms, which are less complex than the KLT to compute. One such transform is the prediction-based lower triangular transform (PLT) [10], which is investigated in this paper for SVQ at reduced complexity. By proper implementation, which involves the Cartesian SVQ proposed in Section V, its performance is better than scalar quantization while its gain matches that of the KLT.

II. Prediction transform for any space

The prediction-based lower triangular transform (PLT) [10] $\boldsymbol{B}$ transforms the $p \times 1$ zero-mean source vector $\boldsymbol{x}$ with covariance matrix $\boldsymbol{R}_{xx}$ into vector

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x} \tag{1}$$

with covariance matrix $\boldsymbol{R}_{yy}$, which is diagonal, where $p \times p$ matrix $\boldsymbol{B}$ is the lower triangular analysis matrix. Unlike the KLT, however, its diagonal entries are not its eigenvalues, but the residues or backward prediction error variances β_m of increasing order $m = 0, 1, \ldots, p-1$, where p is the dimension of the source vector space.

The PLT may be understood in a general linear prediction (LP) context where the vector space may be any so that the vectors need not be constrained to blocks of time delayed samples as assumed in its original proposal [10] and the component shift operator is used instead as is outlined in the Appendix. In this case the only constraints on the covariance matrix are its positive definiteness and its symmetry so that the proper LP method to be used is the covariance method [11] in contrast to the autocorrelation method suitable for stationary sample vectors.

The covariance matrix for multivariate vector $\boldsymbol{x}$ is defined as

$$\boldsymbol{R}_{xx} = E\left[\boldsymbol{x}\boldsymbol{x}^T\right], \tag{2}$$

where $E[\cdot]$ stands for the expected value with respect to the joint probability density function (pdf) of random vector $\boldsymbol{x}$. Since this pdf is not readily available, we use a training data matrix $\boldsymbol{\Xi}_c$ with N_c columns $\boldsymbol{\xi}$ of source vectors from a given Gaussian cluster c in order to estimate the $p \times p$ source covariance matrix $\boldsymbol{R}_{xx}$ with entries

$$r_{xx}(i,j) = \sum_{n=0}^{N_c-1} \xi(i,n)\xi(j,n) \tag{3}$$

for $i, j = 0, 1, \ldots, p-1$.

Analysis matrix $\boldsymbol{B}$ may be obtained by the upper-lower (UL) Cholesky factorization of $\boldsymbol{R}_{xx}^{-1}$ as

$$\boldsymbol{R}_{xx}^{-1} = \boldsymbol{B}^T \boldsymbol{R}_{yy}^{-1} \boldsymbol{B}. \tag{4}$$

where diagonal matrix $\boldsymbol{R}_{yy}$ has main diagonal entries β_m, for $m = 0, 1, \ldots, p-1$, which may be interpreted as backward prediction error variances according to the derivation given in the Appendix.

Alternatively, we may be interested in obtaining the inverse transfom matrix $\boldsymbol{S} = \boldsymbol{B}^{-1}$ directly for the implementation of a minimum noise structure as outlined below. In this case, we carry out a lower-upper (LU) Cholesky factorization of $\boldsymbol{R}_{xx}$ as

$$\boldsymbol{R}_{xx} = \boldsymbol{S} \boldsymbol{R}_{yy} \boldsymbol{S}^T. \tag{5}$$

The PLT is not a unitary transform so that its inverse is not its transpose. But it attains the same gain as the KLT as long as it is implemented in a minimum noise structure. Two such structures have been proposed, MINLAB(I) and MINLAB(II) [10]. We will use the former, which turns out to be less complex when the sequence of vectors is much longer than the dimension.

In order to implement MINLAB(I), the inverse transform matrix $\boldsymbol{S}$ must be derived and then factored as

$$\boldsymbol{S} = \boldsymbol{S}_1 \cdot \boldsymbol{S}_2 \cdots \boldsymbol{S}_{p-1}, \tag{6}$$

where $\boldsymbol{S}_m$ takes its mth row from $\boldsymbol{S}$ and the remaining rows from the identity matrix.

Next, the transform matrix may be recovered by inverting Eq. (6) as

$$\boldsymbol{B} = \boldsymbol{S}_{p-1}^{-1} \cdot \boldsymbol{S}_{p-2}^{-1} \cdots \boldsymbol{S}_1^{-1}, \tag{7}$$

where the matrix inverses are quite straightforward to obtain since row m in S_m^{-1} is obtained as

$$\left[S_m^{-1}\right]_m = \left[\begin{array}{ccccccc} -s_{m0} & \cdots & -s_{m,m-1} & s_{mm} & \cdots & 0 \end{array} \right] \tag{8}$$

from row m in S_m whereas the remaining rows are just repeated so that diagonal entries s_{mm} are unity and do not change sign upon inversion.

III. Gaussian mixture model clustering

Split vector quantization is to be performed over subvectors so that each split quantizer becomes isolated from the other subvectors in the vector to be quantized, thereby causing a split loss [12]. In order to enhance the overall performance of the quantizer, a joint GMM-SVQ system is used.

The whole training source vectors are used for modeling their joint probability density function $f_{\boldsymbol{X}}(\boldsymbol{x})$ by a Gaussian mixture model

$$f_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{i=1}^{M} c_i f_i(\boldsymbol{x}|\boldsymbol{\theta_i}), \tag{9}$$

where M is the number of Gaussian components or clusters in the mixture and

$$\boldsymbol{\theta} = \left\{ \begin{array}{ccccccccc} c_1, & c_2, & \ldots, & c_M, & \boldsymbol{\theta}_1, & \boldsymbol{\theta}_2, & \ldots, & \boldsymbol{\theta}_M \end{array} \right\}$$

are the mixture parameters with c_i and $\boldsymbol{\theta}_i = \left\{ \begin{array}{cc} \boldsymbol{\mu}_i, & \boldsymbol{R}_i \end{array} \right\}$ for $i = 1, 2, \ldots, M$ being the a priori cluster probabilities and the parameters for each Gaussian component, which are its mean vector $\boldsymbol{\mu}_i$ and its covariance matrix $\boldsymbol{R}_i$, so that the component pdfs are

$$f_i(\boldsymbol{x}|\boldsymbol{\theta}_i) = \frac{1}{(2\pi)^{p/2}\sqrt{\det(\boldsymbol{R}_i)}} \times \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \boldsymbol{R}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right] \tag{10}$$

for $i = 1, 2, \ldots, M$. The number of clusters M could be a model parameter [5] but we have chosen to fix it at $M = 8$ in order to keep the computational complexity manageable while the model is still verified to be efficient.

The model parameters are estimated over the training database, described in Section VII, that consists of a $p \times N$ data matrix $\boldsymbol{\Xi}$ holding N p-dimensional LSF coefficients. At first, they are sequentially segmented into M equiprobable clusters, that is, initial a priori probabilities are $c_i = 1/M$ for $i = 1, 2, \ldots, M$ and the mean vectors and the covariance matrices are estimated over each cluster, thereby defining the initial model. Then the expectation-maximization (EM) algorithm [13] is run in iterations consisting of two steps

1) *Expectation:* For each training vector $\boldsymbol{\xi}(n)$, the a posteriori probability that it was generated by component m in the mixture is computed as

$$\chi_m(n) = \frac{c_m f_m(\boldsymbol{\xi}(n)|\boldsymbol{\theta_m})}{\sum_{i=1}^{M} c_i f_i(\boldsymbol{\xi}(n)|\boldsymbol{\theta_i})} \tag{11}$$

for $n = 0, 1, \ldots, N-1$ and $m = 1, 2, \ldots, M$.

2) *Maximization:* The likelihood is maximized by reevaluating for each cluster

$$c_m = \frac{1}{N}\sum_{n=0}^{N-1} \chi_m(n) \tag{12}$$

$$\boldsymbol{\mu}_m = \frac{\sum_{n=0}^{N-1} \chi_m(n)\boldsymbol{\xi}(n)}{\sum_{n=0}^{N-1} \chi_m(n)} \tag{13}$$

$$\boldsymbol{R}_m = \sum_{n=0}^{N-1} \chi_m(n)(\boldsymbol{\xi}(n)-\boldsymbol{\mu}_m)(\boldsymbol{\xi}(n)-\boldsymbol{\mu}_m)^T \times \left(\sum_{n=0}^{N-1} \chi_m(n)\right)^{-1} \tag{14}$$

for $m = 1, 2, \ldots, M$.

With the mixture model estimated, the training vectors are assigned to the cluster whose component pdf provides the maximum likelihood, that is,

$$\hat{m}(\boldsymbol{\xi}(n)) = \underset{i \in \{1,2,\ldots,M\}}{\arg\max}\, f_i(\boldsymbol{\xi}(n)|\boldsymbol{\theta_i}). \tag{15}$$

These clusters are referred to as Gaussian clusters.

IV. Prediction transform and scalar quantization

Ideally, the transform should remove the correlation in the vector to be coded and leave the complementary modeling of

the probability density function (pdf) to the scalar quantizer, which is aided in this task by the GMM clustering described in Section III.

The scalar quantizers are just inserted in cascade between the analysis and synthesis filterbanks when a unitary transform is used. However, for the minimum noise PLT the scalar quantizer bank must be interleaved with the factored implementation of the analysis filterbank. This may be represented by means of diagonal functional operator matrices $\boldsymbol{Q}_m(\cdot)$ with diagonal

$$\left[\begin{array}{ccccccc} 1 & \dots & 1 & q_m(\cdot) & 1 & \dots & 1 \end{array}\right]$$

for $m = 0, 1, \dots, p-1$, with scalar quantizer $q_m(\cdot)$ at the mth column. This allows us to represent the MINLAB(I) implementation of the encoder as

$$\tilde{\boldsymbol{y}} = \boldsymbol{Q}_{p-1}\left(\boldsymbol{S}_{p-1}^{-1} \cdot \boldsymbol{Q}_{p-2}\left(\boldsymbol{S}_{p-2}^{-1} \cdots \boldsymbol{Q}_1\left(\boldsymbol{S}_1^{-1}\boldsymbol{Q}_0\left(\boldsymbol{x}\right)\right)\cdots\right)\right), \tag{16}$$

where $\tilde{\boldsymbol{y}}$ is the quantized transformed vector. Therefore, in this implementation, transforming and quantizing are interleaved.

Furthermore, it is interesting to remark that this algorithm implements subband noise feedback from lower-frequency bands.

Conversely, the inverse transform

$$\tilde{\boldsymbol{x}} = \boldsymbol{S}\tilde{\boldsymbol{y}} \tag{17}$$

for decoding may be implemented by using either the ladder decomposition in Eq. (6) or, equivalently, by using matrix $\boldsymbol{S}$ directly.

Since the subband signals are uncorrelated after transforming, optimal rate allocation among scalar quantizers is determined by the prediction error variances β_m as

$$R_m = \frac{R}{p} - \frac{\log_2 M}{p} + \frac{1}{2}\log_2 \frac{\beta_m}{\prod_{l=0}^{p-1}\beta_l}, \tag{18}$$

where R is the bit rate per vector, R/p is the average bit rate per sample, $\log_2 M$ is the bit rate per vector for GMM cluster selection and R_m for $m = 0, 1, \dots, p-1$ are the bit rates per sample for each subband.

V. Prediction transform and vector quantization

In principle, scalar quantization is optimal when the transform generates vectors with independent components before quantization as pointed out in Section IV for the KLT and the PLT. However, the dependence may not be completely removed due to estimation errors and nonlinear dependence [7]. The latter is significant in many practical situations so that VQ increases the coding gain as shown in Secion VII.

In resolution-constrained quantization, linear and nonlinear dependences are measured by the memory advantage of VQ over SQ [6], [7], which for square distortion and p-dimensional vectors is defined as

$$M(p,2) = \left(\frac{\int \prod_{m=0}^{k-1} f_{X_m}^{p/(p+2)}(x_m)dx_m}{\int\ f_{\boldsymbol{X}}^{p/(p+2)}(\boldsymbol{x})d\boldsymbol{x}}\right)^{(p+2)/p} \tag{19}$$

where $f_{\boldsymbol{X}}(\boldsymbol{x})$ is the joint pdf and $f_{X_m}(x_m)$ for $m = 0, 1, \dots, p-1$ are the marginal pdfs. Assuming a jointly Gaussian vector, the memory advantage is

$$M_G(p,2) = \left(\frac{\prod_{m=0}^{p-1} r_{xx}(m,m)}{\prod_{m=0}^{p-1}\beta_m}\right)^{1/p}. \tag{20}$$

The PLT actually provides a measure of nonlinear dependence by β_{p-1}, the prediction error variance of highest order, which quantifies how much of the variance goes unexplained by *linear* prediction. When it equals zero, all dependence is linear; otherwise, $\beta_{p-1} > 0$ indicates remaining nonlinear dependence between vector coordinates. The absence of any significant independent interferer is assumed.

Split quantization is proposed to take partial advantage of eventual nonlinear dependence. However, quantization noise is harder to compensate at the split level than by a scalar interleaved structure. Fortunately, an interesting association exists between the noise minimization provided by scalar quantization and the encoding benefits of vector quantization. It is achieved by Cartesian SVQ (CSVQ) as described below.

Scalar codebooks $\boldsymbol{\mathcal{C}}_m$ are designed for each dimension $m = m_{0i}, m_{0i}+1, \dots, m_i$ in split i using a MINLAB(I) structure. For SVQ, the codebook for split i is obtained by the Cartesian product

$$\boldsymbol{\mathcal{D}}_i = \boldsymbol{\mathcal{C}}_{m_{0i}} \times \boldsymbol{\mathcal{C}}_{m_{0i}+1} \times \cdots \times \boldsymbol{\mathcal{C}}_{m_i}. \tag{21}$$

In fact, the Cartesian codebook structure enables the nonlinear memory advantage of VQ over SQ to be used while enforcing the minimum noise condition as shown by the results in Section VII. Its analysis is less complex due to the lower triangular structure of the analysis matrix, which can be easily factored.

VI. Complexity

The operational complexity for a transform quantizer may be broken down into the following pieces: mean subtraction (MSUB), analysis filterbank (ANAFB), quantization (Q), distortion calculation (DIST), synthesis filterbank (SYNFB), mean addition (MADD) and final vector comparison (FVECC). The dependence of these complexity components upon number of clusters and vector dimension is given in Table I but for the quantization component.

TABLE I
COMPLEXITY BREAKDOWN OF TRANSFORM PROCESSING FOR QUANTIZATION OF p-DIMENSIONAL GAUSSIAN-MIXTURE MODELED VECTORS, CLASSIFIED INTO M CLUSTERS.

Operation	Transform complexity (flop/frame) PLT	(flop/frame) KLT
MSUB	Mp	Mp
ANAFB	$Mp(p-1)$	$Mp(2p-1)$
DIST	$M(4p-1)$	$M(4p-1)$
SYNFB	$Mp(p-1)$	$Mp(2p-1)$
FVECC	M	M

Quantization may be implemented as block scalar quantization (BSQ) or vector quantization (VQ). We have considered

mainly nonuniform BSQ with binary search, whose computational complexity is

$$\begin{aligned} C(\mathrm{Q},\mathrm{BSQ}) &= \sum_{i=1}^{M}\sum_{j=1}^{p} R_{ij} \\ &= M\left(R-\log_2 M\right), \end{aligned} \tag{22}$$

where R_{ij} is the rate for component j in cluster i, R is the total bit rate per vector and $\log_2 M$ is the bit rate per vector for GMM cluster selection.

Using the partial complexities in Table I, the total computational complexity for PLT BSQ is

$$C_{\mathrm{tot}}(PLT,BSQ) = 2Mp^2+3Mp+M\left(R-\log_2 M\right) \tag{23}$$

and the total computational complexity for KLT BSQ is

$$C_{\mathrm{tot}}(KLT,BSQ) = 4Mp^2+3Mp+M\left(R-\log_2 M\right) \tag{24}$$

so that the total complexity for scalar PLT is about half that of scalar KLT. Specifically, in the range of situations tested in Section VII, this ratio is around 54%.

For VQ ANAFB, complexity will have to be distributed over splits and clusters, leading to

$$C(\text{ANAFB \& Q},\mathrm{SVQ}) = \sum_{i=1}^{M}\sum_{j=1}^{\varsigma}\left(4p_j-1\right)2^{R_{ij}}, \tag{25}$$

where R_{ij} is the rate for split j in cluster i, ς is the number of splits and M is the number of clusters. Therefore, in order to evaluate this complexity component the dimension split and the bit allocation per split are necessary. This is exemplified in Section VII.

VII. Experimental results

The transform quantization methods discussed and proposed have been applied to sequences of line spectral frequency (LSF) vectors extracted from wideband speech signals. The adaptive multirate wideband (AMR-WB) [14] coder has been used to compute LSF vectors at a rate of 50 Hz for the signals in the TIMIT database [15], whose training partition with 705,580 vectors has been used for training the quantizers while its test partition with 257,852 vectors has been assigned for testing. For the simulations, MATLAB has been used.

For the training set of LSF vectors, the mean vector is evaluated and then subtracted from each vector, thereby obtaining centered vectors. Spectral weighting coefficients are computed from the sensitivity matrix of each LSF vector under high-rate approximation [16] and the bit rate is optimally allocated to scalar quantizers according to prediction residue variances for the PLT and eigenvalues for the KLT with rounding and adjustment. For vector quantizers, the allocations are cumulated over each split. An exceptional allocation is made for the pure split vector quantizer in Table II, considered as a reference.

Performance is measured according to the criteria set forth by Paliwal and Atal for transparent quantization [2]:

- The average spectral distortion (SD) is about 1 dB.
- There is no outlier frame with SD above 4 dB.
- The ratio of outlier frames in the range from 2 dB to 4 dB is less than 2%.

The best reference for scalar transform quantization is the KLT scalar quantizer (KLT SQ), implemented with bit allocation based on Eq. (18), whose performance is shown in Table III, and can be seen to outperform SVQ in mean spectral distortion by more than 0.15 dB and by a lower number of outliers in the 2 dB to 4 dB range, even though it is slightly inferior in outlier performance above 4 dB.

Now the stage is set for evaluating the performance of PLT scalar quantization (PLT SQ), displayed in Table IV, which is found to outperform KLT SQ at 45 bit/fr and 46 bit/fr and by following rather close the performance of KLT SQ at lower rates and consistently exceeding it in outlier performance above 4 dB. Further, when the lower complexity of PLT SQ is taken into consideration, it sounds like a better option for transform SQ.

For training the transform vector quantizers, the training vectors are first clustered into eight classes through a Gaussian Mixture Model (GMM) and then a vector quantizer is designed for each cluster. The Linde-Buzo-Gray (LBG) algorithm [17] is used initializing with a single codevector at the centroid and doubling the number of codevectors in centroid splitting steps. For testing, each test vector is quantized with the vector quantizer for each Gaussian cluster and the lowest distortion result is selected.

Using the procedure outlined above, the performance of KLT SVQ is found to improve significantly over the scalar quantization version as shown in Table V, particularly in outlier performance in both ranges.

Finally, PLT Cartesian SVQ (PLT CSVQ) has a gain in performance over its scalar version as can be seen from the results in Table VI. This is to be expected since speech spectral parameters are known to have significant nonlinear dependence [1] as discussed in Section V. It is most noticeable that outliers are greatly reduced either over the scalar version performance as over the KLT SVQ performance. Still the average distortions are somewhat higher for PLT CSVQ but they may be traded off for the significantly lower operational complexity incurred by PLT CSVQ as compared with KLT SVQ, shown in Table VII to be around 3/4 as much.

TABLE II
PERFORMANCE OF STANDARD SPLIT VECTOR QUANTIZATION FOR 16-DIMENSIONAL LSF VECTORS IN (3,3,3,3,4)-DIMENSIONAL SPLITS, INCLUDING MEAN LOG SPECTRAL DISTORTION AND TWO CLASSES OF OUTLIERS.

Bit rate		Mean	Outliers	
Per frame (bit/frame)	Per split (bit/split)	SD (dB)	2 – 4 dB (%)	$>$ 4 dB (ppm)
41	(8,9,8,8,8)	1.124	1.43	4
42	(8,9,9,8,8)	1.079	1.05	0
43	(8,9,9,9,8)	1.043	0.78	4
44	(9,9,9,9,8)	1.015	0.72	4
45	(9,9,9,9,9)	0.957	0.38	0
46	(9,10,9,9,9)	0.919	0.29	0

TABLE III
PERFORMANCE OF KLT SCALAR QUANTIZATION FOR 16-DIMENSIONAL LSF VECTORS, INCLUDING MEAN LOG SPECTRAL DISTORTION AND TWO CLASSES OF OUTLIERS.

Bit rate	Mean	Outliers	
Per frame (bit/frame)	SD (dB)	2 – 4 dB (%)	>4 dB (ppm)
41	0.981	1.14	19
42	0.950	0.91	16
43	0.916	0.72	12
44	0.882	0.55	12
45	0.855	0.49	12
46	0.824	0.36	12

TABLE IV
PERFORMANCE OF PLT SCALAR QUANTIZATION FOR 16-DIMENSIONAL LSF VECTORS, INCLUDING MEAN LOG SPECTRAL DISTORTION AND TWO CLASSES OF OUTLIERS.

Bit rate	Mean	Outliers	
Per frame (bit/frame)	SD (dB)	2 – 4 dB (%)	>4 dB (ppm)
41	1.015	1.39	4
42	0.974	1.03	4
43	0.926	0.69	4
44	0.893	0.56	0
45	0.854	0.42	0
46	0.815	0.32	0

TABLE V
PERFORMANCE OF KLT SVQ FOR 16-DIMENSIONAL LSF VECTORS IN (2,2,2,2,4,4)-DIMENSIONAL SPLITS, INCLUDING MEAN LOG SPECTRAL DISTORTION AND TWO CLASSES OF OUTLIERS.

Bit rate	Mean	Outliers	
Per frame (bit/frame)	SD (dB)	2 – 4 dB (%)	>4 dB (ppm)
41	0.920	0.58	4
42	0.888	0.42	4
43	0.854	0.34	4
44	0.818	0.25	4
45	0.782	0.18	4
46	0.753	0.14	4

TABLE VI
PERFORMANCE OF PLT CARTESIAN SVQ FOR 16-DIMENSIONAL LSF VECTORS IN (2,3,3,3,3,2)-DIMENSIONAL SPLITS, INCLUDING MEAN LOG SPECTRAL DISTORTION AND TWO CLASSES OF OUTLIERS.

Bit rate	Mean	Outliers	
Per frame (bit/frame)	SD (dB)	2 – 4 dB (%)	>4 dB (ppm)
41	0.963	0.65	4
42	0.942	0.61	0
43	0.906	0.46	0
44	0.861	0.34	0
45	0.835	0.26	0
46	0.804	0.21	0

VIII. CONCLUSION

The prediction transform has been proposed for a transform quantizer designed over clusters determined by a low-order Gaussian mixture model which improves the performance of split VQ by reducing its split loss. The transform matrices have been derived by the covariance method of linear prediction for general vector spaces using component shifts in contrast to the original proposal of the PLT for time shifts. A scalar quantizer has been proposed in a minimum noise structure with interleaved analysis and quantization, whose computational complexity is almost half that of KLT scalar quantization. The coding gain has been enhanced by using VQ, which in a novel PLT Cartesian SVQ comes close to KLT SVQ average performance at low rates with improved outlier performance and complexity of almost 3/4 that of KLT SVQ. This has been achieved because of the novel minimum noise structure for split VQ.

APPENDIX
GRAM-SCHMIDT ORTHOGONALIZATION FOR COMPONENTWISE PREDICTION

Relations between component random variables in a multivariate vector may be expressed by means of the shift operator, which may be represented by the lower shift matrix

$$\boldsymbol{Z} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{26}$$

and its powers, or, alternatively, by the polynomial z^{-1} and its powers. The latter allows an interpretation of the transforming operations in the context of linear prediction, establishing a correspondence between entries in rows of the analysis matrix and coefficients in the backward prediction error polynomial of the same order as the row index.

The covariance matrix $\boldsymbol{R}_{xx}$ induces an inner product for a polynomial space associated to the vector space under analysis. Writing these polynomials in the variable z^{-1}, consider two such polynomials $P(z) = \sum_{i=1}^{K} c_i z^{-i}$ and $Q(z) = \sum_{j=1}^{L} d_l z^{-j}$. Due to the distributive property of inner products over vector addition, the inner product of these polynomials may be expanded as

$$\langle P(z), Q(z) \rangle = \sum_{i=1}^{K} \sum_{j=1}^{L} c_i d_j \left\langle z^{-i}, z^{-j} \right\rangle \tag{27}$$

so that the inner product is completely defined by the products of monomials

$$\left\langle z^{-i}, z^{-j} \right\rangle = r_{xx}(i,j) \tag{28}$$

for $i, j = 0, 1, \ldots, p-1$ as long as our interest is restricted to polynomials of degrees $K \leq p$ and $L \leq p$. These monomials are shift operators for the vector coordinates and should not be identified with time delays, which they may even be just as a special case. It should be observed that this is a valid definition for an inner product because matrix $\boldsymbol{R}_{xx}$ is positive definite and symmetric.

Given a $p \times p$ covariance matrix $\boldsymbol{R}_{xx}$ defining the inner product in the vector space of polynomials with degree less than or equal to p in the variable z^{-1} and the canoni-

TABLE VII
OPERATIONAL COMPLEXITY OF PLT CARTESIAN SVQ FOR 16-DIMENSIONAL LSF VECTORS IN (2,3,3,3,3,2)-DIMENSIONAL SPLITS COMPARED TO THAT OF KLT SVQ FOR A (2,2,2,2,4,4)-DIMENSIONAL PARTITION.

Bit rate	PLT CSVQ	KLT SVQ	Ratio
Per frame (bit/frame)	Complexity (kflop/frame)	Complexity (kflop/frame)	PLT to KLT (%)
41	63	81	78
42	70	86	81
43	74	90	82
44	75	106	71
45	82	117	70
46	86	135	64

cal basis $\left\{z^{-i}\right\}_{i=1}^{p}$, we want to obtain an orthogonal basis $\{B_m(z)\}_{m=0}^{p-1}$.

As mentioned at the beginning of this section, the operator z^{-1} used in this paper does not stand for a time delay but for a one-place shift down in vector component and the polynomials are not interpreted as transfer functions for finite-length impulse response filters. Otherwise, we follow a standard Gram-Schmidt procedure [11], which is outlined here for the sake of completeness. It starts with the first basis vector assignment

$$B_0(z) = z^{-1}. \tag{29}$$

Then, by finite induction, we assume that we have the orthogonal basis $\{B_l(z)\}_{l=0}^{m-1}$ determined after having orthogonalized the basis vectors up to the basis $\left\{z^{-i}\right\}_{i=1}^{m}$. So we proceed by including the next canonical basis vector $z^{-(m+1)}$ to find the next orthogonalized basis vector from

$$B_m(z) = z^{-(m+1)} - \sum_{i=0}^{m-1} \gamma_{mi} B_i(z), \tag{30}$$

where the projection coefficients γ_{mi}, for $i = 0, 1, \ldots, m-1$, are found by imposing the set of orthogonality constraints

$$\langle B_m(z), B_i(z) \rangle = 0 \tag{31}$$

for $i = 0, 1, \ldots, m-1$, which, upon substitution of Eq. (30) and simplification based on the orthogonality between the new basis vectors reduce to

$$\left\langle z^{-(m+1)}, B_i(z) \right\rangle - \gamma_{mi}\beta_i = 0 \tag{32}$$

for $i = 0, 1, \ldots, m-1$, where the square norms of the basis vectors are

$$\beta_i = \langle B_i(z), B_i(z) \rangle \tag{33}$$

for $i = 0, 1, \ldots, m-1$. Solving the set of equations (32), the projection coefficients are found to be

$$\gamma_{mi} = \frac{\left\langle z^{-(m+1)}, B_i(z) \right\rangle}{\beta_i} \tag{34}$$

for $i = 0, 1, \ldots, m-1$. Now, replacing the projection coefficients in Eq. (30) by their expressions in Eq. (34), the coefficients for the new orthogonal basis vector

$$B_m(z) = z^{-(m+1)} + \sum_{j=1}^{m-1} b_{mj} z^{-(j+1)} \tag{35}$$

are found to be

$$b_{mj} = -\sum_{l=j}^{m-1} \gamma_{ml} b_{lj}. \tag{36}$$

Finally, the square norm

$$\beta_m = \langle B_m(z), B_m(z) \rangle \tag{37}$$

is computed by replacing the first factor by its expression in Eq. (30) and using the orthogonal relations between the orthogonal basis vectors to obtain

$$\begin{aligned} \beta_m &= \left\langle z^{-(m+1)} - \sum_{i=0}^{m-1} \gamma_{mi} B_i(z), B_m(z) \right\rangle \\ &= \left\langle z^{-(m+1)}, B_m(z) \right\rangle, \end{aligned} \tag{38}$$

where the new basis vector is replaced by its expression in Eq. (35) to obtain the computable expression

$$\beta_m = r_{xx}(m+1, m+1) + \sum_{j=0}^{m-1} b_{mj} r_{xx}(m+1, j+1). \tag{39}$$

So orthogonalization is complete up to the $(m+1)$th basis vector and may be carried on to include the next one up to the pth basis vector.

Finally, the set of coefficients b_{mj}, for $j = 0, 1, \ldots, m$ and $m = 0, 1, \ldots, p-1$, is used to populate the lower triangle in analysis matrix $\boldsymbol{B}$ whereas the backward prediction error variances β_m define the diagonal entries of covariance matrix $\boldsymbol{R}_{yy}$ for $m = 0, 1, \ldots, p-1$.

REFERENCES

[1] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1457608.

[2] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Speech Audio Processing*, vol. 1, no. 1, pp. 3–14, Jan. 1993, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=221363.

[3] M. Arjona Ramírez, "Optimized subvector processing in split vector quantization," *Journal of Communication and Information Systems*, vol. 25, pp. 20–24, Apr. 2010, http://iecom.dee.ufcg.edu.br/~jcis/Abril%202010/volume25/JCIS_2010_25_1_003.pdf.

[4] M. Arjona Ramírez, "Vector quantization with renormalized splits for wideband speech," in *Proc. of DSP 2011 17th International Conference on Digital Signal Processing*, Corfu, Greece, 2011, pp. 1–4, http://dx.doi.org/10.1109/ICDSP.2011.6004911.

[5] A. D. Subramaniam and B. D. Rao, "PDF optimized parametric vector quantization of speech line spectral frequencies," *IEEE Trans. Speech Audio Processing*, vol. 11, no. 2, pp. 130–142, Mar. 2003, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1193579.

[6] T. D. Lookabaugh and R. M. Gray, "High-resolution quantization theory and the vector quantizer advantage," *IEEE Trans. Inform. Theory*, vol. 35, no. 5, pp. 1020–1033, Sept. 1989, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=42217.

[7] Y. Lee, W. Jung, and M. Y. Kim, "GMM-based KLT-domain switched-split vector quantization for LSF coding," *IEEE Signal Processing Lett.*, vol. 18, pp. 415–418, July 2011, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5766714.

[8] S. Chatterjee and T. V. Sreenivas, "Gaussian mixture model based switched split vector quantization of LSF parameters," in *Proc. IEEE Int. Symp. Signal Process. Inf. Tech.*, Cairo, Egypt, 2007, pp. 1054–1059, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4458124.

[9] ——, "Analysis-by-synthesis based switched transform domain split VQ using Gaussian mixture model," in *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tapei, Taiwan, 2009, pp. 4117–4120, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4960534.

[10] S.-M. Phoong and Y.-P. Lin, "Prediction-based lower triangular transform," *IEEE Trans. Signal Processing*, vol. 48, no. 7, pp. 1947–1955, July 2000, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=847781.

[11] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. Berlin: Springer, 1976.

[12] F. Nordén and T. Eriksson, "On split quantization of LSF parameters," in *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, Montreal, Canada, 2004, pp. 157–160, http://dx.doi.org/10.1109/ICASSP.2004.1325946.

[13] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, Apr. 1984, http://dx.doi.org/10.1137/1026034.

[14] B. Bessette, R. Salami, R. Lefebvre, M. Jelínek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Järvinen, "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 8, pp. 620–636, Nov. 2002, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1175533.

[15] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus," Linguistic Data Consortium, 1993, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1.

[16] W. R. Gardner and B. D. Rao, "Theoretical analysis of the high-rate vector quantization of LPC parameters," *IEEE Trans. Speech Audio Processing*, vol. 3, no. 5, pp. 367–381, Sept. 1995, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=466658.

[17] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84–95, Jan. 1980, http://dx.doi.org/10.1109/TCOM.1980.1094577.

Miguel Arjona Ramírez is Associate Professor at Escola Politécnica, University of São Paulo, where he is a member of the Signal Processing Laboratory. He received the E.E. degree from Instituto Tecnológico de Aeronáutica, Brazil, and the M.S., the Dr. and the Habilitation (Livre-Docência) degrees in Electrical Engineering from University of São Paulo, Brazil, in 1992, 1997 and 2006, respectively, and the Electronic Design Eng. degree from Philips International Institute, The Netherlands, in 1981. In 2008 he carried post-doctoral research in time-frequency speech analysis and coding at the Royal Institute of Technology in Stockholm, Sweden. He was Engineering Development Group Leader for Interactive Voice Response Systems (IVRs) for Itautec Informática, Brazil, where he served from 1982 to 1990. He is a Senior Member of the IEEE since 2000 and a Member of the Brazilian Telecommunications Society (SBrT). His research interests include signal compression, speech analysis, coding and recognition, speaker identification and audio analysis and coding.