16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

LOW COMPLEXITY WIDEBAND LSF QUANTIZATION USING GMM OF UNCORRELATED GAUSSIAN MIXTURES

Saikat Chatterjee and T.V. Sreenivas

Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore, India
saikat@ece.iisc.ernet.in, tvsree@ece.iisc.ernet.in

ABSTRACT

We develop a Gaussian mixture model (GMM) based vector quantization (VQ) method for coding wideband speech line spectrum frequency (LSF) parameters at low complexity. The PDF of LSF source vector is modeled using the Gaussian mixture (GM) density with higher number of uncorrelated Gaussian mixtures and an optimum scalar quantizer (SQ) is designed for each Gaussian mixture. The reduction of quantization complexity is achieved using the relevant subset of available optimum SQs. For an input vector, the subset of quantizers is chosen using nearest neighbor criteria. The developed method is compared with the recent VQ methods and shown to provide high quality rate-distortion (R/D) performance at lower complexity. In addition, the developed method also provides the advantages of bitrate scalability and rate-independent complexity.

1. INTRODUCTION

Low complexity, but high quality vector quantization (VQ) of line spectrum frequency (LSF) parameters has attracted much attention in the current literature [6], [7], [9], [11], [12], [14], [15]. The complexity issues in VQ are more important to address for the applications where the requirement of higher perceptual quality is achieved through the allocation of higher bitrate; in these applications, such as in wideband speech coding, the complexity of LSF VQ is very high and thus, it is important to keep the complexity under check without sacrificing the rate-distortion (R/D) performance. One of the most cited low complexity VQ schemes is split VQ (SVQ) which was first proposed by Paliwal and Atal for telephone-band speech LSF coding [2] and then extended to wideband speech LSF coding [5]. So and Paliwal have recently proposed switched SVQ (SSVQ) method [14] [9] which is shown to provide better R/D performance than SVQ at lower computational complexity, but at the requirement of higher memory. We also have proposed two stage VQ methods [12], [15] which are shown to provide comparable R/D performance of SSVQ, but at the requirement of much lower complexities (both computational and memory).

In addition to addressing the trade-off between complexity and R/D performance, current VQ schemes also focus on two important issues: seamless bitrate scalability and rate-independent complexity. The enormous complexity, required for training the optimum codebooks at different bitrates, hinders the conventional VQ framework to allow seamless bitrate scalability according to the user requirement and channel condition. At most, it may be possible to store the optimum codebooks for different bitrates, but at the cost of prohibitive memory requirement. Also, the computational complexity increases exponentially as more bits are used in a scalable coder for achieving better quality according to the user requirement and thus, it is important to design a VQ scheme with rate-independent complexity. For telephone-band speech LSF coding, both of the issues (bitrate scalability and rate-independent complexity) are addressed by Subramaniam and Rao [6] using the Gaussian mixture model (GMM) based PDF optimized parametric VQ (GMVQ) method. The technique of GMVQ method is further used for matrix quantization of telephone-band speech [10] and wideband speech [8] LSF parameters by So and Paliwal. In GMVQ [6], typically 8 or 16 number of Gaussian mixtures are used for LSF vector source modeling and an optimum transform domain scalar quantizer (TrSQ) is designed for each Gaussian mixture where the mixture specific KLT is used along-with the optimum bit allocation to transform domain scalar components. An input LSF vector is quantized using all the TrSQs and then the best performing TrSQ is selected as the winning quantizer using the computationally intensive spectral distortion (SD) measure based post-processing (SD measure is the perceptually relevant objective measure used for evaluating the LSF quantization performance). The post-processing is nothing but a VQ method that uses SD as the distortion measure to choose the best quantized vector from the set of quantized vectors produced by all the TrSQs. It can be noted that the increase in number of Gaussian mixtures for better modeling of the source PDF using GMM leads to higher complexity in GMVQ.

In this paper, we show that the GMVQ method of [6] can be modified to yield high quality quantization performance even at the requirement of much lower complexity. In the modified method, we model the PDF of LSF vector using a GMM with higher number of uncorrelated Gaussian mixtures and thus, the quantization method is referred to as UGMVQ. The developed UGMVQ method is shown to provide comparable R/D performance of SSVQ at the requirement of lower complexity and also retaining the advantages of bitrate scalability and rate-independent complexity. In this paper, we choose the SSVQ as the baseline method for comparison with UGMVQ because it is shown in a recent comparative study [13] of several wideband LSF coding methods (including GMVQ of [6]) that the SSVQ provides transparent quality wideband LSF quantization performance at a better trade-off between all the VQ performance measures.

2. GMVQ WITH UNCORRELATED GAUSSIAN MIXTURES (UGMVQ)

The GMVQ method of [6] is modified to develop the UGMVQ method for achieving low complexity. The modifications are as follows:

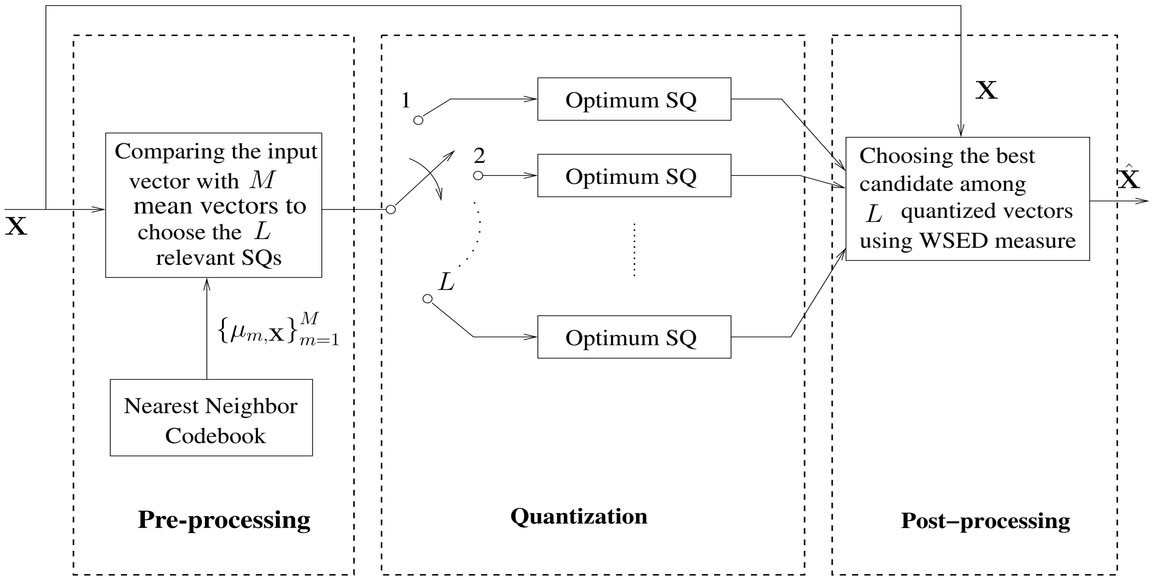


Figure 1: UGMVQ method: GMVQ method where the GM density is modeled using uncorrelated Gaussian components.

(1) Using GMM with higher number of uncorrelated Gaussian mixtures and thus, using optimum scalar quantizer (SQ) for each Gaussian mixture instead of TrSQ.
(2) Incorporating a pre-processing block for choosing the relevant subset of optimum SQs from the set of all optimum SQs. For an input vector, the subset of optimum SQs is chosen adaptively by rank ordering using the nearest neighbor criteria.
(3) Post-processing using computationally simple weighted square Euclidean distance (WSED) measure which is shown in [3] as an approximation to SD measure. We use the spectral sensitivity coefficients [3] as the weighting values to compute WSED measure.

2.1 Source vector PDF modeling

Let, $\mathbf{X}$ be the p-dimensional LSF source vector; the PDF of $\mathbf{X}$ is modeled using a GM density of M Gaussian mixtures as,

$$f_{\mathbf{X}}(\mathbf{x}) \approx \sum_{m=1}^{M} \alpha_m \, \mathrm{N}(\mu_{m,\mathbf{X}}, C_{m,\mathbf{X}}) \tag{1}$$

where α_m, $\mu_{m,\mathbf{X}}$ and $C_{m,\mathbf{X}}$ are the prior probability, mean vector and covariance matrix of the mth Gaussian component, $\mathrm{N}(\mu_{m,\mathbf{X}}, C_{m,\mathbf{X}})$. The approximate equality, used in Eqn. 1, is because of modeling a source PDF using the GMM with finite number of M mixtures. However M is chosen large so that each mixture density has small covariance and the covariance matrix is also assumed to be diagonal, i.e. $C_{m,\mathbf{X}} = \mathrm{diag}\left[\left\{\sigma_{m,1}^2, \sigma_{m,2}^2, \ldots, \sigma_{m,p}^2\right\}\right]$, $1 \leq m \leq M$; here $\sigma_{m,i}^2$ is the variance of ith scalar component for mth Gaussian mixture. The well-known expectation-maximization (EM) algorithm is used to evaluate the GMM parameters. The use of large number of Gaussian mixtures allows us to capture the details of source PDF and thus, better quantization performance is guaranteed.

2.2 Quantization algorithm

For each Gaussian mixture, we resort to direct independent SQ instead of transform domain SQ (TrSQ) because of modeling the source PDF using GMM of uncorrelated Gaussian components. However, the higher complexity due to increased M in GM density modeling is kept under check through the use of subset of SQs from the set of all M SQs. The use of subset of SQs, instead of all M SQs, is experimentally shown to affect the LSF quantization performance minimally. For each Gaussian mixture specific SQ, optimum bit allocation is carried out to achieve the best R/D performance. An input vector is quantized using L number of relevant optimum SQs instead of the total M number of optimum SQs and then the winning candidate, among the L quantized vectors, is selected for transmission by comparing with the input vector. Even though the GMVQ of [6] is claimed to be computationally efficient, a large amount of computation is spent on quantizing the input vector using all the mixture specific quantizers. Ostensibly, this is due to the requirement of minimizing the perceptually relevant SD measure which is not amenable for linear analysis. Also, due to the use of restricted number of mixture components (typically 8 or 16) for GM density modeling in GMVQ [6], the mixtures are highly overlapping in nature and thus, it is imperative to use all the mixture specific quantizers. Through experimentation [16], we have recently found that there exists a monotonic relation between SD measure and square Euclidean distance (SED) measure. Hence, in the developed UGMVQ method, we need not use the entire set of M SQs for an input vector, but instead use a subset of L SQs (where $L \ll M$) to achieve the near optimum quantization performance. For this, we incorporate a pre-processing block where an input vector is compared with M mean vectors of all Gaussian mixtures using SED measure (i.e. using nearest neighbor criteria) and the subset of L relevant optimum SQs is chosen from the set of M optimum SQs using rank ordering (sorting). The reduced search to L optimum SQs, permits us to choose a much higher value of M for better source density modeling using

GM density and also, reducing the intra-mixture covariance matrix to the near diagonal one; this removes the need for a de-correlating transform (KLT), as used in GMVQ of [6], and thus, helps further to reduce the complexity. Fig. 1 shows the developed UGMVQ method. The algorithmic steps associated with UGMVQ method are as follows:

(1) Pre-processing: The input vector is compared with the M mean vectors of all Gaussian mixtures using SED measure and the SED distance values are rank ordered (sorted); according to the rank ordering, the L number of relevant SQs are chosen from M SQs for quantization.
(2) Quantization: Quantizing the input vector using the selected L best SQs.
(3) Post-processing: Reconstructing the L quantized vectors. Choosing the best quantized vector using WSED measure (by comparing with the input vector) and transmitted.

2.3 SQ with optimum bit allocation

For each Gaussian mixture, the low complexity SQ, instead of several product code VQ methods, is chosen so that the quantizer retains the advantages of bitrate scalability and rate-independent complexity. To quantize an input vector using the mth Gaussian mixture specific optimum SQ, the input vector is mean removed and the scalar components of mean removed vector is variance normalized. Let us denote the mean removed and variance normalized vector for mth Gaussian mixture as $\underline{\mathbf{X}}_m = \text{diag}\left[\left\{\frac{1}{\sigma_{m,i}}\right\}_{i=1}^{p}\right][\mathbf{X} - \boldsymbol{\mu}_{\mathbf{m},\mathbf{X}}]$; the scalar components of $\underline{\mathbf{X}}_m$ vector are quantized independently using SQ with optimum bit allocation.

For a b bits/vector quantizer, $\log_2 M$ bits are used for transmitting the winning quantizer identity and thus, the rest $(b - \log_2 M)$ bits are used for SQ. For the mth Gaussian component specific optimum SQ, $(b - \log_2 M)$ bits are allocated optimally to the p scalar components as [1],

$$b_{m,i} = \frac{b - \log_2 M}{p} + \frac{1}{2}\log_2 \frac{\sigma_{m,i}^2}{\left(\prod_{k=1}^{p}\sigma_{m,k}^2\right)^{\frac{1}{p}}}, 1 \leq i \leq p, \quad (2)$$

where $b_{m,i}$ is the number of bits allocated to the ith scalar component for mth Gaussian mixture. The well-known water-filling algorithm may be applied to find the integer bit allocation from the above real bit allocation formula for practical requirement. It may be noted that the optimum bit allocation for each mixture specific SQ, provides the PDF dependent coding advantage, although the use of SQ results in a loss of space-filling advantage. But, the use of SQ provides for lower complexity and easy bitrate scalability. To address the issue of bitrate scalability, we mention that the optimality of R/D performance for UGMVQ method can be addressed easily by evaluating Eqn. 2 at any bitrate (i.e. at any allocation of b bits/vector) without retraining the quantizer for finding optimum codebooks.

According to the algorithmic steps, an input vector is quantized using the relevant L number of SQs. The L quantized vectors are compared with the input vector in post-processing stage for choosing the best quantized vector (or the winning SQ). Thus, the quantized informations are $\hat{L}$ using $\log_2 M$ bits and the indices of SQ codebooks using $\{b_{\hat{L},i}\}_{i=1}^{p}$ bits. At the receiver, the reconstructed vector is obtained after variance de-normalization and mean addition.

Table 1: Details of computational complexity for UGMVQ method

Operation	Complexity (flops)
Pre-processing: Choosing L relevant SQs from M SQs	$(3p+1)M + LM$
Quantization: Using L optimum SQs and reconstruction	
Mean vector subtraction	pL
Variance normalization: Find $\sigma_{m,i}$ from $\sigma_{m,i}^2$ and divide	$2pL$
Scalar quantization	$\mathbf{C}_{\text{SQ}}$
Post-processing: Reconstructing L quantized vectors and choosing the best one using WSED measure	
Variance de-normalization: Multiplying by $\sigma_{m,i}$	pL
Mean vector addition	pL
Choosing the best one using WSED measure	$(4p+1)L$

2.4 Computational complexity and memory requirement

The details of computational complexity (in flops)[1] incurred for UGMVQ method is shown in Table 1. The total computational complexity is given as (in flops),

$$\begin{aligned} \mathbf{C}_{\text{UGMVQ}} &= (3p+1+L)M + (9p+1)L + \mathbf{C}_{\text{SQ}} \\ &\approx (3p+1+L)M + (9p+1)L \end{aligned} \quad (3)$$

where $\mathbf{C}_{\text{SQ}}$ represents the necessary computation for SQ which maybe considered negligible compared to other terms.

Let us calculate the memory requirement (in floats)[2] for the proposed method. The number of parameters for defining the GMM with diagonal covariance matrices is: $M + 2pM$. Further, for each mth mixture specific optimum SQ, we need to store the bit allocation vector, i.e. $\{b_{m,i}\}_{i=1}^{p}$. Also, we need to store a bank of optimum SQ codebooks at different bits designed for zero mean, unit variance Gaussian random variable. Thus, the total required memory is given as (in floats),

$$\begin{aligned} \mathbf{M}_{\text{UGMVQ}} &= M + 2pM + pM + \mathbf{M}_{\text{SQ}} \\ &\approx (1+3p)M \end{aligned} \quad (4)$$

where $\mathbf{M}_{\text{SQ}}$ represents the necessary memory to store the bank of optimum SQ codebooks (designed for the zero mean, unit variance Gaussian random variable) which maybe considered negligible compared to other terms.

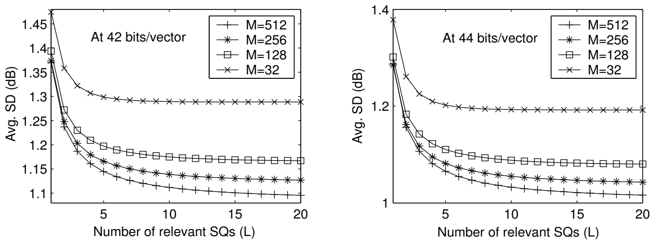
From the above calculations, it is observed that the developed UGMVQ method functions with a rate-independent complexity (i.e. computational complexity and memory requirement do not vary with the allocated b bits/vector). It is clear from Eqn. 3 and Eqn. 4 that the value of M determines mainly the computational complexity and memory requirement, since $L \ll M$. However, the computational complexity is much lower than the GMVQ method of [6], where effectively $L = M$ and additional complexity is required for the transforms, companding and SD computation.

3. QUANTIZATION EXPERIMENTS

The speech data used in the experiments is from the TIMIT data base, where the speech is sampled at 16 kHz. We have used the specification of AMR-WB speech codec [17] to

[1] In the current literature [6], [11], [14], [13], it is a standard practice to assume that each operation like addition, subtraction, multiplication, division and comparison needs one floating point operation (flop). With this assumption, the codebook search complexity for a b bits/vector VQ using SED measure is: $(3p+1)2^b$ flops. Using WSED measure, total computation required is: $(4p+1)2^b$ flops.

[2] The "float" represents the required memory to store a real value.



(a) (b)

Figure 2: Average SD (in dB) performance of UGMVQ method for different M and L to choose the optimum M and L. (a) Performance at 42 bits/vector. (b) Performance at 44 bits/vector.

Table 2: Performance of the developed UGMVQ method with $M = 256$ and $L = 10$

bits/vector	Avg. SD (dB)	SD Outliers (in %)	
		2-4 dB	>4 dB
42	1.138	1.86	0.001
43	1.095	1.37	0.000
44	1.054	1.05	0.000
45	1.015	0.82	0.000
46	0.976	0.58	0.000

compute 16-th order LP coefficients (i.e. $p = 16$), which are then converted to LSF parameters. In all the experiments, 361,046 LSF vectors are used for training and 87,961 vectors are used for testing (distinct from training data).

To measure the wideband speech LSF quantization performance, we use the traditional measure of SD. The conditions for transparent quality LSF quantization in telephone-band speech case are [2]:

(1) the average SD is around 1 dB.

(2) no outlier frame '> 4 dB' of SD.

(3) $< 2\%$ outlier frames are within the range of 2-4 dB.

It is mentioned in [4] that the same conditions are also valid for transparent quality quantization of wideband LSF parameters, as endorsed in [9], [13], [15]. Thus, we also use the above mentioned conditions in this paper for evaluating the wideband speech LSF quantization performance.

3.1 Performance of UGMVQ method

We experimentally find the optimum values of M and L such that the UGMVQ method provides a reasonable trade-off between R/D performance and complexity. Fig. 2 shows the average SD (in dB) performance of the UGMVQ method for different M and L (at 42 and 44 bits/vector). It is observed that the performance becomes better as M increases; this is due to the fact that the use of higher number of Gaussian mixtures results in better modeling of the LSF distribution. Also, for each M, the performance becomes better and then saturates as L increases, which is utilized to reduce complexity. We choose $M = 256$ so that the UGMVQ method incurs reasonable complexity. From Fig. 2, we choose $L = 10$ for the chosen value of $M = 256$. Even though the R/D performance improves for $M = 512$, but the complexities increase double-fold compared to $M = 256$ and thus, we restrict to $M = 256$. Table 2 shows the performance of UGMVQ method for $L = 10$ and $M = 256$; the associated rate-independent complexities are shown in Table 3. From Table 2, it can be observed that the UGMVQ method provides transparent quality quantization performance at 45 bits/vector.

Table 3: Rate-independent computational complexity and memory requirement of the UGMVQ method (with $M = 256$ and $L = 10$)

Computational complexity (CPU)	16.55 kflops/vector
Memory requirement (ROM)	12.54 kfloats/vector

Table 4: Performance of the traditional split VQ (SVQ) method

bits/vector	Avg. SD (dB)	SD Outliers		kflops/ vector (CPU)	kfloats/ vector (ROM)
		2-4 dB (in %)	>4 dB (in %)		
42	1.258	2.63	0.000	24.32	5.63
43	1.214	2.02	0.000	27.64	6.40
44	1.182	1.83	0.000	30.97	7.16
45	1.116	0.97	0.000	35.32	8.19
46	1.074	0.74	0.000	41.98	9.72

3.2 Comparison with other methods

We compare the performance of UGMVQ method over the traditional split VQ (SVQ), and recently proposed SSVQ and normalized two stage SVQ (NT_SSVQ) [15] methods. For SVQ, SSVQ and NT_SSVQ methods, we use WSED measure for quantization. In [13], it was shown that the SSVQ requires much lower computational complexity than the GMVQ method [6] for achieving transparent quality wideband LSF quantization performance; also, in [15], NT_SSVQ method was shown to provide comparable R/D performance with SSVQ at lower computational complexity and memory requirement.

In the case of SVQ [5], the 16-th dimensional LSF vector is split into 5 parts of (3,3,3,3,4) dimensional sub-vectors[3]; the performance of SVQ is shown in Table 4. The performance of five part SSVQ, with 8 switching directions, is shown in Table 5 where the bit allocation to the split sub-vectors is carried out according to [13]. Table 6 shows the performance of the NT_SSVQ method where the bit allocation is carried out according to [15].

It is noted that the developed UGMVQ method functions with rate-independent complexity as shown in Table 3. Comparing Table 2 and Table 4, it can be observed that the UGMVQ provides better R/D performance than SVQ at much lower computational complexity, but with higher memory requirement; the UGMVQ method saves 3 bits/vector compared to the SVQ method. Comparing with the performance of SSVQ method (Table 5), we observe that the UGMVQ method provides nearly same R/D performance; at 45 bits/vector, the UGMVQ method provides transparent quality quantization performance similar to SSVQ, but at much lower computational complexity and memory requirement. We note from Table 6 that the NT_SSVQ method is unable to provide transparent quality quantization performance be-

[3]Five part SVQ is also implemented in [9] [13] to compare with five part SSVQ and several other VQ methods.

Table 5: Performance of the recently proposed switched split VQ (SSVQ) method

bits/vector	Avg. SD (dB)	SD Outliers 2-4 dB (in %)	SD Outliers >4 dB (in %)	kflops/ vector (CPU)	kfloats/ vector (ROM)
42	1.123	1.28	0.001	19.08	34.94
43	1.071	0.85	0.001	20.74	38.01
44	1.036	0.72	0.000	22.40	41.08
45	1.003	0.59	0.000	25.73	47.23
46	0.967	0.46	0.000	29.06	53.37

Table 6: Performance of the recently proposed normalized two stage SVQ (NT_SSVQ) method

bits/vector	Avg. SD (dB)	SD Outliers 2-4 dB (in %)	SD Outliers >4 dB (in %)	kflops/ vector (CPU)	kfloats/ vector (ROM)
42	1.157	2.38	0.001	13.82	3.84
43	1.100	1.73	0.004	15.39	4.86
44	1.063	1.34	0.003	17.05	5.24
45	1.032	1.17	0.003	18.72	5.63
46	0.972	0.62	0.002	20.89	6.14

cause of its poor performance in the sense of '$>$ 4 dB' outliers. Thus, considering both complexity and R/D performance, the new UGMVQ method can be chosen as the best solution for LSF quantization in wide-band speech coding.

4. CONCLUSIONS

We have shown that the the challenging problem of achieving transparent quality LSF quantization performance in wide-band speech coding can be addressed using a low complexity UGMVQ method which is rate-independent and bitrate scalable. The new method is shown to perform best compared to the recent quantizers in the literature. We mention that, there is still scope for further improvement in R/D performance considering the theoretical lower bound of [16]; for example, the loss of space-filling advantage, due to using scalar quantization (SQ) in UGMVQ method, can be partially recovered using lattice VQ at a moderate increment in complexity.

REFERENCES

[1] A. Gersho and R.M. Gray, "Vector Quantization and Signal Compression," Kluwer Academic, 1992.

[2] K.K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 1, pp. 3-14, Jan 1993.

[3] W.R. Gardner and B.D. Rao, "Theoretical analysis of the high-rate vector quantization of LPC parameters," *IEEE Trans. Speech and Audio Processing*, vol. 3, No.5, pp. 367-381, Sept 1995.

[4] G. Guibe, H.T. How and L. Hanzo, "Speech spectral quantizers for wide-band speech coding," *European Trans. on Telecommunications*, 12(6), pp. 535-545, 2001.

[5] G.Biundo, S.Grassi, M.Ansorge, F. Pellandini and P. A. Farine, "Design techniques for spectral quantization in wide-band speech coding," *Proc. 3rd COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication*, pp. 114-119, Budapest, Oct. 2002.

[6] 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, March 2003.

[7] S.Kang, Y.Shin and T.R.Fischer, "Low-complexity predictive trellis coded quantization of speech line spectral frequencies," *IEEE Trans. Signal Processing*, vol. 52, No. 7, pp. 2070-2079, July 2004.

[8] S. So and K.K. Paliwal, "Multi-frame GMM-based block quantisation of line spectral frequencies for wide-band speech coding," *Proc. ICASSP*, Philadelphia, Vol. I, pp. 121-124, Mar. 2005.

[9] S. So and K.K. Paliwal, "Switched split vector quantisation of line spectral frequencies for wide-band speech coding," *Proc. EUROSPEECH (INTERSPEECH)*, Lisbon, Portugal, pp. 2705-2708, Sept. 2005.

[10] S. So and K.K. Paliwal, "Multi-frame GMM-based block quantisation of line spectral frequencies," *Speech Communication*, Vol. 47, pp. 265-276, Nov. 2005.

[11] 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, Language Processing,* vol. 14, Issue 2, pp. 524-532, March 2006.

[12] S. Chatterjee and T.V. Sreenivas, "Two stage transform vector quantization of LSFs for wideband speech coding," *Proc. ICSLP (INTERSPEECH)*, Pittsburgh, USA, pp. 233-236, Sept. 2006.

[13] S. So and K.K. Paliwal, "A comparative study of LPC parameter representations and quantisation schemes for wide-band speech coding," *Digital Signal Processing*, Vol. 17, Issue 1, pp. 114-137, Jan 2007.

[14] S.So and K.K.Paliwal, "Efficient product code vector quantisation using the switched split vector quantiser," *Digital Signal Processing*, Vol. 17, Issue 1, pp. 138-171, Jan 2007.

[15] S.Chatterjee and T.V.Sreenivas, "Normalized two stage SVQ for minimum complexity wide-band LSF quantization," *Proc. EUROSPEECH (INTERSPEECH)*, Antwerp, Belgium, pp. 1657-1660, Aug. 2007.

[16] S. Chatterjee and T.V. Sreenivas, "Predicting VQ performance bound for LSF coding," *IEEE Signal Processing Letters*, vol. 15, pp. 166-169, 2008.

[17] "AMR wide-band speech codec, transcoding functions (Release 5):" 3GPP TS 26.190 V 5.1.0