

Optimum Transform Domain Split VQ

Saikat Chatterjee, *Student Member, IEEE*, and T. V. Sreenivas, *Senior Member, IEEE*

Abstract—In this letter, we develop an optimum transform domain split vector quantization (TrSVQ) method. We address both the issues of achieving best rate-distortion (R/D) performance and less complexity. For quantizing a multivariate Gaussian source, we derive the mean-square error (MSE) performance expression for the TrSVQ method using high rate theory and optimum bit allocation. Also, to reduce the complexity, we develop a binary split-based iterative algorithm and use the algorithm in a tree structured manner to find the optimum subvectors' dimensions (i.e., optimum splits).

Index Terms—Transform coding, vector quantization.

I. INTRODUCTION

THE direct use of a full search vector quantization (VQ)¹ is limited because of its exponential complexity (for example, the full search VQ becomes prohibitively complex if the bit allocation exceeds 12–13 bits/vector). To overcome the complexity limitation, several structured/product code VQ methods [7] have been proposed and used widely.

A widely used product code VQ method is the split VQ (SVQ), also referred to as partitioned VQ [7], [8], [11]. In SVQ, the full vector is split into subvectors and then quantized independently. This approach results in a coding loss [13] as the correlation between the subvectors is not exploited. We have recently proposed the conditional PDF-based SVQ method [14], [15] to recover the coding loss. It can be noted that the use of a de-correlating transform before quantization will alleviate the coding loss of SVQ [10]. Splitting the transform domain vector into subvectors makes it feasible to handle larger dimensional vectors in the sense of manageable complexity and thus helps to achieve better performance by allocating a higher number of bits. Also, the transform domain quantization permits use of noise shaping techniques for achieving better subjective quality. It was shown in [4] that the SVQ of Fourier transformed signal vector provides better subjective performance than the traditional transform coders that do not use SVQ.

In the transform domain SVQ (TrSVQ) method [4], [10], [12], the distribution of the energy among the transform domain coefficients are highly skewed because of the decorrelating transform's compaction property. Thus, it is necessary to allocate the bits optimally among the subvectors to achieve the best R/D performance. Also, to reduce the complexity, it is important to address the issue of how to optimally split the transform domain vector into subvectors. The simple approach may be splitting the transform domain vector into subvectors of

same size and allocating the bits in proportion to the energy of the subvectors. This approach results in nonuniform bit allocation and thus leads to high complexity. Alternatively, to reduce the complexity, equal bits can be allocated to all the subvectors and the subvectors' dimensions can be adjusted based on either a distortion measure [6] or an energy measure [10], [12]. The subvectors' dimensions are experimentally adjusted in [10], [12] such that the total energy of the signal vector is distributed among the subvectors as evenly as possible. In [9], the issue of splitting is also addressed from a perceptual point of view. The above-mentioned methods do not readjust the subvectors' dimensions according to the allocated bits/vector for achieving least complexity; thus, the subvectors' dimensions are kept fixed for all the bitrates.

In this letter, we develop an optimum TrSVQ method for quantizing a multivariate Gaussian source using square Euclidean distance (SED) measure. We first address the R/D performance optimality for the TrSVQ method using optimum bit allocation and high rate MSE expression of VQ. The classic transform domain scalar quantization (TrSQ) method [1], [7] is shown to be a special case of the TrSVQ method. Using the derived optimum bit allocation formula, we also address the issue of complexity for the TrSVQ method. We develop a binary split-based iterative algorithm and use the algorithm in a tree structured manner to find the optimum subvectors' dimensions. At a particular bitrate, the iterative algorithm allows us to choose the subvectors' dimensions optimally for achieving less complexity but without sacrificing R/D performance.

II. PRELIMINARIES

In this section, we indicate the R/D performance of a full search VQ for quantizing a multivariate Gaussian source. Suppose the p -dimensional source vector \mathbf{X} is Gaussian distributed as $f_{\mathbf{X}}(\mathbf{x}) = \mathbf{N}(\mu_{\mathbf{X}}, C_{\mathbf{X}})$. Using the high rate quantization theory [5], the mean-square error (MSE) performance of a full search VQ is given as [15]

$$D_{VQ} = \mathcal{E}\{\|\mathbf{X} - \hat{\mathbf{X}}\|^2\} = (2^b)^{-\frac{2}{p}} K_p |C_{\mathbf{X}}|^{\frac{1}{p}} \quad (1)$$

where b is the allocated bits/vector to quantize the source; K_p is a constant which is dependent only on the value of dimension p as [15], $K_p = 2((p/2)\Gamma(p/2))^{(2/p)}((p+2)/p)^{(p/2)}$; $\Gamma(\cdot)$ is the usual gamma function as $\Gamma(n) = \int_0^\infty e^{-u} u^{n-1} du$.

III. R/D PERFORMANCE OF TRSVQ

We address the optimality of R/D performance for the TrSVQ method in this section. For quantizing a Gaussian source, KLT is the optimum de-correlating transform. Let the KLT and the transformed vector be, respectively, denoted by \mathbf{T} and \mathbf{Z} ; $\mathbf{Z} = \mathbf{T}\mathbf{X}$ and its distribution is characterized by a Gaussian, $f_{\mathbf{Z}}(\mathbf{z}) = \mathbf{N}(\mu_{\mathbf{Z}}, C_{\mathbf{Z}})$, where

$$\mu_{\mathbf{Z}} = \mathbf{T}\mu_{\mathbf{X}} \quad (2)$$

and

$$C_{\mathbf{Z}} = \mathbf{T}C_{\mathbf{X}}\mathbf{T}^T. \quad (3)$$

Manuscript received October 4, 2007; revised December 2, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Weifeng Su.

The authors are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560 012, India (e-mail: saikat@ece.iisc.ernet.in; tvsree@ece.iisc.ernet.in).

Digital Object Identifier 10.1109/LSP.2008.916723

¹In this letter, we use VQ and full search VQ interchangeably.

Equation (3) is nothing but the eigen value decomposition of $C_{\mathbf{X}}$. Also, $|C_{\mathbf{X}}| = |C_{\mathbf{Z}}|$ since \mathbf{T} is orthonormal. Let the eigen values of $C_{\mathbf{X}}$ be $\rho_1 \geq \rho_2 \geq \dots \geq \rho_p$, and thus, $C_{\mathbf{Z}} = \text{diag}\{\rho_i\}_{i=1}^p$. To reduce the VQ complexity, the vector \mathbf{Z} is split into S number of subvectors ($1 \leq S \leq p$) such that $\mathbf{Z} = [\mathbf{Z}_1^T \mathbf{Z}_2^T \dots \mathbf{Z}_S^T]^T$ and each \mathbf{Z}_i is coded independently. Let the dimension of \mathbf{Z}_i be q_i such that $\sum_{i=1}^S q_i = p$. Also, the maximum number of splits is $S = p$ when $q_i = 1$, and the TrSVQ method becomes the TrSQ method.

The use of KLT fully exploits the linear redundancy by removing the correlation between the vector components. Thus, each \mathbf{Z}_i can be independently quantized using SVQ technique without sacrificing the inter-subvector correlation, and the method is referred to as TrSVQ. Instead of the scalar quantization (SQ) as in TrSQ, the use of SVQ allows us to exploit the higher dimensional space filling advantage [3] of VQ.

Using the split vector notation, we can express the parameters of transform domain Gaussian distribution, $f_{\mathbf{Z}}(\mathbf{z}) = \mathbf{N}(\mu_{\mathbf{Z}}, C_{\mathbf{Z}})$, as

$$\begin{aligned} \mu_{\mathbf{Z}} &= [\mu_{\mathbf{Z}_1}^T \mu_{\mathbf{Z}_2}^T \dots \mu_{\mathbf{Z}_S}^T]^T, \quad \text{and} \\ C_{\mathbf{Z}} &= \begin{bmatrix} C_{\mathbf{Z}_1 \mathbf{Z}_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & C_{\mathbf{Z}_2 \mathbf{Z}_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & C_{\mathbf{Z}_S \mathbf{Z}_S} \end{bmatrix} \\ &= \text{diag}[\rho_1 \rho_2 \rho_3 \dots \rho_p] \end{aligned} \quad (4)$$

where $\mu_{\mathbf{Z}_i}$ and $C_{\mathbf{Z}_i \mathbf{Z}_i}$ are, respectively, the mean vector and covariance matrix of the i th subvector, \mathbf{Z}_i ; each $C_{\mathbf{Z}_i \mathbf{Z}_i}$ is a diagonal matrix of dimension $q_i \times q_i$; each subvector \mathbf{Z}_i is also Gaussian distributed as $f_{\mathbf{Z}_i}(\mathbf{z}_i) = \mathbf{N}(\mu_{\mathbf{Z}_i}, C_{\mathbf{Z}_i \mathbf{Z}_i})$.

To minimize the MSE distortion in TrSVQ, the available b bits/vector need to be allocated optimally among the transform domain subvectors. Using b_i number of bits for the i th subvector, the overall MSE distortion for the Gaussian source is written as

$$\begin{aligned} \mathcal{D}_{\text{tot}} &= \mathcal{E}\{\|\mathbf{X} - \hat{\mathbf{X}}\|^2\} = \mathcal{E}\{\|\mathbf{Z} - \hat{\mathbf{Z}}\|^2\} \\ &= \sum_{i=1}^S \mathcal{E}\{\|\mathbf{Z}_i - \hat{\mathbf{Z}}_i\|^2\} = \sum_{i=1}^S \mathcal{D}_i(b_i) \end{aligned} \quad (5)$$

where $\mathcal{D}_i(b_i)$ is the distortion for vector quantizing \mathbf{Z}_i with b_i bits allocated. Using the Gaussian source analysis of (1), $\mathcal{D}_i(b_i)$ is given as

$$\mathcal{D}_i(b_i) = (2^{b_i})^{-\frac{2}{q_i}} K_{q_i} |C_{\mathbf{Z}_i \mathbf{Z}_i}|^{\frac{1}{q_i}}, \quad 1 \leq i \leq S \quad (6)$$

where $K_{q_i} = 2((q_i/2)\Gamma(q_i/2))^{(2/q_i)}((q_i+2)/q_i)^{(q_i/2)}$. The optimum bit allocation is decided by minimizing the overall distortion subject to the constraint of total b bits/vector as

$$\min_{b_i} \left\{ \mathcal{D}_{\text{tot}} = \sum_{i=1}^S \mathcal{D}_i(b_i) \right\} \quad \text{subject to} \quad \sum_{i=1}^S b_i = b. \quad (7)$$

Using Lagrange optimization, we show the following result.

Theorem: For quantizing a Gaussian source, the optimum bit allocation to the transform domain subvectors in the TrSVQ

method, which minimizes the overall MSE distortion under the bitrate constraint of $\sum_{i=1}^S b_i = b$, is given by

$$b_i = q_i \frac{b}{p} + \frac{q_i}{2} \log_2 \left[\frac{\frac{1}{q_i} K_{q_i} |C_{\mathbf{Z}_i \mathbf{Z}_i}|^{\frac{1}{q_i}}}{\left[\prod_{j=1}^S \left(\frac{1}{q_j} K_{q_j} \right)^{q_j} \right]^{\frac{1}{p}} |C_{\mathbf{X}}|^{\frac{1}{p}}} \right] \quad 1 \leq i \leq S. \quad (8)$$

Using the optimum bit allocation, the minimum MSE distortion for the TrSVQ method is given by

$$\mathcal{D}_{\text{TrSVQ}} = (2^b)^{-\frac{2}{p}} \left[p \left[\prod_{i=1}^S \left(\frac{1}{q_i} K_{q_i} \right)^{q_i} \right]^{\frac{1}{p}} \right] |C_{\mathbf{X}}|^{\frac{1}{p}}. \quad (9)$$

Proof: Let $\mathcal{L} = \mathcal{D}_{\text{tot}} + \lambda(\sum_{i=1}^S b_i - b) = \sum_{i=1}^S (2^{b_i})^{-(2/q_i)} K_{q_i} |C_{\mathbf{Z}_i \mathbf{Z}_i}|^{1/q_i} + \lambda(\sum_{i=1}^S b_i - b)$. Using partial differentiation, $(\partial \mathcal{L})/(\partial b_i) = 0, 1 \leq i \leq S$, we get $b_i = -(q_i/2) \log_2[(\lambda/2 \ln 2)] - (q_i/2) \log_2[(q_i/K_{q_i}) |C_{\mathbf{Z}_i \mathbf{Z}_i}|^{(1/q_i)}]$. Summing $\forall i$ and using $\sum_{i=1}^S q_i = p, \sum_{i=1}^S b_i = b$, we get $-(1/2) \log_2[(\lambda/2 \ln 2)] = (b/p) + (1/p) \sum_{i=1}^S (q_i/2) \log_2[(q_i/K_{q_i}) |C_{\mathbf{Z}_i \mathbf{Z}_i}|^{(1/q_i)}]$. Substituting $-(1/2) \log_2[(\lambda/2 \ln 2)]$ appropriately and using $|C_{\mathbf{X}}| = |C_{\mathbf{Z}}| = \prod_{i=1}^S |C_{\mathbf{Z}_i \mathbf{Z}_i}|$ and dummy variable j , we get the optimum bit allocation formula of (8). Using the optimum bit allocation, the MSE of the TrSVQ method can be evaluated and shown in (9).

A. Coding Gain of TrSVQ Over TrSQ

For $S = p$ (i.e., $q_i = 1$), the TrSVQ method becomes the TrSQ method; using (9), the MSE performance of the TrSQ method is given as [7]

$$\mathcal{D}_{\text{TrSQ}} = \mathcal{D}_{\text{TrSVQ}} |_{S=p, q_i=1} = (2^b)^{-\frac{2}{p}} p k_g |C_{\mathbf{X}}|^{\frac{1}{p}} \quad (10)$$

where $k_g = K_{q_i} |_{q_i=1} = (\sqrt{3}\pi)/(2)$. Using (9) and (10), the coding gain of TrSVQ method over TrSQ method is given as,

$$\frac{\mathcal{D}_{\text{TrSQ}}}{\mathcal{D}_{\text{TrSVQ}}} = \frac{k_g}{\left[\prod_{i=1}^S \left(\frac{1}{q_i} K_{q_i} \right)^{q_i} \right]^{\frac{1}{p}}} \geq 1 \quad (11)$$

where the equality holds iff $S = p$. The above coding gain is purely a dimensionality-dependent parameter, independent of the Gaussian source parameters. Fig. 1 shows the coding gain of the TrSVQ method over the TrSQ method for different S . The coding gain of a full search VQ over TrSQ is also shown as the best achievable R/D performance (a point to note is that when $S = 1$, the TrSVQ method is nothing but a full search VQ). In Fig. 1, the discrete points are shown for the cases when the dimension p is divisible by the number of splits S such that $q_i = p/S, 1 \leq i \leq S$. As expected, the R/D performance of the TrSVQ method degrades as S increases.

IV. OPTIMUM SPLITS FOR TRSVQ

For a given S , an interesting question is how to split the transform domain vector into subvectors such that the TrSVQ method incurs less complexity. Thus, we address the issue of finding the optimum set of q_i . Using the method of independent codebook

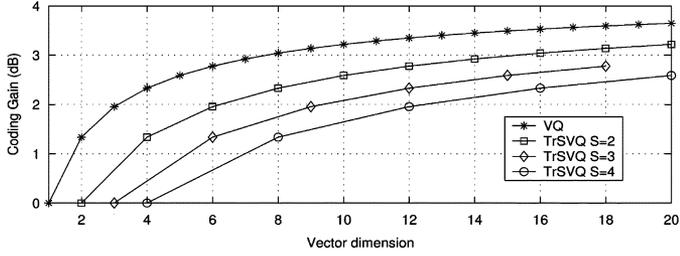


Fig. 1. Theoretical coding gain (in dB) of TrSVQ method (for different S) over TrSQ method. The coding gain of VQ is shown as the best achievable performance.

TABLE I

COMPARISON OF PRACTICAL SQNR PERFORMANCE (in dB) OF OPTIMUM TRSVQ METHOD WITH THE BEST ACHIEVABLE THEORETICAL PERFORMANCES OF TRSQ AND FULL SEARCH VQ METHODS

bits/vector	Practical SQNR (in dB)		Theoretical SQNR (in dB)	
	TrSVQ		TrSQ	Full search VQ
	$S = 2$	$S = 4$		
12	11.87	11.39	9.18	12.22
16	14.69	14.14	12.19	15.23
20	17.51	16.84	15.20	18.24
24	20.41	19.89	18.21	21.25
42	-	32.78	31.75	34.80
46	-	35.64	34.77	37.81

searching for each subvector, the required computational complexity of the TrSVQ method is given as (in flops)²

$$C = C' + \sum_{i=1}^S (3q_i + 1)2^{b_i} \approx C' + 3 \sum_{i=1}^S q_i 2^{b_i} \quad (12)$$

where C' accounts for all the computation (such as taking transform) except codebook searching. Also, the memory complexity of TrSVQ method is given as (in floats)

$$M = M' + \sum_{i=1}^S q_i 2^{b_i} \quad (13)$$

where M' accounts for all the memory (such as storage for KLT matrix) except memory required for codebook storage. Both computational complexity and memory requirement are reduced if $\sum_{i=1}^S q_i 2^{b_i}$ is minimized. We note that the complexity increases exponentially with the allocated bits/subvector (b_i) and linearly with the subvectors' dimensions (q_i). To counter the exponential rise of complexity, the full transformed vector needs to be split into subvectors such a way that the optimally allocated bits (b_i) are nearly uniform. For this, the subvectors' dimensions (q_i) have to be in ascending order since the eigen values are in descending order.

It can be seen that the problem of finding optimum q_i s is a combinatorial one and thus requires an algorithmic solution. For this, we develop a binary split-based iterative algorithm and use the algorithm in a tree structured manner (i.e., using the divide and conquer approach) to find S splits. Let us allow to

²It is assumed that each operation like addition, subtraction, multiplication, division, and comparison needs one floating point operation (flop). With this assumption, for p -dimensional vector and a b bits/vector full search VQ, the codebook search complexity using SED measure is $(3p + 1)2^b$ flops and codebook memory complexity is $p2^b$ floats. The "float" represents the necessary memory to store a real value.

describe the binary splitting algorithm. For given bits/vector, a binary splitting function is designed to split its input argument vector into two subvectors ($S = 2$) in such a way that the total bits/vector is optimally allocated using (8), but the complexity is also minimized. The binary splitting function is described below where we also address the issue of real to integer bit allocation for practical implementation.

Binary Splitting Function

Input: The diagonal covariance matrix (C_Z) of input argument vector Z which needs to split optimally into two subvectors as $Z = [Z_1^T Z_2^T]^T$; total allocated b bits/vector.

Output: The optimum dimensions of two subvectors; optimum bits allocated to the two subvectors.

Algorithm:

Declaration: $p = \dim(Z)$, $q_1 = \dim(Z_1)$, $q_2 = \dim(Z_2)$,

$$b_1 = \text{bits to } Z_1, b_2 = \text{bits to } Z_2, c = \sum_{i=1}^2 q_i 2^{b_i};$$

Initialization: $q_1^0 = \text{floor}(p/2)$, $q_2^0 = p - q_1^0$;

Find $C_{Z_1 Z_1}$ from C_Z , where it is assumed $\dim(Z_1) = q_1^0$;

Find b_1^0 using optimum bit allocation formula of (8);

Find $b_2^0 = b - b_1^0$ and $c^0 = \sum_{i=1}^2 q_i^0 2^{b_i^0}$;

Loop:

for ($k = 1; k < \text{floor}(p/2); k = k + 1$) {

$$q_1^k = q_1^{k-1} - 1, q_2^k = p - q_1^k; /* Ascending q_i s */$$

Find $C_{Z_1 Z_1}$ from C_Z , where it is assumed $\dim(Z_1) = q_1^k$;

Find b_1^k using optimum bit allocation formula of (8);

Find $b_2^k = b - b_1^k$ and $c^k = \sum_{i=1}^2 q_i^k 2^{b_i^k}$;

}

Minimum complexity: $k^* = \arg \min_k \{c^k\}$;

Integer bit allocation: $b_1 = \text{round}(b_1^{k^*})$, $b_2 = b - b_1$;

If ($b_2 < 0$) { $b_2 = 0$ }; $b_1 = b$; /* Nonnegative bit */

Return: $q_1 = q_1^{k^*}$, $q_2 = p - q_1^{k^*}$, b_1 and b_2 ;

Let us assume S be a power of 2. For the full transformed vector, the binary split-based function is called recursively in a tree structured manner ($\log_2 S$ stages) to find S number of optimum splits. For example, let $p = 8$ and $S = 4$. The eight-dimensional vector is split into two optimum dimensional subvectors using the binary splitting function. As the optimum bits allocated to each of the two subvectors are known, each of the two subvectors can be easily split further into two optimum subvectors using the binary splitting function, and thus, we get four optimum subvectors with optimum bit allocation. The tree structured manner of successive splitting does not incur any loss of R/D performance optimality as the full transformed vector is uncorrelated. Now, we extend the use of above algorithm for the case where S is not a power of 2. For this case, we first find S'

TABLE II
R/D PERFORMANCE AND COMPLEXITY COMPARISON BETWEEN THE OPTIMUM SPLIT TRSVQ AND THE EQUAL SPLIT TRSVQ METHODS AT $S = 2$

bits/ vector	Optimum split TrSVQ					Equal split TrSVQ				
	Optimum q_i s (q_1, q_2)	Optimum b_i s (b_1, b_2)	Theoretical SQNR (dB)	Computation (kflops/vector)	Memory (kfloats/vector)	Equal q_i s (q_1, q_2)	Optimum b_i s (b_1, b_2)	Theoretical SQNR (dB)	Computation (kflops/vector)	Memory (kfloats/vector)
12	(2, 6)	(6, 6)	11.55	1.66	0.51	(4, 4)	(10, 2)	11.51	13.36	4.11
16	(2, 6)	(7, 9)	14.56	10.62	3.32	(4, 4)	(12, 4)	14.52	53.45	16.44
20	(3, 5)	(11, 9)	17.49	28.67	8.70	(4, 4)	(14, 6)	17.53	213.82	65.79
24	(3, 5)	(13, 11)	20.56	114.68	34.81	(4, 4)	(16, 8)	20.54	855.29	263.16

number of optimum subvectors, where S' is the nearest lower integer number of S as power of 2. Then each of the higher dimensional ($S - S'$) number of subvectors is optimally split into two subvectors to find the optimum S number of splits. For example, let $S = 11$, and thus, $S' = 8$. First, we find eight optimum splits with optimum bit allocations, and then, three larger dimensional subvectors are again optimally split into six subvectors (with optimum bit allocations) to find total 11 subvectors' dimensions. Thus, the developed iterative algorithm provides both the optimum subvectors' dimensions or optimum splits (q_i) and optimum bits allocated to the subvectors (b_i).

V. QUANTIZATION RESULTS

We investigate the performance of the optimum TrSVQ method at different splits (S). The optimum bit allocations (b_i) and splits (q_i) are found using the iterative algorithm. We consider an eight-dimensional ($p = 8$) Gaussian vector source with zero mean and a covariance matrix with eigen values as (256,128,64,32,16,8,4,2). We use 500 000 vectors as the testdata for all the experiments carried out. The well-known LBG [2] algorithm is used to train the necessary quantizer codebooks using a separate training data of 500 000 vectors.

The performance of the developed optimum split TrSVQ method is investigated using $S = 2$ and $S = 4$. The signal-to-quantization-noise ratio (SQNR) performance is shown in Table I. For $S = 2$, the range of simulations is complexity limited because of the fact that the optimum bit allocation results in more than 13 bits/subvector at higher bitrate. We note that the TrSVQ method with $S = 2$ provides better performance than $S = 4$ as expected. The theoretical performances of TrSQ [using (10)] and full search VQ [using (1)] methods are also included in Table I as the best achievable performances for the respective methods. These results help us for direct comparison with the practical SQNR performances of TrSVQ. We note that the TrSVQ method provides considerable improvement over the TrSQ method. As expected, the full search VQ provides best theoretical performance, but its usage is complexity limited.

In Table II, we compare the R/D performance and required complexities of the developed optimum split TrSVQ method with the equal split TrSVQ method for $S = 2$. For the optimum split TrSVQ method, the use of iterative algorithm provides the optimum splits (optimum q_i) and optimum bit allocation (optimum b_i) according to the allocated bits/vector. We note that the iterative algorithm provides for subvectors' dimensions which are not fixed throughout all the bitrates. On the other hand, in the equal split TrSVQ method, the eight-dimensional transformed vector is split into (4, 4) dimensional subvectors

and the optimum bit allocation is carried out using (8). We note that both the methods provide comparable theoretical SQNR performances, but the equal split TrSVQ method incurs considerably higher complexity than the developed optimum split TrSVQ method.

VI. CONCLUSION

For a Gaussian source, we derive the R/D performance expression of the TrSVQ method using optimum bit allocation and thus show the coding gain over the existing TrSQ method in the high rate regime. For practical implementation of the TrSVQ method, we develop an iterative algorithm to choose the subvectors' dimensions in such a way that the TrSVQ method requires less complexity but without sacrificing the R/D performance.

REFERENCES

- [1] J. J. Y. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Trans. Commun.*, vol. CS-11, pp. 289–296, Sep. 1963.
- [2] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
- [3] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985.
- [4] P. C. Chang, R. M. Gray, and J. May, "Fourier transform vector quantization for speech coding," *IEEE Trans. Commun.*, vol. 35, no. 10, pp. 1059–1068, Oct. 1987.
- [5] R. M. Gray, *Source Coding Theory*. Dordrecht, The Netherlands: Kluwer, 1990.
- [6] W. Y. Chan and A. Gersho, "Constrained storage quantization of multiple vector sources by codebook sharing," *IEEE Trans. Commun.*, vol. COM-38, no. 12, pp. 11–13, Dec. 1991.
- [7] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
- [8] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 1, no. 1, pp. 3–14, Jan. 1993.
- [9] D. Chang, Y. K. Cho, and S. Ann, "Efficient quantization of LSF parameters using classified SVQ combined with conditional splitting," in *Proc. ICASSP*, 1995, vol. 1, pp. 736–739.
- [10] H. L. Vu and L. Loïs, "Efficient distance measure for quantization of LSF and its Karhunen-Loeve transformed parameters," *IEEE Trans. Speech, Audio Process.*, vol. 8, no. 6, pp. 744–746, Nov. 2000.
- [11] L. Hanzo, F. C. A. Somerville, and J. P. Woodard, *Voice Compression and Communications: Principles and Applications for Fixed and Wireless Channels*. New York: IEEE Press, 2001.
- [12] W. B. Mikhael and V. Krishnan, "Energy-Based split vector quantizer employing signal representation in multiple transform domains," *Digit. Signal Process.*, vol. 11, no. 4, pp. 359–370, Oct. 2001.
- [13] F. Norden and T. Eriksson, "On split quantization of LSF parameters," in *Proc. ICASSP*, May 2004, vol. 1, pp. I-157–160.
- [14] S. Chatterjee and T. V. Sreenivas, "Conditional PDF-based split vector quantization of wideband LSF parameters," *IEEE Signal Process. Lett.*, vol. 14, no. 9, pp. 641–644, Sep. 2007.
- [15] S. Chatterjee and T. V. Sreenivas, "Analysis of conditional PDF based split VQ," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 781–784, Nov. 2007.