ELSEVIER

Signal Processing 76 (1999) 311–322

SIGNAL PROCESSING

www.elsevier.nl/locate/sigpro

An associatively classified partitioned vector quantizer

C.Q. Chen, S.N. Koh*, I.Y. Soon

School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore

Received 24 April 1998; received in revised form 18 January 1999

Abstract

Vector quantization is an effective means of data compression which maps an ordered set of real numbers into a single integer. However, the unconstrained use of vector quantizer (VQ) for accurately compressing a high-dimension signal vector requires high computational complexity and large storage requirement. In order to avoid this problem, structures with some constraints are often used to involve more than one codebook in representing an input vector jointly. In this paper, a novel associatively classified partitioned VQ (ACPVQ) is presented as an attempt to exploit, in a simple way, both the intra and inter subset correlations among the original signal vector components. By the adoption of a modified M-L tree search algorithm plus a novel design procedure of joint partition-and-centroid over different sub-spaces, the novel ACPVQ achieves an average spectral distortion value below 1 dB at 21 bits/frame in quantizing line spectral frequencies (LSFs) which are widely used to represent the spectral envelope information of speech. Compared with the conventional partitioned VQ (Paliwal and Atal, 1993) and enhanced multistage VQ (LeBlanc et al., 1993), which can quantize LSFs with average spectral distortion values below 1 dB at 24 and 22 bits/frame, respectively, the new ACPVQ is obviously more efficient. However, the reduction in bit rate is accompanied by increases in both computational complexity and storage memory. © 1999 Elsevier Science B.V. All rights reserved.

Zusammenfassung

Ein effektives Verfahren zur Datenkompression stellt die Vektorquantisiering dar, welche eine geordnete Menge reeller Zahlen auf eine einzelne ganze Zahl abbildet. Allerdings benötigt die ausschließliche Verwendung des Vektorquantisierers (VQ) zur exakten Kompression eines hochdimensionalcen Signalvektors eine hohe Rechenkomplexität und großen Speicherplatzbedarf. Um dieses Problem zu umgehen, werden häufig Strukturen verwendet, die einigen Beschränkungen unterliegen, um mehr als eine Codeworttabelle in die Repräsentation eines Eingangsvektors einzubeziehen. In diesem Artikel wird ein neuartiger assoziativ klassifizierender partitionierter VQ (ACPVQ) vorgestellt. Es wird auf einfache Weise versucht, sowohl die Korrelationen innerhalb als auch zwischen Untermengen der ursprünglichen Signalvektorkomponenten auszunutzen. Durch Anpassung eines modifizierten Suchalgorithmus fur M-L Baume und mit Hilfe eines neuartigen Entwurfsverfahrens zur gemeinsamen Segmentierung und Schwerpunktsbestimmung über verschiedene Unterräume erzielt der neuartige ACPVQ einen durchschnittlichen spektralen Störpegel unterhalb von 1 dB mit 21 bits/Datensegment beim Quantisieren von Linienspektrumfrequenzen (LSFs), die breite Verwendung zur Repräsentation der spektralen Einhüllenden von Sprache finden. Im Vergleich zum konventionellen partitionierten VQ (Paliwal und Atal, 1993) und zum erweiterten mehrstufigen VQ (LeBlanc et al., 1993), die LSFs bei mittleren spektralen

* Corresponding author. Tel.: 65-799-5429; fax: 65-791-2687; e-mail: esnkoh@ntu.edu.sg

PII: S0165-1684(99)00017-1

Storpegeln unter 1 dB mit 24 bzw. 22 bits/Datensegment quantisieren können, ist der neue ACPVQ offensichtlich effizienter. Jedoch ist die Reduktion der Bitrate mit einer Erhöhung der Rechenkomplexität und des Speicherbedarfs verbunden.

Résumé

La quantification vectorielle est une manière efficace de comprimer des données. Elle associe un ensemble ordonné de nombres réels à un seul entier. Cependant, l'utilisation sans constraintes de la quantification vectorielle (QV) pour la compression précise d'un vecteur signal de grande dimension nécessite une complexité de calcul importante et des grandes exigences de stockage. Afin d'éviter ce problème, des structures présentant certaines contraintes sont souvent utilisées, afin d'impliquer plus d'un dictionnaire pour représenter conjointement un vecteur d'entrée. Dans cet article, la QV partitionnée classifiée associativement (QVPCA) est présentée comme une tentative d'exploitation de façon unique, à la fois les corrélations des sous-ensembles inter et intra parmi les composantes du vecteur signal original. Par l'adoption d'un algorithme de recherche en arbre M-L modifié, plus une nouvelle procédure de conception jointe de partitions et de centroïdes sur différents sous-espaces, la nouvelle QVPCA atteint une valeur de distorsion spectrale moyenne inférieure à 1 dB à 21 bits/trame en quantifiant les fréquences spectrales de lignes (FSL) qui sont largement utilisées pour représenter l'information d'enveloppe spectrale de la parole. Comparée à la QV partitionnée conventionnelle (Paliwal et Atal, 1993) et la QV multi-étape améliorée (LeBlanc et al., 1993), qui peut quantifier les FSL avec des valeurs de distorsion spectrale moyenne inférieures à 1 dB à 24 et 22 bits/trame respectivement, la nouvelle QVPCA est d'évidence bien plus efficace. Cependent, la réduction du débit binaire s'accompagne d'augmentations à la fois en complexité calcul et en mémoire de stockage.

Keywords: Vector quantizer; Speech coding

1. Introduction

In the use of unconstrained Voronoi VQ, an input vector is evaluated in terms of its distance to each of a set of codevectors. The codevector which matches the input vector most closely is used as an approximation to the input vector, and only its index is transmitted. Thus, the computational complexity and memory requirement involved in codebook search and storage are both proportional to $k \cdot N$, k where is the vector dimensionality and N is the codebook level. Applications of unconstrained VQ for accurate approximation to a high-dimension signal vector have consequently been impeded, although VQ has the potential to achieve coding performance close to the rate-distortion limit with increasing value of k. In order to circumvent the obstacle, a structure consisting of more than one codebook can be employed to achieve the desired coding precision for practical implementations. Such a structure is so designed as to optimize the coder performance at a specified value of resolution, while ensuring that the off-line training, real-time encoding complexity and the storage requirement involved in each sub-coder are at acceptable levels. Various forms of structurally constrained vector quantizer [1–10] such as multistage VQ, partitioned VQ and classified VQ have been suggested in the past.

Partitioned VQ [1,2] is well known as one of the simplest and most direct ways to encode a high-dimension signal vector, among the different kinds of structurally constrained VQ. As illustrated in Fig. 1, it seeks to achieve a good performance by designing and searching codebooks in different sub-spaces independently. However, as the inter-dependencies among different vector partitions are sacrificed in this procedure, the best possible overall performance may not be attained.

The use of multistage VQ (MSVQ) involves division of an encoding task into a number of successive stages. Taking advantage of the reduced mean square dynamic range of each co-ordinate in the residual stage, MSVQ reduces the dynamic volume to be covered by the residual codebook, where dynamic volume is the volume of the subspace within which a signal vector is confined in the statistical sense. It thereby achieves accurate

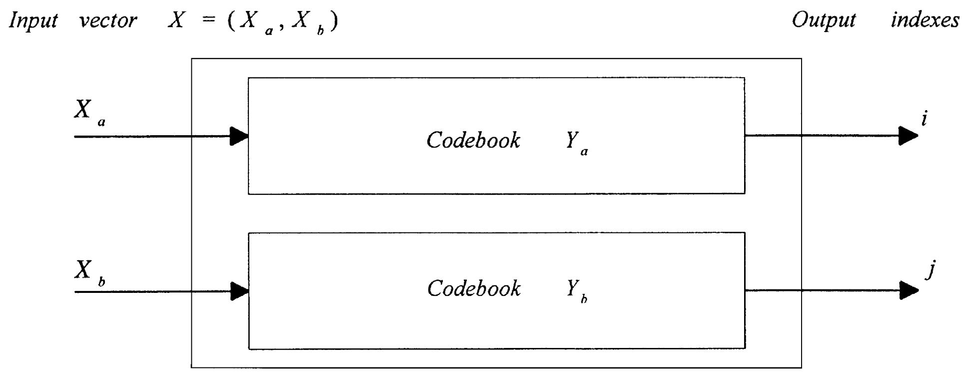

Fig. 1. The conventional two-partition VQ.

representation with an acceptable amount of computation. To include correlations among vectors in successive stages, many modified versions of MSVQ were proposed [3–15].

In a heuristic sense, classified vector quantizer puts each input into one of several patterns. The input vectors involved in each of the different patterns are approximated with each of the different pattern-oriented sub-codebooks. Usually, there are stronger correlations among the components of input vectors belonging to an individual pattern. Since stronger vector component dependencies are always tied to a smaller dynamic volume, and thus a better coding performance for a given resolution, classified vector quantizer is able to accurately encode the signal vector at an acceptable level of computational complexity. However, the efficiency of a classified VQ may not be as high as desired because some bits have to be used to identify the pattern to which each input vector belongs.

2. The basic concepts of the associatively classified partitioned vector quantizer

There are dependencies among different vector components of some signal sources. For example, the line spectrum frequencies of a frame of speech are strongly correlated [17]. Therefore, those input vectors whose projections are neighbours of each other in one sub-space tend to produce projections also close to each other in another sub-space; different subsets of vector components incline to a similar way of clustering or assembling. Also due to the component dependencies of a correlated signal source, input vectors tend to assume several quite different patterns in each sub-space.

Based on the above observations, an associatively classified partitioned VQ (ACPVQ) has been proposed. Its framework is similar to that of the conventional partitioned VQ. As illustrated in Fig. 2, in the encoding procedure, the first subvector is quantized by means of the first sub-space codebook. According to the resulting index of codevector, the second sub-vector is associated or predicted and then identified as one of several patterns via a simple process of retrieving look-up tables. The difference between the second sub-vector and its predicted version is calculated to form the associative residual. Finally, the identified pattern-oriented sub-codebook in the second subspace is fully searched to provide an approximation to the current associative residual. Such an 'associative classification and search' encoding process is repeated until the last vector partition is reached.

Without the use of any extra bits in the subvector classification, the proposed ACPVQ is found to outperform the conventional partitioned VQ significantly. Both the vector quantizers have the same total number of bits and highest codebook level assigned to each vector partition. They also employ the same LBG algorithm [11] to generate codevector sets. In the design procedure of

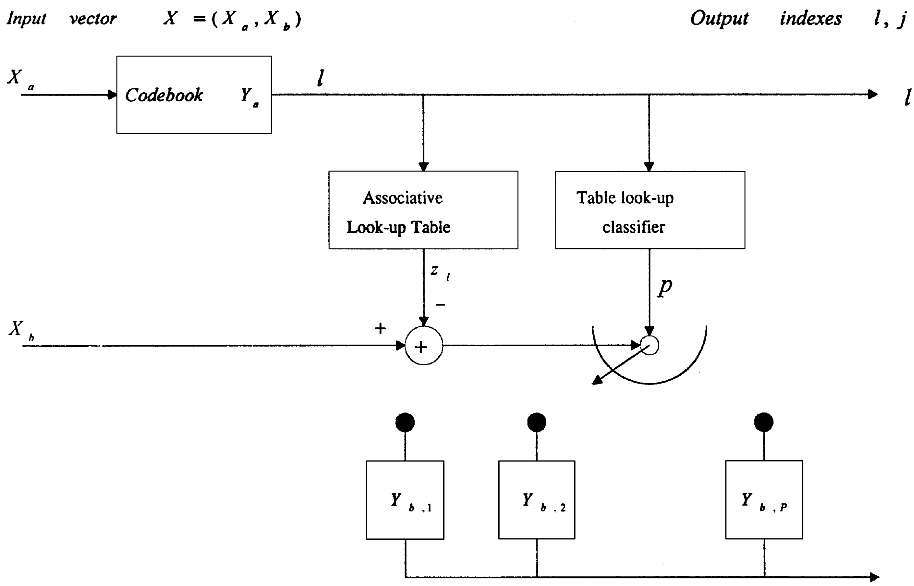

Fig. 2. The 'straightforward' encoder of ACPVQ.

ACPVQ, the first sub-space codebook is generated, first. According to the resulting clusters of the first sub-vectors in the above codebook generation procedure, the second subset of the training vectors is then clustered via a simple 'look-up assembling' process. That is, the second sub-vectors of those full-length training vectors whose projections in the first sub-space belong to the same cluster are assembled to form one cluster in the second sub-space. The clusters of the second sub-vectors thus obtained are referred to as the *associative clusters* and the centroids of all these associative clusters constitute an *associative look-up table*. Next, the associative clusters are classified into several patterns. Each member of an associative cluster is subtracted by the cluster centroid to produce an *associative residual*. All the associative residuals grouped under each pattern of associative clusters are used to generate each pattern-oriented sub-codebook in the second sub-space. Such a procedure is repeated until the last vector partition is reached.

In this paper, the process to approximate 'subsequent-to-the-first' sub-vector of each input with the centroid of its host associative cluster, according to the codevector indexes specified in the preceding sub-coders, is defined as an *association* operation, while the process to group a sub-vector of each input into one of the associative clusters in the current sub-space, according to the resulting clusters of training vectors in the preceding sub-coders, is defined as an *associative assembly*. Obviously, the 'closeness' of the sub-vectors belonging to an associative cluster depends on the inter-partition dependencies among different subsets of the vector components; a stronger inter sub-space correlation always leads to more 'cohesive' associative clusters.

3. Enhanced encoding and design procedures

In the straightforward encoder mentioned above, codebooks in different sub-spaces are searched sequentially. That is, the selection of codevectors in the second sub-coder is based on the outcome from the first sub-coder. Obviously, it involves constraint optimization of the reconstruction. To weaken the constraint, a modified version of M-L tree search [18] is proposed. By allowing multiple choices in the first sub-coder, a larger than zero but less than one degree of freedom is added into the average distortion minimization. The proposed procedure thus gains its advantage over the straightforward encoder. Beginning with the first sub-coder, the M codevectors which are the closest to the first sub-vector of each input are chosen and thus M associative residuals of each second input sub-vector are calculated. The pattern-oriented sub-codebook set in the second sub-space is then searched M times. The M con-catenations of the first and second sub-space codevectors that achieve the least reconstruction distortions are identified. Such a process continues for all the partitions. After the last partition is reached, the set of indexes associated with all sub-spaces that results in the minimum overall reconstruction distortion is specified. As this proposed procedure has a frame-work similar to that of the well-known M-L tree search, it is referred to as *associative M-L tree search*, where L is equal to the total number of partitions. Inevitably, such an encoding procedure incurs an increase in real-time computational complexity.

After the above design procedure, a *post-design procedure* is proposed to improve the effectiveness of associative M-L tree search. The mechanism of the post-design procedure is the same as that of GLA. In the procedure, each training sub-vector is partitioned into one of its nearest codevectors by the use of the associative M-L tree search method. The codevectors in different sub-spaces are then updated as the centroids of their members of training sub-vectors. Therefore, the proposed post-design procedure is based on the well-established GLA. It thus is able to improve the effectiveness of associative M-L tree search. The details of the post-design procedure are given in Section 4.

In summary, the proposed ACPVQ gains its superiority over the conventional partitioned VQ through its use of the inter sub-space correlations to enhance the sub-vector component dependencies and thus reduces the dynamic volumes for some sub-coders. Among the different mechanisms adopted by ACPVQ, the association operation is, in some aspects, similar to the approach of predictive vector quantizer [1].

4. The associatively classified partitioned vector quantizer

Suppose that the vector dimensionality is k. Let the upper limit of the number of bits allocated to each codebook, which is associated with the highest level of computational complexity acceptable, be H, and thus the largest size of each codebook be 2^H. Let the total number of bits consumed by an associatively classified *two-partition* vector quantizer be B, $2 \leqslant B \leqslant 2H$, the number of bits assigned to the first sub-coder be b, and thus the number of bits allocated to the second sub-coder be $B - b$. Let the number of patterns of the second vector partition be P. The design, 'straightforward' and 'associative M-L tree search' encoding, as well as decoding procedures of ACPVQ can be stated as follows.

4.1. Design procedure

Step 1. Set m, the dimensionality of the first vector partition, to be $k - 2$.

Step 2. Split each of the training vector, $X = (x_1, x_2, \ldots, x_k)$, at the mth co-ordinate so as to form two subsets of training vectors, i.e.,

$$X = (X_a, X_b); \quad X_a = (x_1, x_2, \ldots, x_m),$$

$$X_b = (x_{m+1}, x_{m+2}, \ldots, x_k).$$

Step 3. Generate the 2^b-level codebook $Y_a = \{y_{a,1}, y_{a,2}, \ldots, y_{a,2^b}\}$ in the first sub-space, using the LBG algorithm [11] for the first subset of training vectors formed in Step 2.

Step 4. Associatively assemble the second partition of each of training vectors, $X_b = (x_{m+1}, x_{m+2}, \ldots, x_k)$, to form associative clusters A_i $(i = 1, 2, \ldots, 2^b)$, whose centroids are denoted by z_i $(i = 1, 2, \ldots, 2^b)$, in the second sub-space.

(a) Cluster the first partition of each training vector, X_a, according to the codebook Y_a generated in Step 3, i.e.,

$$R_{i,a} = \{X_a : d(X_a, y_{a,i}) \leqslant d(X_a, y_{a,j});$$
$$\text{for } y_{a,i}, y_{a,j} \in Y_a \text{ and all } i \neq j \in [1, 2^b]\}.$$

(b) Select those training vectors whose portions in the first sub-space belong to the ith cluster, $\{X : X_a \in R_{i,a}\}$, and then group the second sub-vectors of the selected training vectors to form the ith associative cluster A_i with the centroid, z_i, i. e.,

$$A_i\{X_b : X; X : X_a \in R_{i,a}\}, \quad z_i = cent(A_i),$$

where the value of integer i ranges from 1 to 2^b.

Step 5. Classify all the associative clusters A_i ($i = 1, 2, \ldots, 2^b$) formed in Step 4 into P patterns whose centroids are symbolized as c_j ($j = 1,2, \ldots, P$), by the use of a modified LBG algorithm. In this modified algorithm, the distance measure $d^*(A_i, c_j)$ between an associative cluster A_i and a pattern centroid c_j is defined as the sum total of the Euclidean distances associated with all the second sub-vectors involved in A_i, i.e., $d^*(A_i, c_j) = \sum_{X_b \in A_i} d(X_b, c_j)$. Each centroid c_j is defined as the average or center of the sub-vectors grouped under each pattern of associative clusters and only the pattern centroid with the maximum distortion, instead of all the pattern centroids, is split at each stage. In other words, all the second sub-vectors whose 'elder brothers' are approximated by the same codevector $y_{a,i}$ in the first subcoder are classified into one of P patterns; the index of each codevector in the first subspace codebook, i, points to p, the index of one of p patterns of sub-vectors in the second subspace, i.e., $i \in [1, 2^b] \to p \in [1, P]$.

Step 6. Perform the association operation on X_b, the second partition of each training vector, so as to produce pattern-oriented categories of associative residuals in the second sub-space;

(a) Find the codevector $y_{a,l}$ in the first sub-space codebook y_a which matches X_a, the first partition of each training vector, most closely, i.e.,

$$d(X_a, y_{a,l}) \leqslant d(X_a, y_{a,i}); \; y_{a,l}, y_{a,i} \in Y_a, \quad \text{all } l \neq i \in [1, 2^b].$$

(b) Pick up z_l, the centroid of the l-th associative cluster obtained in Step 4. Calculate the associative residual $X_b - z_l$ and then put it into the p-th category, according to the l-th associative cluster identification, $l \to p$, which is done in Step 5.

Step 7. For $p = 1, 2, \ldots, P$, train all the associative residuals in the p-th category which is obtained in Step 6, $\{X_b - z_l : X : X_a \in R_{a,l};$ all $l \to p, l \in [1, 2^b]\}$, to generate the p-th pattern-oriented 2^{B-b}-level sub-codebook in the second sub-space that is symbolized as $Y_{b,p} = \{y_{b,p,1}, y_{b,p,2}, \ldots, y_{b,p,2^{B-b}}\}$, using the LBG algorithm. Compute the overall distortion of ACPVQ by summing the average distortions of all the sub-coders, where the average distortion of second sub-coder is equal to the membership-weighted average of the Euclidean distortions associated with all the P patterns.

Step 8. Record the current codebook in the first sub-space, Y_a, centroids of associative clusters of the second sub-vectors, z_i ($i = 1, 2, \ldots, 2^b$), pattern-oriented sub-codebooks in the second sub-space, $Y_{b,p}$ ($p = 1, 2, \ldots, P$), dimensionality of the first vector partition m if the coder performance resulting in Step 7 is the best so far. Decrease m by 1. Go to Step 2 if m is not less than 2, otherwise, stop.

Obviously, the different values of b ranging from $(B - H)$ to H have to be evaluated in terms of the average distortion, for the purpose of determining the best allocation of bits to each vector partition. Moreover, the above procedure should be directly generalized to design the sub-coder subsequent to the second, if more than two vector partitions are required. The configuration and all the corresponding codebooks which give rise to the best final coding performance are kept for the use of ACPVQ. More off-line computation is, inevitably, required by the above procedure, compared with that involved in conventionally optimizing partitioned VQ.

4.2. Post-design procedure

Step 1. Search the first sub-space codebook to find the most representative codevectors $y_{a,l_1}, y_{a,l_2}, \ldots, y_{a,l_M}$ for each training sub-vector X_a. According to the resulting indexes $l_1, l_2, \ldots, l_M$, calculate the corresponding associative residuals of each training sub-vector, $X_b - z_{l_i}$ ($i = 1, 2, \ldots, M$).

Step 2. Identify the pth pattern to which the l_ith associative cluster belongs, according to the associative cluster classification done in the above design procedure. Search the identified pattern-oriented sub-codebook in the second sub-space to find the most representative codevector catenation (y_{a,l_i}, y_{b,p,j_i}) for $(X_a, X_b - z_{l_i})$, for $i = 1, 2, \ldots, M$.

Finally, specify the index i^* and thus l_{i^*} as well as j_{i^*} associated with the least overall reconstruction distortion.

Step 3. According to the index l_{i^*} of each first training sub-vector X_a, calculate the centroid of each first sub-space cluster as the improved codevector. Based on the associative cluster classification done in the above design procedure, $l_{i^*} \to p^*$, re-calculate the centroid to improve the associative look-up table and each pattern-oritented sub-codebook in the second sub-space, in terms of the index j_{i^*} of each $X_b - z_{l_{i^*}}$.

Obviously, the above steps should be repeated until the improvement in coding performance due to joint partition-and-centroid is small enough. The use of such a procedure inevitably incurs additional off-line computational requirement.

4.3. The 'straight-forward' encoding procedure

As illustrated in Fig. 2, the 'associative classification and search' strategy incurs only a very small increase in the amount of real-time computation, compared with the independent codebook search manner in the use of the conventional partitioned VQ. Such an increase in complexity is equal to $(k - m)$ additions per input vector.

4.4. The associative M-L tree search (encoding) procedure

Suppose that l_i $(i = 1, 2, \ldots, M)$ are the indexes of the codevectors in codebook Y_a which are the most representative ones for X_a. Let p_i be the index of the pattern to which the l_ith associative cluster belongs, and j_i $(i = 1, 2, \ldots, M)$ be the indexes of the codevectors in codebook Y_{b,p_i} which are the nearest neighbours of $X_b - z_{l_i}$. The associative M-L tree search (encoding) procedure is illustrated in Fig. 3.

Compared with the straightforward encoder, the higher real-time computational complexity involved in this associative M-L tree search procedure is concomitant with a drop in average coding distortion. Both the coding accuracy and computational demand are proportional to the value of M. The difference in real-time complexity between two encoding procedures and, more in general, between the proposed technique and the optimized partitioned VQ is roughly quantified as follows:

Logical comparison: $\leqslant (2M - 1)$,
Multiplication: $(M - 1)(k - m)2^b$,
Addition: $(M - 1)(k - m - 1)2^b$.

4.5. Decoding procedure

The decoding procedure is rather straightforward and it is illustrated in Fig. 4.

It is apparent from the above procedures that ACPVQ draws on the favourable aspects of both partitioned VQ and classified VQ. It also draws on the favourable aspects of multistage VQ for the association operation somewhat executes the function of the first stage in quantizing the second partition of each input vector. Although the optimal form of structurally constrained VQ may not necessarily be thus found, the inter sub-space association operation and associative cluster classification based on a 'cluster similarity classified' strategy does reduce the dynamic volumes and enhance the dependencies among sub-vector components within an individual pattern. The improved performances of the sub-coders subsequent to the first are thus achieved. The use of associative M-L tree search can therefore lead to better performances than the straightforward encoding procedure in the proposed ACPVQ.

The improved performance of the associative M-L tree search method over the straightforward

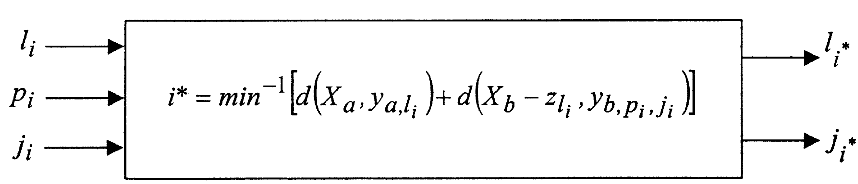

Fig. 3. The 'associative M-L tree search' encoder of ACPVQ.

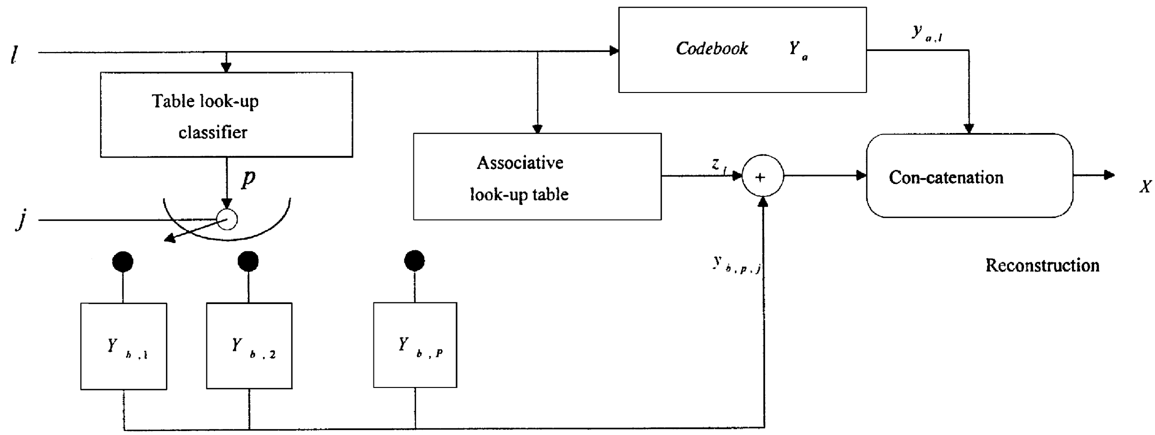

Fig. 4. The decoder of ACPVQ.

procedure can also be explained in a slightly different way. In the straightforward encoding procedure, codebooks in different sub-spaces are searched sequentially. That is, codevector selection in the second sub-coder is based on the outcome from the first sub-coder. Obviously, it is a form of 'constraint optimization'. The associative M-L tree search procedure tries to weaken the constraint by allowing multiple choices in the first sub-coder and thus an improvement in performance is attained. Consequently, the associative M-L tree search method outperforms the straightforward procedure.

5. Application of associatively classified partitioned vector quantizer for the coding of LPC Information

Linear Predictive Coding (LPC) parameters are widely used in speech coders to convey the spectral envelope information. For low bit rate representation, it is important to encode these parameters to achieve as low a value as possible for the product, (number of bits per frame) × distortion. Hence, structurally constrained VQ is an important technique to code LPC information [12–16].

Among different representations of the LPC information, line spectral frequency (LSF) [21] is the most widely used. LSFs are approximately related to the formant frequencies as well as bandwidths. They can be easily obtained from, and/or exactly inversely transformed into, the original LPC coefficients. Moreover, the quantization of LSFs can be done under the natural guarantee for the stability of LPC synthesis filter. Furthermore, the use of LSFs localizes the spectral sensitivity, that is, an adaptation in a specified LSF changes the LPC power spectrum only in the neighbourhood of the adapted frequency.

In our application of the proposed ACPVQ for coding LPC information, an LPC vector is made up of ten line spectral frequencies which are transformed from the LPC coefficients obtained by the use of a fast fixed point covariance lattice algorithm [19,20]. The training and test data sets consist of selections of speech signals from the TIMIT database [22], which contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. The sampling frequency is 8 kHz and the analysis frame rate is 50 frames/s with 160 samples/frame.

For each of the specified highest codebook levels, each of the different total numbers of bits, and each of the different training sets, the performances of the proposed VQ structure are compared with those of the conventional two-partition VQ with the optimal configuration; the optimal configuration of partitioned VQ is obtained by evaluating all the possibilities. The simulation programmes were written in C language and the distortion of the encoded LPC parameters inside the training and test sets was measured in terms of the average Euclidean distortion (AED) and the average Spectral distortion (ASD) [23].

Tables 1–5 show that the coder performance of the associatively classified two-partition VQ is always superior to that of the conventional two-partition VQ with the optimal configuration; both of them use the same total number of bits and highest codebook level, while of course, the reconstruction of the LSFs inside the test set is worse than that inside the training set. As expected, the improvement in coding performance is proportional to the pattern number of associative clusters. Also shown in Tables 1–4, the improvement due to increasing by one the number of patterns of the second sub-vectors is somewhat small. This phenomenon may be attributed to the nature of LSFs, as well as the indirect assembly manner of associative clusters, and thus the indirect classification of the second sub-vectors. However, the improvement becomes significant as the pattern number P is large enough; the ACPVQ with $P = 22$ outperforms the optimally configured partitioned VQ by more then 25% according to the average Euclidean distortion inside the training set, as illustrated in Table 1(c).

By substituting the weighted Euclidean distance measure [13] for the conventional Euclidean distance measure, in the procedures of codebook generation and search, further reduction in average spectral distortion can be achieved. This weighted Euclidean distance between the original input vector LSF and its quantized version LSF_q may be denoted as

$$d(\mathrm{LSF}, \mathrm{LSF}_q) = \sum_{i=1}^{k} \{c_i w_i[\mathrm{LSF}(i) - \mathrm{LSF}_q(i)]\}^2,$$

where $\mathrm{LSF}(i)$ and $\mathrm{LSF}_q[i]$ are the ith components of LSF and LSF_q, respectively, $w_i = (P(\mathrm{LSF}(i))^{0.15}$; $P(f)$ is the LPC power spectrum associated with

Table 1

Distortion comparison of ACPVQ and the optimized PVQ with the straightforward encoding process and conventional Euclidean distance measure for $B = 12, H = 6\ (2^H = 64)$ or bit assignment is 6 + 6

	Vector decomposition	AED (*1.e − 3)		ASD (dB)		Storage (real number)
		Training	Test	Training	Test	
(a) The pattern number of the second sub-vectors $P = 2$–5, size of training set = 4536, and size of test set = 4536						
PVQ	(6, 4)	0.639	0.681	2.59	2.66	640
ACPVQ $P = 2$	(3, 7)	0.581	0.649	2.49	2.58	1088
ACPVQ $P = 3$	(3, 7)	0.553	0.640	2.43	2.59	1536
ACPVQ $P = 4$	(3, 7)	0.535	0.631	2.39	2.58	1984
ACPVQ $P = 5$	(3, 7)	0.513	0.630	2.35	2.57	2432
(b) The pattern number of the second sub-vectors $P = 6$–9, size of training set = 9168, and size of test set = 9168						
PVQ	(6, 4)	0.660	0.691	2.62	2.67	640
ACPVQ $P = 6$	(3, 7)	0.550	0.637	2.41	2.56	2880
ACPVQ $P = 7$	(3, 7)	0.538	0.628	2.38	2.56	3328
ACPVQ $P = 8$	(3, 7)	0.531	0.625	2.36	2.55	3776
ACPVQ $P = 9$	(3, 7)	0.523	0.618	2.35	2.54	4224
(c) The pattern number of the second sub-vectors $P = 18$–22, size of training set = 19 098, and size of test set = 19 098						
PVQ	(6, 4)	0.657	0.672	2.59	2.61	640
ACPVQ $P = 18$	(3, 7)	0.499	0.596	2.28	2.47	8256
ACPVQ $P = 19$	(3, 7)	0.496	0.598	2.27	2.47	8704
ACPVQ $P = 20$	(3, 7)	0.491	0.598	2.26	2.47	9152
ACPVQ $P = 21$	(3, 7)	0.484	0.596	2.24	2.46	9600

Table 2
Distortion comparison of ACPVQ and the optimized PVQ with the straightforward encoding process and conventional Euclidean distance measure for $B=16$, $H=8$ ($2^H=256$) or bit assignment is $8+8$, size of training set = 9168, and size of test set = 9168

	Vector decomposition	AED (*1.e − 3)		ASD (dB)		Storage (real number)
		Training	Test	Training	Test	
PVQ	(6, 4)	0.358	0.389	1.97	2.04	2560
ACPVQ $P=2$	(4, 6)	0.316	0.362	1.88	1.99	4096
ACPVQ $P=3$	(4, 6)	0.303	0.359	1.84	1.98	5632
ACPVQ $P=4$	(4, 6)	0.296	0.358	1.82	1.99	7168
ACPVQ $P=5$	(4, 6)	0.291	0.357	1.80	1.98	8704

Table 3
Distortion comparison of ACPVQ and the optimized PVQ with the straightforward encoding process for $B=22$, $H=11$ ($2^H=2048$) or bit assignment is $11+11$, size of training set = 141 320, and size of test set = 32 668

	Vector decomposition	AED (*1.e − 3)		ASD (dB)		Storage (real number)
		Training	Test	Training	Test	
(a) With the use of the conventional Euclidean distance measure						
PVQ	(6, 4)	0.158	0.172	1.33	1.39	20480
ACPVQ $P=4$	(4, 6)	0.122	0.151	1.18	1.29	57344
(b) With the use of the weighted Euclidean distance measure						
PVQ	(6, 4)			1.11	1.23	
ACPVQ $P=4$	(4, 6)			0.93	1.12	

Table 4
Distortion comparison of ACPVQ and the optimized PVQ with the straightforward encoding process for $B=24$, $H=12$ ($2^H=4096$) or bit assignment is $12+12$, size of training set = 14 1320, size of test set = 32 668

	Vector decomposition	AED (*1.e-3)		ASD (dB)		Storage (real number)
		Training	Test	Training	Test	
(a) With the use of the conventional Euclidean distance measure						
PVQ	(6, 4)	0.116	0.133	1.15	1.23	40 960
ACPVQ $P=2$	(4, 6)	0.096	0.120	1.05	1.15	114 688
(b) With the use of the weighted Euclidean distance measure						
PVQ	(6, 4)			0.97	1.09	
ACPVQ $P=2$	(4, 6)			0.85	1.00	

the original input vector LSF and

$$c_i = \begin{cases} 1.0 & \text{for } 1 \leqslant i \leqslant 8, \\ 0.8 & \text{for } i = 9, \\ 0.4 & \text{for } i = 10. \end{cases}$$

By the use of the above weighted Euclidean distance measure, the conventional two-partition VQ achieves an average spectral distortion of lower than 1 dB at 24 bits per frame with the size of each codebook equal to 4096 for 10 order LSF quantization [13]. While the conventional multistage VQ

Table 5
Coding performance of ACPVQ with associative $M-L$ tree search and post-design procedure for $M=28$, $B=21$, $H=12$ ($2^H=2048$) or bit assignment is 12 + 9, size of training set = 141 320, and size of test set = 32 668

	Vector decomposition	AED (*1.e-3)		ASD (dB)	
		Training	Test	Training	Test
(a) With the use of the conventional Euclidean distance measure					
ACPVQ $P=18$	(4, 6)	0.113	0.130	1.12	1.22
(b) With the use of the weighted Euclidean distance measure					
ACPVQ $P=18$	(4, 6)			0.96	1.08

was reported to have performed worse than the conventional partitioned VQ [13], W.P. LeBlanc et al. adopted the M-L search algorithm in the codevector selection procedure of multistage VQ and jointly optimized all the codevector sets over all the stages in the codebook design procedure. They thus achieved 1 dB average spectral distortion of LSFs at 22 bits per frame [15]. As shown in Table 5, the novel associatively classified two-partition VQ with $p=4$ is able to achieve about 1 dB average spectral distortion for LSF quantization at 21 bits per frame; ACPVQ is somewhat more efficient.

It is important to emphasize that the use of the proposed ACPVQ will incur an increase in the storage requirement, compared with the use of the conventional partitioned VQ with the same bit assignment and sub-vector dimensionalities. For the two-partition case, the increase in memory requirement is equal to $P\cdot(k-m)\cdot 2^{B-b}$ real numbers, where P is the pattern number of the associative clusters of the second sub-vectors, k and m are the dimensionalities of the signal vector and the first vector partition, respectively, B and b are the total number of bits and the number of bits assigned to the first sub-coder, respectively.

6. Conclusion

The performance of the proposed associatively classified partitioned VQ has been compared with that of the conventional partitioned VQ with the jointly optimal vector decomposition and bit allocation. The performance measures used are the average Euclidean distortion and Spectral distortion. With respect to both these measures, the proposed ACPVQ is found to outperform the conventional partitioned VQ with the optimal configuration, while using the same total number of bits and highest codebook level as well as the same LBG algorithm for codebook generation. With the improvement in performance, the rise in the real-time computational complexity incurred by the straightforward encoding procedure is negligible, and the increase in the storage requirement is acceptable. By the use of associative $M-L$ tree search plus the post-design process, further improvement in coding performance of ACPVQ can be achieved at the cost of an increase in computational complexity; both the reconstruction accuracy and real-time computational requirement are proportional to the value of M. In addition, the new concepts concerning associative assembly, association operation and associative cluster classification as well as associative $M-L$ tree search may establish bearing points for future investigations. For example, they may be used to jointly quantize the LPC parameters, pitch frequencies and even excitation vectors which may not be 'white' inCELP-based speech coders, so as to reduce bit rate while maintaining a satisfactory quality of the synthesized speech. However, the above achievements are accompanied by a higher level of off-line computational complexity involved in the design procedure.

References

[1] A. Gersho, M.M. Gray, in: Vector Quantization and Data Compression, Kluneer, Boston, 1992.

[2] M.J. Sabin, R.M. Gray, Product code vector quantizers for waveform and voice coding, IEEE Trans. Acoust. Speech, Signal Process. (June 1984) 474–488.

[3] F. Kossentini et al., Conditional entropy-constrained residual VQ with application to image coding, IEEE Trans. Image Process. 5 (February 1996) 311–320.

[4] W.Y. Chan, S. Gupta, A. Gersho, Enhanced multistage vector quantization by joint codebook design, IEEE Trans. Commun. 40 (November 1992) 1693–1697.

[5] F. Kossentini, M.J.T. Smith, Adaptive entropy-constrained residual vector quantization, IEEE Signal Process. Lett. 1 (August 1994) 121–123.

[6] S.A. Rizvi, N.M. Nasrabad, Variable-rate predictive residual vector quantizer, IEEE Signal Process. Lett. 2 (April 1995) 70–72.

[7] S.S. Huang, H.J. Lin, J.S. Wang, W.T. Chen, FASVQ: a feasible vector quantization architecture, in: Proc. 1992 IEEE Internat. Symp. Circuit and System, San Diego, CA, USA, 10–13 May 1992, Vol 5, pp. 2296–2299.

[8] H.S. Huang, N. Moayeri, Trellis coded vector quantization, IEEE Trans. Commun. 40 (August 1992) 1273–1276.

[9] P.F. Swaszek, Unrestricted multistage vector quantizers, IEEE Trans. Information Theory 38 (May 1992) 1169–1174.

[10] L. Chang, M.M. Bayoumi, An economical binary tree structure for vector quantization, in: Proc. ICASSP 91, Ont., Canada, 14–17 April 1991, Vol. 1, pp. 669–672.

[11] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. 28 (January 1980) 84–95.

[12] R.P. Ramachandran, M.M. Sondhi, N. Seshadri, B.S. Atal, A two codebook format for robust quantization for line spectral frequencies, IEEE Trans. Acoust. Speech Signal Process. 3 (May 1995) 157–168.

[13] R.K. Paliwal, B.S. Atal, Efficient vector quantization of LPC parameters at 24 bits/frame, IEEE Trans. Acoust. Speech Signal Process. 1 (January 1993) 3–14.

[14] R. Laroia, N. Phamdo, N. Farvardin, Robust and efficient quantization of speech LSP parameters using structured vector quantizers, in: Proc. ICASSP 91, Toronto, Canada, May 1991, pp. 641–644.

[15] W.P. LeBlanc et al, Efficient search and design procedure for robust multi-stage VQ of LPC Parameters for 4 kb/s speech coding, IEEE Trans. Speech Audio Process. 1 (October 1993) 373–385.

[16] K.T. Malone, T.R. Fischer, Enumeration and trellis-searched coding schemes for speech LSP parameters, IEEE Trans. Speech and Audio Process. 1 (July 1993) pp. 304–314.

[17] A.M. Kondoz, Digital Speech: Coding for Low Bit Rate Communication Systems, Wiley, New York, 1994.

[18] J. Anderson, J. Bodie, Least squares quantization in PCM, IEEE Trans. Inform. Theory IT-21 (July 1975) 379–387.

[19] A. Cumani, On a covariance-lattice algorithm for linear prediction, in: Proc. ICASSP 82, Vol. II, pp. 615–654.

[20] Vector Sum Excited Linear Prediction 11200 Bit Per Second Voice Coding Algorithm, Chicago Corporate Research and Development Center, Motorola Inc., USA, 1990.

[21] F. Itakura, Line spectrum representation of linear predictive coefficients of speech signals, J. Acoust. Soc. Amer. 57 (April 1975) s35.

[22] W.M. Fisher, G.R. Dodding, K.M. Goudie-Marshall, The DARPA speech recognition research databases: specifications and status, in: Proc. DARPA Speech Recognition Workshop, Report No. Saic-86/1546, February 1986, pp. 54–60.

[23] A.H. Gray, Jr., J.D. Markel, Distance measures for speech processing, IEEE Trans. Acoust. Speech Signal Process. ASSP-24 (October 1976) 380–391.