AN ALGORITHM FOR VOICE WAVEFORM VQ CODEBOOK DESIGN BASED ON PCA

F. Madeiro, R. M. Vilar, B. G. Aguiar Neto and M. S. Alencar
Laboratório de Automação e Processamento de Sinais – LAPS
Laboratório de Comunicaçõoes – LABCOM
Departamento de Engenharia Elétrica
Universidade Federal da Paraíba
Av. Aprígio Veloso, 882, Bodocongó, 58109-970, Campina Grande - PB - Brazil
{madeiro,rosa,bganeto,malencar}@dee.ufpb.br

Abstract - Speech compression plays an important role in applications which require the minimization of the storage and/or transmission requirements, such as multimedia, integrated services digital networks (ISDN), voice response systems and cellular telephony. Vector quantization (VQ) is a well-known compression technique which has been widely used in many speech coding systems. In the present work, an algorithm for designing codebooks for voice waveform vector quantization is presented. Simulations are carried out to compare the performance of the proposed algorithm, which is based on principal component analysis (PCA), to the performance of two other design approaches: the traditional LBG (Linde-Buzo-Gray) algorithm and a self-organizing neural network algorithm.

Resumo - A compressão de voz desempenha um papel importante em aplicações que necessitam minimização dos requisitos de armazenamento e/ou transmissão, tais como: multimídia, redes digitais de serviços integrados (ISDN), sistemas de resposta vocal e telefonia celular. Neste contexto, a quantização vetorial (VQ) apresenta-se como uma poderosa técnica de compressão, que tem sido bastante utilizada em diversos sistemas de codificação de voz. No presente trabalho, é apresentado um algoritmo para projeto de dicionários para quantização vetorial de forma de onda de voz. Diversas simulações realizadas apresentam uma comparação do algoritmo proposto, o qual é baseado em análise de componentes principais (PCA), com duas outras técnicas de projeto de dicionários: o tradicional algoritmo LBG (Linde-Buzo-Gray) e um algoritmo de redes neurais auto-organizativo.

Keywords: Vector quantization, codebook design, voice waveform coding, principal component analysis, Karhunen-Loève transform.

1. INTRODUCTION

The digital representation of speech and image signals allows the application of many techniques of digital processing, which assure an adequate use of the available channel capacity. Many advantages may arise from the digital representation, such as the efficient control of channel errors, the application of many techniques of cryptography and the possibility of integration of signals and data. The current technological impositions require high compression rates for representing speech and image signals. Vector quantization (VQ) has been widely used in many speech and image coding systems [1, 2], due to the theoretically proved fact that it results in a lower distortion for a given rate as compared to scalar quantization [3].

Speech coding is essential for applications such as voice messaging, multimedia, teleconferencing, integrated services digital networks (ISDN), voice response systems, wireless communications, digital telephone answering machines and security devices [4]. In these applications, the fundamental purpose of speech compression techniques is to reduce the number of bits required to represent the speech signal, while maintaining a desired level of signal quality.

Speech coding techniques may be classified into two general categories [5, 6]: waveform coders and voice coders (*vocoders*), also known as parametric coders. The purpose of waveform coding speech compression systems is the efficient representation of the actual speech waveform; the coding process is directly carried out in the samples of the speech signal. On the other hand, in parametric coders, which are based on a model of the human voice production mechanism, the coding process is carried out in the parameters that describe such model.

In the present paper, an algorithm for designing codebooks for voice waveform vector quantization is presented. It is based on Principal Component Analysis (PCA). Unlike the LBG (Linde-Buzo-Gray) algorithm and self-organizing neural networks algorithms, the proposed algorithm does not need a training sequence to update the codevectors. It consists on the use of the eigenvalues and the eigenvectors of the covariance matrix of a typical sequence of speech data to compute the codevectors.

The remainder of this paper is organized as follows. Section 2. presents a brief description of vector quantization. Some aspects of principal component analysis, with focus on the eigendecomposition provided by the Karhunen-Loève Transform (KLT), are provided in Section 3.. Section 4. points out the main differences between the role of the KLT on the proposed method for voice waveform VQ codebook design and the conventional application of the KLT in the scenario of transform coding. The proposed algorithm is described in Section 5.. Simulation results are provided in Section 6. and the concluding remarks are given in Section 7..

2. VECTOR QUANTIZATION

Vector quantization [1, 2], also called block quantization or multidimensional quantization, has been established as an effective source coding technique that plays an important role in many speech and image compression systems. It is worth mentioning that according to Shannon's rate-distortion theory [3], a better performance is always achievable in theory by coding blocks of samples (that is, vectors) instead of coding each sample individually (that is, scalars). In other words, that theory states the superiority of vector quantization over scalar quantization.

An N-level vector quantizer of dimension K can be defined as a mapping Q from a vector $\mathbf{x}$ in K-dimensional Euclidean space, R^K, into a a finite subset W of R^K containing N distinct reproduction vectors. Thus,

$$Q : R^K \rightarrow W, \tag{1}$$

where the codebook $W = \{\mathbf{w}_i;\ i = 1,\ 2,\ \ldots,\ N\}$ is the set of K-dimensional codevectors (reconstruction vectors, template vectors, quantization vectors).

The mapping Q leads to a partition of R^K into N subspaces S_i, $i = 1,\ 2,\ \ldots,\ N$, for which:

$$\bigcup_{i=1}^{N} S_i = R^K \text{ and } S_i \cap S_j = \emptyset \text{ if } i \neq j, \tag{2}$$

where each cell or region S_i is defined such as

$$S_i = \{\mathbf{x} : Q(\mathbf{x}) = \mathbf{w}_i\}. \tag{3}$$

According to [2], the i-th cell S_i is sometimes called the inverse image or pre-image of $\mathbf{w}_i$ under the mapping Q and denoted more concisely by $S_i = Q^{-1}(\mathbf{w}_i)$.

Since the Voronoi cell S_i collects together all input vectors mapping to the i-th codevector, as depicted in Figure 1, the codevector (prototype vector) $\mathbf{w}_i$ may be viewed as a pattern-class label (prototype pattern) of the input vectors (input patterns) belonging to S_i [7]. Accordingly, vector quantization may be viewed as a form of pattern recognition where an input pattern is "approximated" by one of a predetermined set (codebook) of standard patterns [2].

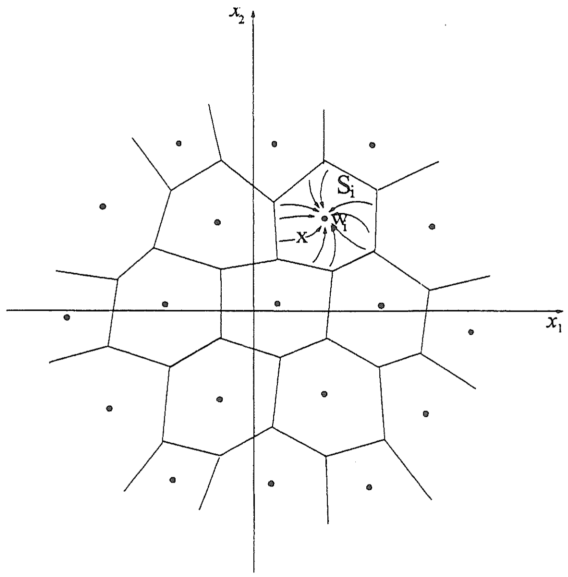

Figure 1. Partition of the two-dimensional Euclidean space, R^2, introduced by the mapping of input vectors $\mathbf{x}$ into representative codevectors $\mathbf{w}_i$. Note that x_1 and x_2 represent the first and second components of vector $\mathbf{x} \in R^2$, respectively

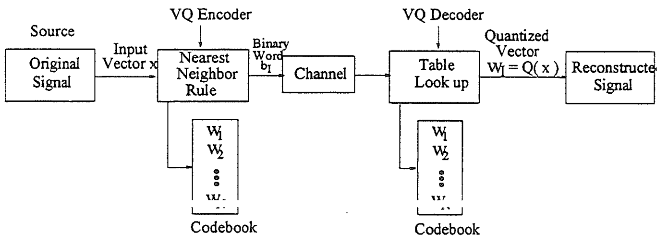

Figure 2. Coding/decoding procedure of a signal compression system based on vector quantization.

Figure 2 shows that a vector quantizer can be considered as a combination of two functions: a VQ encoder and a VQ decoder. The former maps an input vector $\mathbf{x}$ to a codevector $\mathbf{w}_I$ if $d(\mathbf{x}, \mathbf{w}_I) < d(\mathbf{x}, \mathbf{w}_i)$, $\forall i \neq I$, where $d(\cdot)$ is some distortion function. In other words, it follows the nearest neighbor rule to find the codevector that presents the greatest similarity to $\mathbf{x}$. Then, the $\lceil \log_2 N \rceil$ binary representation of the index I, denoted by b_I, is transmitted to the VQ decoder [8]. Upon receiving the binary word b_I, the VQ decoder simply looks up the I-th codevector, $\mathbf{w}_I$, from a copy of the codebook W, and outputs $\mathbf{w}_I$ as the reproduction of $\mathbf{x}$. As shown in Figure 2, the mapping of $\mathbf{x}$ into $\mathbf{w}_I$ is captured by the expression $\mathbf{w}_I = Q(\mathbf{x})$ [9].

Therefore, vector quantization is a lossy compression technique – the reconstructed signal is a degraded version of the original signal. The mean error obtained in representing the input signal by its corresponding quantized version is called the distortion of the quantizer. On the other hand, the code rate of the vector quantizer, which measures the number of bits per vector component, is $R = \frac{1}{K} \log_2 N$. In voice waveform coding [10], R is expressed in bit/sample. In image coding [8], R is expressed in bits per pixel (bpp). An important issue regarding vector quantization is the compromise between rate and distortion.

Codebook design plays a crucial role in the scenario of signal compression systems based on vector quantization. Techniques for codebook design attempt to produce a codebook that is optimum for a given source in the sense that the average distortion in representing the input vectors by the corresponding codevectors may be kept to a minimum.

In the literature, a number of techniques for codebook design has been proposed and evaluated, such as the traditional LBG algorithm [11], the pair-wise nearest neighbor (PNN) algorithm [12], fuzzy algorithms [13], stochastic relaxation [14], the Kohonen learning algorithm [15] and other self-organizing neural networks algorithms [16, 17, 18].

The most widely used technique for designing VQ

It is worth mentioning that the KLT diagonalizes the correlation matrix $\mathbf{R_{ff}}$, i. e., the transform coefficients are uncorrelated [24]. In fact,

$$\mathbf{R_{ff}} = E[\mathbf{ff}^T] = E[\mathbf{Zxx}^T\mathbf{Z}^T] = \mathbf{ZR_{xx}Z}^T. \quad (14)$$

For zero-mean processes $\mathbf{R_{xx}} = \mathbf{C_{xx}}$, therefore,

$$\mathbf{R_{ff}} = \mathbf{ZC}_{xx}\mathbf{Z}^T. \quad (15)$$

From Equation (10),

$$\mathbf{R_{ff}} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_K^T \end{bmatrix} \begin{bmatrix} \lambda_1\mathbf{z}_1 & \lambda_2\mathbf{z}_2 & \cdots & \lambda_K\mathbf{z}_K \end{bmatrix}. \quad (16)$$

Hence, from Equation (5),

$$\mathbf{R_{ff}} = \begin{bmatrix} \lambda_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & & \cdots & . \\ 0 & & \lambda_3 & \cdots & . \\ \vdots & & & & \\ 0 & . & . & \ldots & \lambda_K \end{bmatrix}. \quad (17)$$

Thus, the Karhunen-Loève transform, also known as the Hotelling transform, is an orthogonal transformation which reduces a large set of correlated variables to a smaller set of uncorrelated components [24, 25].

4. USING PCA TO DESIGN CODEBOOKS

Transform techniques have been widely applied to voice and image coding systems, leading to good results. In transform coding, the KLT is the optimum choice for obtaining uncorrelated coefficients, prior to individual scalar quantization of each feature [24]. The problem with this approach is that the KLT must be computed in real time, for each incoming input data vector.

The method proposed in this paper differs from the above mentioned method. The eigenvectors (principal components) are used to find $L \leq K$ directions (in the R^K space) along which the codebook vectors (codevectors) are to be allocated. The eigenvalues corresponding to those principal directions are used to properly adjust the positions of the codevectors along the principal directions. Besides, real time calculation of the KLT is not used. Unlike transform coding, which calculates the KLT for each input vector, the proposed method computes the eigenvalues and the eigenvectors of the covariance matrix of an entire typical speech sequence, that is, the principal components are calculated only once and they are necessary only when designing the codebook, prior to its use in the coding process.

It is worth mentioning that, according to [24], the KLT basis vectors are appropriate for long-term speech statistics. The KLT basis vectors exhibit the greatest similarity to waveform segments that are typical of the input class whose statistics have been used in the KLT derivation. This in turn is reflected by the dimensionality needed for a given representation error to be the smallest when the representation is in term of KLT basis vectors (eigenvectors, principal components). In fact, since these basis vectors are signal-dependent, fewer of them can be used on the average to approximate a given input, as compared to representations using signal-dependent basis vectors.

5. ALGORITHM DESCRIPTION

The proposed algorithm is based on principal component analysis of a typical speech sequence and consists on the following steps:

1. Define dimension K and codebook size N;
2. From the $K \times K$ covariance matrix $\mathbf{C_{xx}}$ of a typical speech sequence $\mathbf{X}$, find the eigenvalues λ_i and the eigenvectors $\mathbf{z}_i$; $i = 1, 2, ..., K$, defined by:

$$\mathbf{C_{xx}z}_i = \lambda_i\mathbf{z}_i; \quad (18)$$

3. Define $L \leq K$ vectors $\mathbf{z}_i$ (principal components) along the directions the codevectors are to be allocated. The number L is chosen according to the relative percent value $\lambda_{i,p}$ of each eigenvalue λ_i:

$$\lambda_{i,p} = \frac{\lambda_i}{\sum_{i=1}^{K} \lambda_i}. \quad (19)$$

Only the L most significant eigenvectors, which define the L principal directions, are chosen, according to the most significant $\lambda_{i,p}$;

4. For each $\mathbf{z}_i$ chosen, determine a vector $\widetilde{\mathbf{z}}_i = r\mathbf{z}_i$ ($i = 1, 2, ..., L$), where the scalar r is the reciprocal of the absolute value of the component with the largest absolute value. Therefore $\widetilde{\mathbf{z}}_i$ has, at least, one component with absolute value 1 and the other absolute values range from 0 to 1;
5. Let N_i be the number of codevectors to be allocated along the direction defined by vector $\widetilde{\mathbf{z}}_i$. Each N_i is chosen as a fraction of N, in proportion to $\lambda_{i,p}$ so that

$$N = \sum_{i=1}^{L} N_i; \quad (20)$$

6. Finally, N_i codevectors $\mathbf{w}_{i,n_i}$ are allocated to each i-th principal direction, according to:

$$\mathbf{w}_{i,n_i} = f(n_i, \lambda_i)\widetilde{\mathbf{z}}_i, \quad (21)$$

where $i = 1, 2, ..., L$, $n_i = 1, 2, ..., N_i$ and the scalars $f(n_i, \lambda_i)$ are determined assuming that each i-th principal direction is related to a set of input vectors whose components have Gaussian distribution with mean equal to to the mean of the speech sequence and variance ${\sigma_i}^2 = \lambda_i$. Scalars $f(n_i, \lambda_i)$, $n_i = 1, 2, ..., N_i$, are determined in such a way that the area under the Gaussian probability density function is equally divided into N_i intervals.

In summary, a set of L eigenvectors is found and a number of codevectors N_i (a fraction of N) is then allocated to each principal direction defined by eigenvectors $\mathbf{z}_i$ $(i = 1, 2, ..., L)$ according to considerations on eigenvalues (variances).

The proposed algorithm (from now on referred to as PCA) differs from that one reported in [26] by the introduction of normalization in the set of eigenvectors, as described in step 4. The multiplication of the normalized vectors $\tilde{\mathbf{z}}_i$ by the scalars $f(n_i, \lambda_i)$ permits the resulting codevectors to incorporate, simultaneously, two important informations of the speech signal: the principal directions (via eigenvectors) and the statistical distribution of vectors (via eigenvalues) in each direction.

6. RESULTS

Voice waveform vector quantizers were designed and tested for a variety of dimensions (K) and number of levels (N). The original speech signals were recorded at a coding rate of 8.0 bit/sample. The sample rate used was 8 kHz. Simulations were carried out to compare the performance of the PCA algorithm to other design procedures: the traditional LBG algorithm and an unsupervised learning neural network algorithm, referred to as SOA (self-organizing algorithm) [18].

The first simulation was carried out to evaluate the coherence of the proposed method. Figure 3 shows a graphical representation of a typical speech signal, which consists of 10 phonetically balanced sentences [27] uttered by 10 different speakers (5 males and 5 females). Figure 4 shows the codebook designed using PCA algorithm. It is easy to observe that the proposed algorithm was able to build representative patterns inside the trained codebook, which is well adapted to the signal statistics. The final codebook configuration has incorporated some typical properties of the speech signal such as predominance of small amplitude components of speech (more codevectors are allocated to regions of greater incidence of speech signal samples) and typical correlation among consecutive samples, shown by the proper positioning of codevectors along the principal component direction.

The second simulation concerned the performance evaluation of the proposed algorithm at different bit/sample rates for dimension $K = 2$. The designed codebooks were tested using a relatively long training sequence, the Portuguese utterance: "O sol ilumina a fachada de tarde. Trabalhou mais do que podia" (29120 samples, 3.64s). The resulting compressed speech signals were evaluated by the segmental signal-to-noise ratio (SNRseg), which is obtained by averaging the conventionally defined signal-to-noise ratio (SNR) over short time intervals [6, 24]. Figure 5 indicates that, for coding rates ranging from 1.7 bit/sample to 2.7 bit/sample, PCA performs better than the LBG algorithm. For coding rates ranging from 2.0 bit/sample to 2.5 bit/sample, the performance of PCA is similar to that of the SOA algorithm. However, both LBG and SOA algorithms outperform PCA in the range 2.7-3.5 bit/sample.

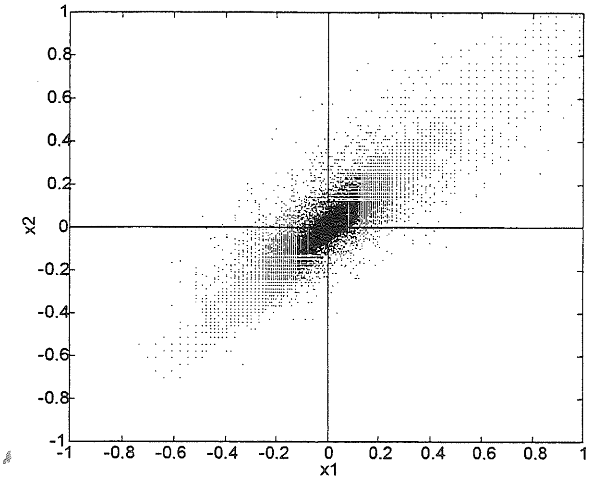

Figure 3. Speech signal used in the determination of the covariance matrix, which consists of 10 phonetically balanced utterances (18.76s, 75040 vectors); x_1 and x_2 represent the first and second components of vector $\mathrm{x} \in R^2$, respectively.

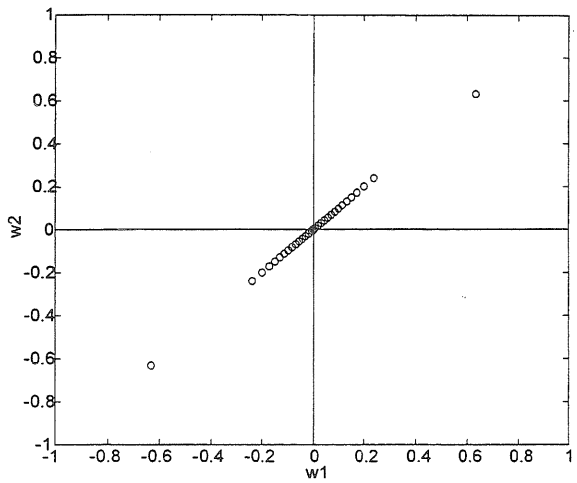

Figure 4. Codebook designed using the PCA algorithm ($K = 2$ and $N = 32$); w_1 and w_2 represent the first the second components of codevector $\mathrm{w} \in R^2$, respectively.

Another simulation consisted of designing vector quantizers at different bit/sample rates for vector dimension $K = 4$. As can be observed on Figure 6, PCA outperforms LBG in the range 0.75-1.35 bit/sample. For rates up to 1.25 bit/sample, the performance of PCA is fairly close to that of SOA. For coding rates higher than 1.35 bit/sample, both LBG and SOA algorithms outperform PCA. It is important to mention that the SNRseg values of the

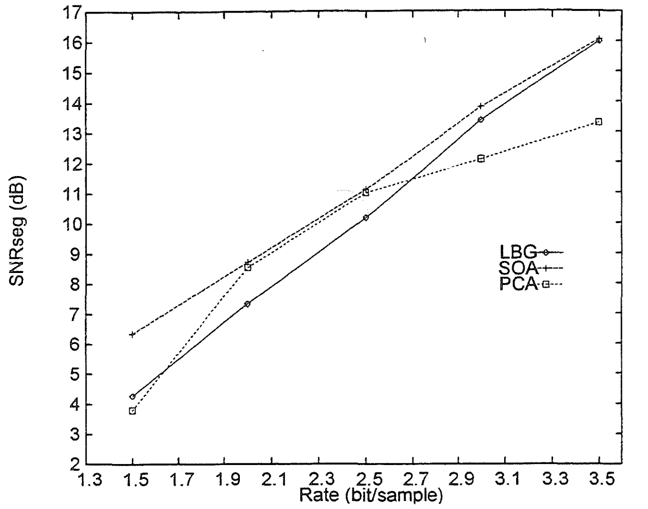

Figure 5. Performance evaluation of LBG, SOA and PCA algorithms in speech compression: segmental signal-to-noise ratio (SNRseg) versus bit/sample rate for dimension $K = 2$.

LBG algorithm in Figure 6 correspond to the best values obtained by applying different initialization strategies of the LBG algorithm, using a very small distortion threshold. Accordingly, the SNRseg values differs from those presented in [28].

It is important to emphasize that unlike LBG and SOA algorithms, the proposed algorithm does not require a training process. It simply "calculates" the codebook according to the eigenvalues and the eigenvectors of the covariance matrix of a representative speech signal. Additionally, unlike SOA and other unsupervised neural networks algorithms, the PCA algorithm has no parameters to be adjusted.

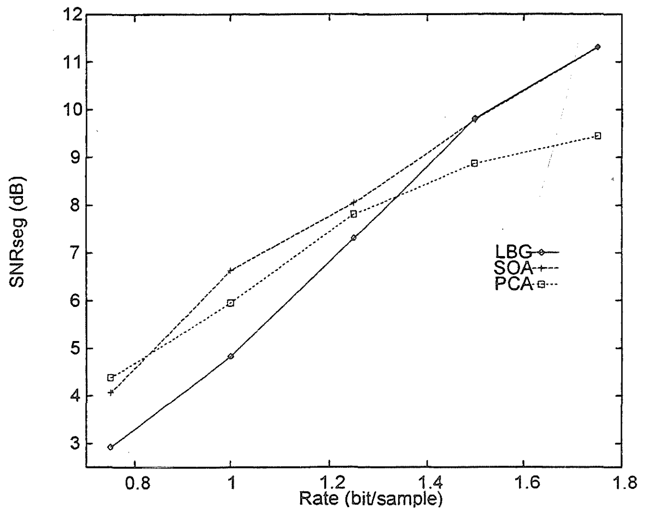

Figure 6. Performance evaluation of LBG, SOA and PCA algorithms in speech compression: segmental signal-to-noise ratio (SNRseg) versus bit/sample rate for dimension $K = 4$.

Concerning informal subjective evaluations of the speech material, it was observed, for all considered coding rates of the 2-dimensional vector quantization, that the reconstructed speech signals obtained by using PCA codebooks lead to a weaker perception of the broadband background noise (quantization noise) when compared to the reconstructed signals obtained by using the LBG and the SOA codebooks. On the other hand, for some coding rates of the 4-dimensional vector quantization, the PCA codebooks made the reconstructed speech signal be affected by the presence of an artificial signal, like a "musical" tone, which is more easily perceived in the silence segments and does not present any similarity with the speech signal or the quantization noise. This effect, denoted in the literature as *musical noise* or *tonal noise* [29, 30] is produced when the spectral components on the regions of higher concentration of small amplitude samples are not adequately reconstructed, as occurs in the background noise on the silence segments. This phenomenon may appear as "spectral-gaps" in the spectrum of the reconstructed signal [31, 32]. In particular, the authors have observed the occurrence of non typical patterns in the spectrogram of the reconstructed signal. However, it is important to emphasize that, despite the introduction of some *musical noise*, the informal subjective tests indicated that the PCA codebooks lead to a broadband background noise (quantization noise) which is less annoying than that one obtained by using both SOA and LBG codebooks. This result from the subjective evaluation was confirmed through objective evaluation. In fact, Table 6. shows an interesting aspect of the PCA algorithm. Unlike what typically occurs when using SOA and LBG codebooks, the segmental signal-to-noise ratio (SNRseg) of the reconstructed signal obtained using PCA codebooks is higher than the conventionally defined signal-to-noise ratio (SNR). This behavior was observed for almost all coding rates and clearly indicates that the PCA codebooks preserve the low energy speech segments as compared to LBG and SOA codebooks.

K	N	SNRseg (dB)	SNR (dB)
2	32	11.01	8.82
2	64	12.11	9.71
2	128	13.29	10.67
4	32	7.81	6.51
4	64	8.87	7.24
4	128	9.44	7.66

Table 1. Segmental signal-to-noise ratio (SNRseg) and the conventionally defined signal-to-noise ratio (SNR) for the reconstructed signals obtained by using PCA codebooks for different values of dimension (K) and number of levels (N).

Simulations were also carried out to evaluate the PCA codebooks as the starting point of the LBG algorithm. Table 6. shows that the PCA codebooks may be used as a good alternative to the initialization of the LBG algorithm since they may lead to a significant improvement in the convergence speed of the LBG algorithm. Using the PCA codebooks as the initial codebooks of the LBG algorithm may result in a considerable decrease in terms of number of iterations of the LBG codebook design method. As an example, for designing a codebook with $K = 4$ and $N = 128$, the application of the PCA initialization strategy under consideration (in other words, using C_{PCA} as the initial codebook) leads to 23 iterations, which represents a lower computational complexity as compared to initialization C_{II}, that requires 82 iterations. Table 6. also shows that, for a fixed dimension K, the gain in terms of convergence speed introduced by initialization C_{PCA} over other initialization strategies seems to increase as the codebook size N increases. It was also observed that using PCA codebooks as the initialization of LBG may lead to a gain in terms of SNRseg when compared to other choices of initial codebooks.

K	N	C_I	C_{II}	C_{III}	C_{PCA}
4	8	21	16	32	22
4	16	33	44	45	34
4	32	59	62	52	53
4	64	72	85	59	39
4	128	77	82	39	23

Table 2. Sensibility of LBG algorithm to four different initial codebooks (C_I, C_{II}, C_{III} and C_{PCA}) in terms of number of iterations for different values of codebook size N for dimension $K = 4$. The subscript PCA denotes the initialization using the PCA codebooks.

An interesting aspect of the PCA codebooks regards the uniformity of the distribution of the source vectors (input vectors) in Voronoi cells. The values of normalized entropy (see Appendix for more details) in Tables 6. and 6. show that the PCA codebooks lead to a more uniform distribution of source vectors in Voronoi regions as compared to SOA codebooks. In fact, the PCA codebooks present higher values of normalized entropy. This behavior favors the PCA based voice waveform VQ codebook design algorithm.

K	N	H_n
2	32	0.94
2	64	0.93
4	32	0.94
4	64	0.95
8	32	0.96
8	64	0.95

Table 3. Normalized entropy (H_n) of the codevectors for different values of codebook size (N) and dimension (K) for the PCA codebooks.

K	N	H_n
2	32	0.88
2	64	0.91
4	32	0.87
4	64	0.88
8	32	0.84
8	64	0.89

Table 4. Normalized entropy (H_n) of the codevectors for different values of codebook size (N) and dimension (K) for the SOA codebooks.

Figure 7 shows that the proposed method generates structured codebooks, unlike both LBG and SOA algorithms, which generate codebooks without structure, as can be seen in Figure 8. It is worth mentioning that structured codebooks may be used as an interesting attempt to circumvent the complexity problem of the vector quantization encoding phase, that is, the nearest neighbor search [33].

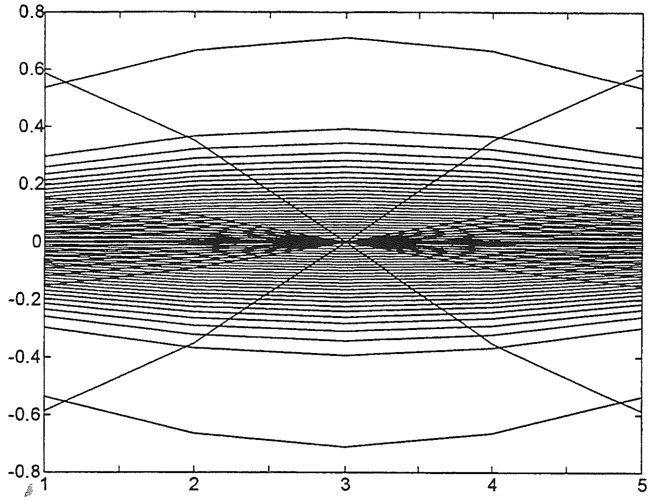

Figure 7. Codebook designed by the PCA algorithm: $K = 5$ and $N = 76$. Each curve in the set of 76 curves is obtained by linking the points corresponding to the components (samples) of the codevectors.

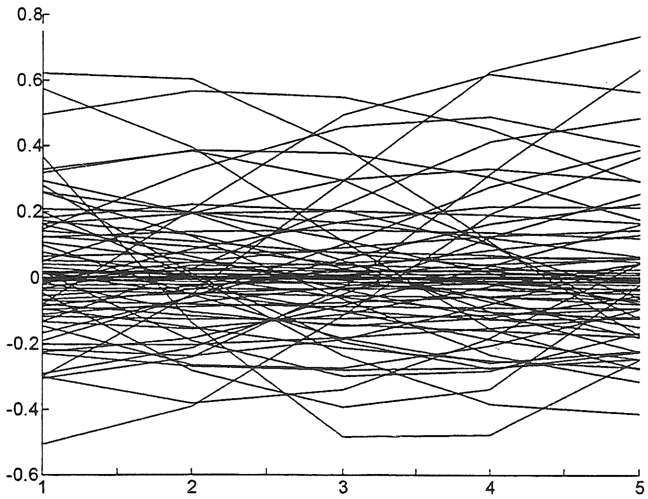

Figure 8. Codebook designed by the SOA algorithm: $K = 5$ and $N = 76$. Each curve in the set of 76 curves is obtained by linking the points corresponding to the components (samples) of the codevectors.

7. CONCLUSIONS

A PCA based algorithm for designing voice waveform VQ codebooks was presented. Unlike other design techniques, such as the LBG algorithm, the Kohonen learning algorithm and other self-organizing neural networks algorithms, the proposed technique does not need a training sequence to iteratively update the codevectors, since the codebook vectors are calculated in a very simple and intuitive manner. Basically the proposed method performs an eigendecomposition on the covariance matrix of a representative speech sequence, orders the eigenvectors (principal components) according to the descending order of the corresponding eigenvalues, chooses the most significant eigenvectors and allocates (along the direction defined by each chosen eigenvector) a number of codevectors according to considerations on the eigenvalues. Additionally, the PCA based algorithm, unlike Kohonen's self-organizing algorithm and other unsupervised learning algorithms, has no parameters to be adjusted. Since the PCA based method does not need the definition of an initial codebook, unlike the LBG algorithm, both performance and convergence speed do not depend on an initial codebook.

Simulations results have shown that, for fixed dimensions $K = 2$ and $K = 4$, at a wide range of bit/sample rates evaluated, the PCA based algorithm performs better than the traditional LBG and close to the performance presented by an unsupervised learning neural network algorithm, referred to as SOA (self-organizing algorithm). Regarding informal subjective evaluations of the speech material, it was observed that the reconstructed speech signals obtained by using PCA codebooks lead to a weaker perception of the broadband background noise (quantization noise) when compared to the reconstructed signals obtained by using the LBG and the SOA codebooks. Unlike what typically occurs when using SOA and LBG codebooks, the segmental signal-to-noise ratio of the reconstructed signal by using PCA codebooks is higher than the conventionally defined signal-to-noise ratio. This behavior clearly indicates that the PCA codebooks preserve the low energy speech segments as compared to the LBG and the SOA codebooks.

Through a set of simulations, it was observed that the PCA codebooks may be used as a good alternative to the initialization of the LBG algorithm since they may lead to a significant improvement in the convergence speed of the LBG algorithm.

It was shown that the PCA based algorithm generates structured codebooks. The authors also observed that the codebooks designed by the proposed algorithm incorporate important characteristics of speech signals, such as predominance of codevectors with small amplitude components and typical correlation among consecutive samples of the codevectors. This happens according to the allocation of the codevectors to the principal components of the speech signal, due to information extracted from the eigenvectors and eigenvalues of the covariance matrix of the speech sequence. Therefore, the proposed algorithm is simple, intuitive and presents a good performance.

ACKNOWLEDGMENTS

The authors would like to express their thanks to CAPES and CNPq for the financial support to the research.

NORMALIZED ENTROPY

In the present section, the normalized entropy [34] of the codevectors is described. It serves as a measure for assessing the uniformity of the source vectors distribution along the Voronoi cells.

Let $W = \{\mathbf{w}_i \mid i = 1, 2, ..., N\}$ be a N-level codebook, where $\mathbf{w}_i$ is a K-dimensional codevector. Let p_i be the probability that a given source vector belongs to the Voronoi region or cell corresponding to the codevector $\mathbf{w}_i$. In other words, p_i corresponds to the probability that $\mathbf{w}_i$ be the nearest codevector for a given input vector $\mathbf{x}$.

Let $S = \{\mathbf{x}_s,\ s = 1, ..., n\}$ be a long representative training sequence, that is, S is a large sequence of representative vectors which are used in the codebook design algorithm. The sequence S may also be a large representative sequence of source vectors to be coded (i.e., to be vector-quantized) by using a previously designed codebook. According to the mapping introduced by vector quantization, the set S is partitioned into N non-intercepting subsets or Voronoi cells S_i, $i = 1, ..., N$, where each cluster S_i collects together all source vectors mapping into the i-th codevector, that is, $S_i = \{\mathbf{x}_s : Q(\mathbf{x}_s) = \mathbf{w}_i\}$. In other words, $\mathbf{w}_i$ is the nearest codevector to all $\mathbf{x}_s \in S_i$.

Let n_i denote the size of the subset S_i, that is, n_i is the number of input vectors mapping to the i-th codevector. The probability that $\mathbf{w}_i$ be the nearest codevector to any $\mathbf{x}_s$ (which is the probability that S_i be the Voronoi cell of $\mathbf{x}_s$) may be obtained as

$$p_i = \frac{n_i}{n}. \tag{22}$$

The entropy H_K of the codevectors is defined as

$$H_K = \sum_{i=1}^{N} p_i \log_2\left(\frac{1}{p_i}\right). \tag{23}$$

The normalized entropy H_n of the codevectors is defined as

$$H_n = \frac{H_K}{\log_2 N}, \tag{24}$$

that is,

$$H_n = \frac{\sum_{i=1}^{N} p_i \log_2\left(\frac{1}{p_i}\right)}{\log_2 N} \tag{25}$$

The maximum normalized entropy occurs for equiprobable codevectors. In fact, equiprobability implies $H_K = \log_2 N$. As a consequence, $H_n = 1$. It is worth mentioning that $H_n \to 1$ as the uniformity in the distribution of input vectors along the Voronoi cells increases, that is, $H_n \to 1$ as $H_K \to \log_2 N$. On the other hand, the normalized entropy decreases as the number of small Voronoi cells increases.

REFERENCES

[1] Gray, R. M. Vector Quantization. *IEEE ASSP Magazine*, pages 4–29, April 1984.

[2] Gersho, A. and Gray, R. M. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, MA, 1992.

[3] Berger, T. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[4] Rabiner, L. R. Applications of Voice Processing to Telecommunications. *Proceedings of the IEEE, Vol. 82, No. 2*, pages 199–228, February 1994.

[5] Jayant, N. S. Coding Speech at Low Bit Rates. *IEEE Spectrum*, pages 58–63, August 1986.

[6] Deller Jr., J. R., Proakis, J. G. and Hansen, J. H. L. *Discrete-time Processing of Speech Signals*. Macmillan Publishing Co., 1993.

[7] Kosko, B. *Neural Networks and Fuzzy Systems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1992.

[8] Ramamurthi, B. and Gersho, A. Classified Vector Quantization of Images. *IEEE Transactions on Communications, Vol. COM-34, No. 11*, pages 1105–1115, November 1986.

[9] Moaveri, N., Neuhoff, D. L., and Stark, W. E. Fine-Coarse Vector Quantization. *IEEE Transactions on Signal Processing, Vol. 39, No. 7*, pages 1503–1515, July 1991.

[10] Abut, H., Gray, R. M. and Rebolledo, G. Vector Quantization of Speech and Speech-Like Waveforms. *IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-30, No. 3*, pages 423–435, June 1982.

[11] Linde, Y., Buzo, A. and Gray, R. M. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications, Vol. COM - 28, No. 1*, pages 84–95, January 1980.

[12] Equitz, W. H. A New Vector Quantization Clustering Algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 37, No. 10*, pages 1568–1575, October 1989.

[13] Karayiannis, N. B. and Pai, P.-I. Fuzzy Vector Quantization Algorithms and Their Applications in Image Compression. *IEEE Transactions on Image Processing, Vol. 4, No. 9*, pages 1193–1201, September 1995.

[14] Zeger, K., Vaisey, J. and Gersho, A. Globally Optimal Vector Quantizer Design by Stochastic Relaxation. *IEEE Transactions on Signal Processing, Vol. 40, No. 2*, pages 310–322, February 1992.

[15] Kohonen, T. *Self-Organization and Associative Memory (3rd ed)*. Springer-Verlag, Berlin, 1989.

[16] Vilar França, R. M. and Aguiar Neto, B. G. Voice Waveform Vector Quantization Using a Competitive Algorithm. *"Records of the IEEE GLOBECOM'94"*, pages 872–875, November 1994.

[17] França, R. M., Madeiro, F. and Aguiar Neto, B. G. Building Voice Patterns Inside Vector Quantization Codebooks with a Modified Kohonen's Algorithm. *Proceedings of IEEE ICSP'97*, pages 291–295, August 1997.

[18] Madeiro, F., Vilar, R. M., Fechine, J. M. and Aguiar Neto, B. G. A Self-Organizing Algorithm for Vector Quantizer Design Applied to Signal Processing. *International Journal of Neural Systems, Vol. 9, No. 3, Special Issue on the Vth Brazilian Symposium on Neural Networks*, pages 219–226, June 1999.

[19] Lee, D., Baek, S. and Sung, K. Modified K-means Algorithm for Vector Quantizer Design. *IEEE Signal Processing Letters, Vol. 4, No. 1*, pages 2–4, January 1997.

[20] Xu, L. and Yuille, A. L. Robust Principal Componente Analysis by Self-Organizing Rules Based on Statistical Physics Approach. *IEEE Transactions on Neural Networks, Vol. 6, No. 1*, pages 131–143, January 1995.

[21] Diamantaras, K. J. & Kung, J. S. *Principal Component Neural Networks - Theory and Applications*. John Wiley & Sons, New York - NY, 1996.

[22] Fukunaga, J. *Statistical Pattern Recognition*. Academic Press, New York, 1990.

[23] Haykin, S. *Neural Networks - A Comprehensive Foundation*. IEEE Press, Englewood Cliffs - NJ, 1994.

[24] Jayant, N. S. and Noll, P. *Digital Coding of Waveforms*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[25] Chen, C. S. and Huo, K.-S. Karhunen-Loeve Method for Data Compression and Speech Synthesis. *IEE Proceedings - I, Vol. 138, No. 5*, pages 377–380, October 1991.

[26] Vilar França, R. M. and Aguiar Neto, B. G. Designing Codebooks for Voice Waveform Vector Quantization Based on the Karhunen-Loève Transform. *"Records of the IEEE International Telecommunications Symposium ITS'96"*, pages 44–48, October 1996.

[27] Alcaim, A., Solewicz, J. A. and Moraes, J. A. Freqüência de Ocorrência dos Fones e Listas de Frases Foneticamente Balanceadas no Português Falado no Rio de Janeiro. *Revista da Sociedade Brasileira de Telecomunicações, Vol. 7, No. 1*, pages 23–41, December 1992.

[28] Madeiro, F., Vilar, R. M., Aguiar Neto, B. G. and Alencar, M. S. Voice Waveform VQ Codebook Design Based on PCA. *Proceedings of XVII Simpósio Brasileiro de Telecomunicações (SBT'99)*, pages 115–120, September 1999, Vila Velha, Espírito Santo, Brazil.

[29] Sondhy, M. M., Schmidt, C. E. and Rabiner, L. R. Improving the Quality of a Noise Speech Signal. *Bell System Technical Journal, Vol. 60, No. 6*, pages 1847–1859, October 1981.

[30] Dendrinos, M., Bakamidis, S. and Carayannis, G. Speech Enhancement from Noise: A Regenerative Approach. *Speech Communication, Vol. 10, No. 1*, pages 45–57, February 1991.

[31] Aguiar Neto, B. G. Signalaufbereitung in Digitalen Sprachsignal Übertragungssystemen. *Doctor thesis. Technische Universitaet Berlin - Germany*, November 1987.

[32] Vary, P., Heute, U. and Hess, W. *Digitale Sprachsignalverarbeitung*. Teubner Verlag, Stuttgart, 1998.

[33] Gersho, A. and Cuperman, V. Vector Quantization: A Pattern-Matching Techinique for Speech Coding. *IEEE Communications Magazine*, pages 15–20, December 1983.

[34] Paliwal, K. K. and Ramasubramanian, V. Effect of Ordering the Codebook on the Efficiency of the Partial Distance Search Algorithm for Vector Quantization. *IEEE Transactions on Communications, Vol. 37, No. 5*, pages 538–540, May 1989.

F. Madeiro was born in Fortaleza, Ceará, Brazil, on March 27, 1972. He received his Bachelor Degree and his Master Degree, both in Electrical Engineering, from Universidade Federal da Paraíba (UFPB), Brazil, in 1995 and 1998, respectively. He is now working towards his Ph.D. degree at UFPB. His research activities focus on speech and image coding, with main interests on techniques involving vector quantization, neural networks and wavelets. Francisco Madeiro is a member of the Brazilian Telecommunications Society (SBrT).

R. M. Vilar has worked as a lecturer in Digital Communications at the Department of Electrical Engineering of Universidade Federal da Paraíba (UFPB) in Campina Grande, Brazil, for 22 years. During that time she has developed research in the areas of channel and source encoding, vector quantization and neural networks applied to voice digitalization. Rosângela Maria Vilar has recently left the University to join The Institute for Entrepreneurship and Innovation (IEMPI) where she is currently involved with training entrepreneurs in technological areas. Her main professional activities at the moment are Technology Transference, Internet Businesses and Knowledge Management. She is also a Director for the SOLUTION Info, an information providing service for Internet.

B. G. Aguiar Neto was born in São Luís, Maranhão, Brazil, in 1954. He is Full Professor of Universidade Federal da Paraíba (UFPB) and since 1997 he is Dean of the Center of Science and Technology (CCT) of UFPB. He received his Bachelor Degree in Electrical Engineering and his Master Degree in Electrical Engineering from UFPB in 1977 and 1982, respectively, and his Dr.-Ing. degree in Electrical Engineering, in the area of Speech Processing, from *Technische Universitaet Berlin*, Germany, in 1987. At CCT he was Undergraduate Adviser from 1993 to 1996 and Coordinator of the Reengineering Educations Program from 1996 to 1998. Professor Benedito Guimarães Aguiar Neto has been with the Department of Electrical Engineering of UFPB since 1978, were he was Coordinator of the Undergraduate Electrical Engineering Course, Coordinator of the Automation and Signal Processing Laboratory and Member of the Graduate Electrical Course Council. He is presently Tutor of the Special Training Program of the Undergraduate Electrical Engineering Course, member of the University Counsil of UFPB and Editor of "Revista de Ensino de Engenharia", published by the Brazilian Association of Engineering Education (ABENGE). He is involved with teaching and research activities in the area of Digital Processing and Transmission of Speech Signals. He is supervisor of Undergraduate and Graduate students, has supervised 11 Master Dissertations at the Electrical Engineering Graduate Program and published over 50 scientific papers. His research interests include Speech and Image Coding, Speech and Speaker Recognition, Speech Synthesis and Speech Enhancement for Digital Transmission.

M. S. Alencar was born in Serrita, Pernambuco, Bra\-zil in 1957. He received his Bachelor Degree in Electrical Engineering, from Universidade Federal de Pernambuco (UFPE), Brazil, 1980, his Master Degree in Electrical Engineering, from Universidade Federal da Pa\-ra\-íba~(UFPB), Brazil, 1988 and his Ph.D. from University of Waterloo, Department of Electrical and Computer Engineering, Canada, 1993. Dr. Alencar has almost 20 years of engineering experience and is Senior Member of IEEE. He has done consulting for many companies, agencies, and universities, including Telecom Italia Mobile, MCI-Embratel, Telebras, Chesf, CNPq, Finep, UFPE, UFMA, CSU and Telern. He was also awarded several scholarships. He published over 70 engineering and scientific papers, and two books: Princípios de Comunicações, Editora Universitária da UFPB, and Telefonia Digital, Editora Érica Ltda. For the past 16 years he has been working for the Department of Electrical Engineering of the Universidade Federal da Paraíba, where he is Full Professor. Previously, he worked for the State University of Santa Catarina. While on leave, he spent some time working for MCI-Embratel and University of Toronto, as Visiting Professor. Professor Marcelo Sampaio de Alencar has been involved with some IEEE and SBrT activities, including being a member of the Advisory or Technical Program Committee in several editions of the following events: PIMRC, GLOBECOM, ICC, SBT and ITS. He is currently serving on the Board of Directors of SBrT. He has been acting as reviewer for the following journals: IEEE Transactions on Communication, IEEE Transactions on Vehicular Technology, Wireless Personal Communications, published by Kluwer Academic Publishers, Brazilian Transactions on Telecommunications, published by SBrT. Dr. Alencar is member of SBMO, SBPC,SBEB and ABENGE.