Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 63192, 11 pages
doi:10.1155/2007/63192

Research Article

Using Geometrical Properties for Fast Indexation of Gaussian Vector Quantizers

E. A. Vassilieva, D. Krob, and J. M. Steyaert

Laboratoire d'Informatique de l'Ecole Polytechnique (LIX), Ecole Polytechnique, 91128 Palaiseau Cedex, France

Received 2 November 2005; Revised 26 August 2006; Accepted 10 September 2006

Recommended by Satya Dharanipragada

Vector quantization is a classical method used in mobile communications. Each sequence of d samples of the discretized vocal signal is associated to the closest d-dimensional *codevector* of a given set called *codebook*. Only the binary indices of these codevectors (*the codewords*) are transmitted over the channel. Since channels are generally noisy, the codewords received are often slightly different from the codewords sent. In order to minimize the distortion of the original signal due to this noisy transmission, codevectors indexed by one-bit different codewords should have a small mutual Euclidean distance. This paper is devoted to this problem of index assignment of binary codewords to the codevectors. When the vector quantizer has a Gaussian structure, we show that a fast index assignment algorithm based on simple geometrical and combinatorial considerations can improve the SNR at the receiver by 5dB with respect to a purely random assignment. We also show that in the Gaussian case this algorithm outperforms the classical combinatorial approach in the field.

1. INTRODUCTION

Taking into account the constraints of the transmission channel between the base transceiver stations (BTS) and mobile stations (MS), voice is coded in mobile networks with the help of techniques allowing to minimize the quantity of information required for its good reconstitution. Among these techniques one finds *vector quantization*. This method consists of replacing the vector y from $\mathbb{R}^d$, obtained by finite discretization of the input vocal signal, by the element c_i, taken from a set $\mathcal{C} = \{c_0, c_1, \dots, c_{N-1}\}$ of vectors of reference, which is the closest to y. The set $\mathcal{C}$ is called a *codebook* and its elements the *codevectors*. Instead of transmitting the initial discretization y, one transmits a string of 0's and 1's which is the *binary codeword* $b(c_i)$ associated with the codevector c_i of the codebook $\mathcal{C}$ which is the closest to y. Because of some interfering noises on the transmission channel, the string s actually received can be different from $b(c_i)$. The output signal is then c_j such that $b(c_j) = s$ (see Figure 1). In what follows, the mapping b that associates with each codevector c_i a binary word $b(c_i)$ representing a nonnegative integer will be called the *indexation* (or the *index assignment*) of the codebook. We will also refer to $b(c_i)$ as the *index* associated with the codevector c_i.

From the very start, vector quantization showed itself as an extremely efficient data compression system. Indeed, it gave impressive performance results in various speech and image coding situations (see, e.g., [1, 2]). Practical implementations of this technique include *code excited linear prediction* (CELP) algorithm (see [3]), which sets the basis of voice encoding within GSM and CDMA protocols. Considering that speech is an autoregressive process, that is, that each sample is the sum of a linearly predictable part and an innovation part, CELP algorithms first perform a *linear predictive coding* (LPC) analysis on the transmitted signal. Second, vectors of prediction errors (excitation) are quantized and eventually encoded according to the method described above. Voice encoding in GSM networks is mainly based on this scheme. The initial GSM vocoder *linear prediction coding with regular pulse excitation* (LPC-RPE) as well as the more recent *algebraic code excited linear prediction* (ACELP) vocoder are adaptations of this technology. CDMA networks are based on the *Selectable mode vocoder* speech coding standard. This voice coding technology (named as such for it can be operated at a premium, standard and economy mode) uses a multistage algorithm. After a pre-processing stage, LPC analysis as well as pitch search are performed and frames of input signal are classified as silence/background noise,

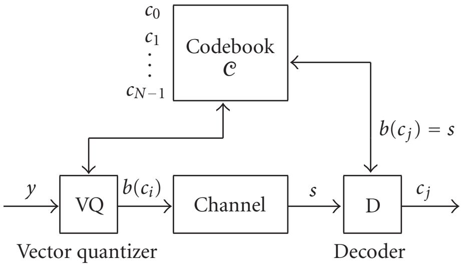

Figure 1: Signal transmission by vector quantization.

stationary unvoiced, nonstationary unvoiced, onset, nonstationary voiced, or stationary voiced. Depending on the frame type, either a eight-rate, fourth-rate, half-rate, and full-rate codec is selected. While background noise and stationary unvoiced frames are represented by a spectrum and energy modulated noise and are encoded with the fourth or eight rate codec, voiced frames are encoded with the full or half rate codec according to an extension of CELP, namely *extended CELP (eX-CELP)*. For such frames with a low pitch gain, eX-CELP behaves similarly as traditional CELP. For frames with high and stable pitch gain eX-CELP uses less bits for pitch encoding and allow more for excitation representation (see [4, 5] or [6] for further details).

In spite of its evident success in modern speech coding technologies, vector quantization has also a big drawback: the slightest transmission error on the string representing a binary codeword (e.g., one single bit error) can induce a very large difference (in terms of Euclidean distance) between the input and the output codevectors and, by consequence, an important distortion of the transmitted signal. Hence, the indexation of the codebook should be as robust as possible with respect to this problem.

The practical importance of vector quantization induced a relatively active research on the indexation problem. The main classical indexation algorithms that can be found in the literature (cf. [7–12]) are using general optimization techniques or heuristics. While achieving high performance, these methods require time consuming convergence stages.

A few combinatorial approaches (cf. [13, 14]) have also been proposed. These methods do not use any a priori assumption on the geometrical structure of the codebook. In this paper, we will study the problem of indexation starting of a hypothesis that the vectors of the codebook are distributed according the multidimensional Gaussian laws (we show that this hypothesis holds in practical situations). We use the fact that Gaussian laws can be approximated in a *discrete manner* by binomial laws to design new combinatorial algorithms of indexation resulting in higher performance and lower time complexity.

The paper is organized as follows. In Section 2 a mathematical formulation of the index assignment problem and a short survey of the existing methods are presented. Section 3 starts with a brief discussion justifying the hypothesis of the Gaussian structure of the codebook and a presentation (through several examples) of a classical discrete model of Gaussian distribution in terms of binomial coefficients. Further, a combinatorial algorithm based on this model for the assignment of binary codewords to the codevectors of vector quantizer is developed. In Section 5 the performance of the algorithm is analyzed. It is compared with that of the Wu and Barba's algorithm and of a system with randomly chosen binary codewords. Finally, Section 6 includes some preliminary ideas on potential applications of the algorithm and a conclusion.

2. VECTOR QUANTIZATION: PRINCIPLES AND CLASSICAL APPROACHES

2.1. Mathematical formulation of the problem

In this paper, we will assume that the vector quantizer is designed and fixed. That means we have a finite set (codebook) $\mathcal{C} = \{c_0, c_1, \dots, c_{N-1}\} \subset \mathbb{R}^d$ of codevectors as well as a mapping (the quantizer) $\mathcal{Q} : \mathbb{R}^d \to \mathcal{C}$, that associates to each vector y of $\mathbb{R}^d$ (input signal), the element c from $\mathcal{C}$ such that the Euclidean distance between y and c is minimal. Binary codewords on the output of vector quantizer are to be sent over a noisy channel. The channel is assumed to be a binary, symmetric, and memoryless chanel (BSC) with error probability ϵ. The length of the binary codewords is fixed and equal $K = \log_2 N$ (i.e., $N = 2^K$). The objective is to construct an indexation mapping b (as defined above) that makes this communication model as robust as possible with respect to transmission errors while having a minimum time complexity.

In order to formalize this statement we need to introduce some new notations and to define our performance criterions. Assume that the codevector c_i occurs with probability $p(c_i)$. Let $p(b(c_j) \mid b(c_i))$, $i, j = 0, 1, \dots, N-1$, denote the conditional probability of decoding c_j when transmitting c_i, which is equal to

$$p(b(c_j) \mid b(c_i)) = \epsilon^{d_H(b(c_i),b(c_j))}(1-\epsilon)^{N-d_H(b(c_i),b(c_j))}, \qquad (1)$$

where $d_H(b(c_i), b(c_j))$ is the Hamming distance between the binary words associated with codevectors c_i and c_j. Let us also denote by $d\ (c_i, c_j)$ the distance between codevectors c_i and c_j. In this paper we will consider the widely used squared-error distortion based on the usual Euclidean distance:

$$d(x, y) = \|x - y\|^2. \qquad (2)$$

Then the distortion performance criterion we adopt is determined in the following manner:

$$\mathcal{D} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} p(c_i) p(b(c_j) \mid b(c_i)) d(c_i, c_j). \qquad (3)$$

Therefore the index assignment problem is just the problem of finding an index assignment function b that minimizes (3). The objective of this paper consists in proposing a combinatorial approach for finding an appropriate suboptimal solution to this problem. Our solution will be based on the geometrical properties of the codebook.

2.2. *Classical approaches*

2.2.1. Heuristics

Since the problem is precisely formulated, in the rest of this section we envisage to recall some design and performance issues related to vector quantization for noisy channels. However, before we start several comments are in order.

There are two main questions concerning vector quantization that (apart from some exceptions as [15] or [16]) are usually treated separately: how to distribute the codevectors over the source, and how to choose the codewords, or indices, so that the effect of channel errors is not too degrading on the performance. This is due to the vector extension of the theorem for scalar quantization by Totty and Clark (see [17]) suggesting the separation of the overall distortion into the sum of the quantization distortion and the channel distortion for the squared-error distortion measure. Here we will mainly refer to the articles devoted to resolving the second question (as well as our present paper).

One more preliminary remark we wish to make is concerning the difficulty of index assignment problem. In fact, it is well known that the search problem of index assignment is NP-hard (see, e.g., [18]) and, as a consequence, all the proposed algorithms are necessarily suboptimal.

For this reason, most of heuristic algorithms we can find in literature first perform a deterministic search in a set of admissible configurations and then, in order not to terminate in a local minimum of a cost function, adopt a randomized approach (e.g., randomly generating the next configuration and allowing within reasonable limits the configurations of higher cost than the present). Among the first papers that assessed an index assignment problem for vector quantization by an heuristic approach one can find those of De Marca and Jayant [7], and of Chen et al. [19]. Farvardin [20] employed to the problem a simulated annealing algorithm. Zeger and Gersho [12] proposed a binary switching method, where pairs of codevectors change index in an iterative fashion, determined by a cost function. Potter and Chiang [18] presented a paper using minimax criterion based on hypercube that improves the worst case performance, important for the image perception. Knagenhjelm and Agrell employed the Hadamard transform to derive first an objective measure on success of index assignment [10] and then, to design efficient index assignment algorithms [11]. Similar theory was applied by Hagen and Hedelin [16] for designing vectors quantizers with good index assignments.

2.2.2. Combinatorial approaches

Only a few combinatorially flavored approaches have already been proposed. Cheng and Kingsbury (cf. [13]) designed a recursive algorithm based on hypercube transformations.

Wu and Barba (cf. [14]) proposed a method with smaller time complexity than the above solutions. Within their approach, minimization programs over the set of nonassigned codevectors are successively solved for each codeword. The aim of these minimization programs is to ensure that the Euclidean distance between two codevectors, which have been assigned codewords with Hamming distance equal to one, is kept low. More precisely, given a codebook with 2^K elements, the algorithm is initialized by assigning the codeword composed of $K0$'s to the codevector with the highest occurrence probability and then by indexing its K closest neighbors with the codeword of Hamming weight 1. Then for each already assigned codeword b all the binary words having a Hamming distance of 1 with b and higher Hamming weight are attributed one after the other to the not yet assigned codevectors minimizing a given criterion.

2.3. *Limitations of the classical combinatorial approach and baseline of our method*

Although Wu and Barba's method provides a simple and elegant solution allowing good improvement of the system's robustness to noise, this method does not consider any geometrical properties of the quantizer. Throughout this paper, we show how taking into account its geometrical structure may allow further reduction of the time complexity while increasing the performance. Using a classical discrete model of the Gaussian distribution we split the codebook into zones where each zone corresponds to a predefined set of codewords. Performing minimization programs within the zones rather than within the whole dictionary saves time complexity. Besides, at each step of the assignment process, it provides a better solution for the trade-off between optimizing the current codeword assignment and those of the remaining ones. In the following section, we present the discrete model of the Gaussian distribution we used to split the codebook. The index assignment method itself is described afterwards.

3. GAUSSIAN VECTOR QUANTIZERS AND DISCRETE MODEL OF GAUSSIAN DISTRIBUTION

3.1. *Working with Gaussian codebooks*

Our index assignment approach is specially designed for codebooks whose codevectors are distributed according to multidimensional Gaussian laws. The best results are for the codebooks with Gaussian distributions close to symmetric with respect to their mean point. However, the numerical simulations on nonsymmetric Gaussian codebooks presented at the end of the paper show that our method is also very well adapted for nonsymmetric Gaussian distributions with different variance values along the different principal directions.

The Gaussian hypothesis is valid in practical situations. Indeed, we studied several real codebooks provided by industrial partners. The results of applying Kolmogorov-Smirnov test to random samples, representing each coordinate, are satisfactory. Besides the reader is proposed to see Figure 2 for a projection on a plane of a real four-dimensional codebook and Figure 3 for a normalized repartition of the codevectors of another codebook along its principal axis.

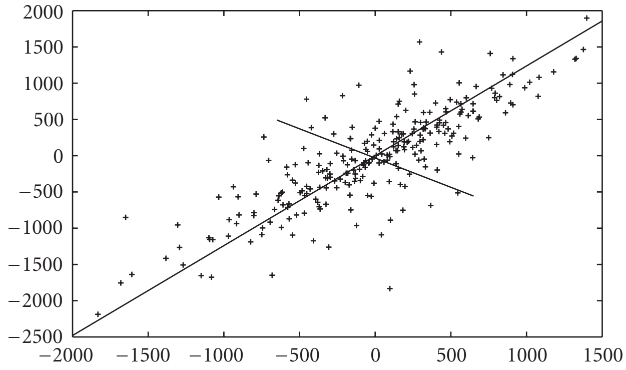

Figure 2: Projection on a plane of a real codebook with $N = 256$.

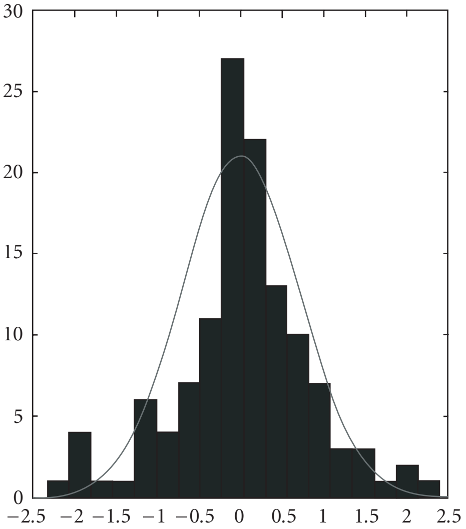

Figure 3: Normalized repartition of the projections of another codebook along its principal axis.

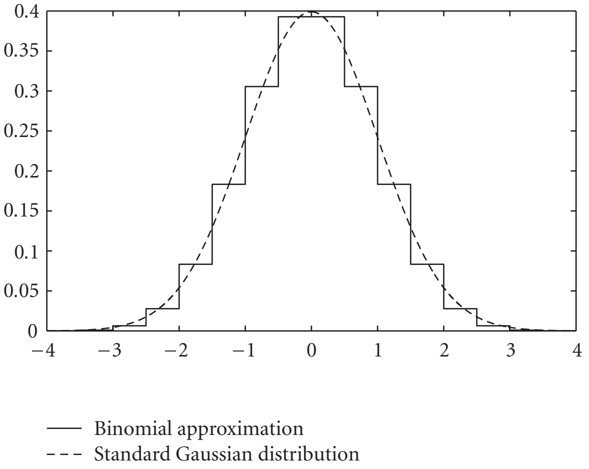

Figure 4: Binomial approximation of the standard Gaussian distribution for $K = 15$.

3.2. A classical discrete model of Gaussian distribution

In our algorithm the codebook $\mathfrak{C}$ is interpreted as an N-point discrete realization of the Gaussian distribution. More precisely, we use the well-known approximation in terms of binomial coefficients of the Gaussian density. This section gives a description of the one-dimensional version of this model followed by its generalization to d ($d > 1$) dimensions.

One-dimensional model

Let S be a segment on the line of length $(K+1)r$ (r being the parameter of the model). Consider a partition of S into $K+1$ adjacent segments numbered from left to right $S_0, S_1, \ldots, S_K$ of equal length r on the line. An N-point discrete approximation of the Gaussian distribution centered on the mean point of S with standard-deviation $2r$ is obtained by considering the *pdf*:

$$p^1(x) = \frac{1}{Nr}\binom{K}{i}, \quad \text{if } x \in S_i \tag{4}$$

and 0 if x does not belong to any of the segments. This step function corresponds to the repartition histogram that one would get if N points were distributed on S such that $\binom{K}{i}$ of them are in S_i (see Figure 4 for the binomial approximation for $K = 15$).

Generalization to d-dimensional symmetric Gaussian distributions

In d ($d > 1$) dimensions, the model is generalized by replacing the segments on the line by regions delimited by hyperspheres. Let S be a d-ball with center O intersected by a hyperplane H containing O. As we are considering symmetric Gaussian distributions any hyperplane containing O can be indifferently chosen. Then a carefully defined system of $\lfloor K/2 \rfloor + 1$ embedded d-balls centered in O

$$S \supset \mathcal{S}^1 \supset \cdots \supset \mathcal{S}^{\lfloor K/2 \rfloor} \tag{5}$$

provides together with the hyperplane H the partition of S into $K+1$ regions S_i ($0 \leq i \leq K$) of equal d-content. Let r (parameter of the model) be the radius of the central and smallest ball. Then the radiuses of the balls are related by the following equations:

$$\begin{aligned} R_i &= \sqrt[d]{(K+1)/2 - i}\, r, \quad i = 0, \ldots, (K-1)/2, \text{ for odd } K \\ R_i &= \sqrt[d]{K+1-2i}\, r, \quad i = 0, \ldots, K/2, \text{ for even } K. \end{aligned} \tag{6}$$

The regions S_i and S_{K+1-i} (with $i \neq \lfloor K/2 \rfloor$) are the symmetric halves of $\mathcal{S}^i \setminus \mathcal{S}^{i+1}$ with respect to the splitting hyperplane H. For odd K two central regions $S_{(K-1)/2}$ and $S_{(K+1)/2}$ are the halves of the central ball $\mathcal{S}^{(K-1)/2}$ and for even K the

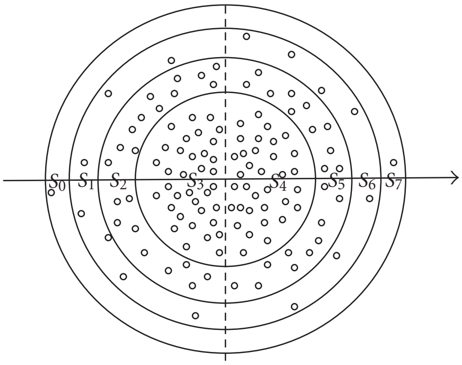

Figure 5: Partition of a 128 point 2-dimensional Gaussian codebook into 8 regions.

central region $S_{K/2}$ is simply the central ball $\mathcal{S}^{K/2}$. Similarly to the one-dimensional case we have an approximation of the Gaussian distribution centered in O and with variance-covariance matrix $(2r)^2 I$ (where I is the identity matrix) by considering the *pdf*:

$$p^d(x) = \frac{1}{N \times \mathrm{Vol}\,(S_i)}\binom{K}{i}, \quad \text{if } x \in S_i \tag{7}$$

and 0 if $x \notin S$ ($\mathrm{Vol}\,(S_i)$ denotes the content of S_i). As in the previous case, this step function corresponds to the d-dimensional histogram depicting the geometrical repartition of N points distributed in such a way that $\binom{K}{i}$ of them are in S_i ($i = 0, \dots, K$).

3.3. Link with Gaussian codebooks

In a few words, the main feature of the model is that an N point Gaussian codebook can be approximately partitioned into $K + 1$ regions with equal d-content, each region being bounded by two d-dimensional semihyperspheres such that the radiuses of the hyperspheres are related by (6), and the number of points in the ith region is $\binom{K}{i}$ (see Figure 5 for a two-dimensional example with $K = 7$). Note, that the above discrete model is chosen as it suits very well our problem. Indeed, it gives a very natural way to split a Gaussian codebook containing 2^K elements into $K + 1$ subsets of $\binom{K}{i}$ codevectors ($i = 0, \dots, K$). The binomial coefficient $\binom{K}{i}$ can also be interpreted as the number of codewords of length K and fixed Hamming weight i. That suggests for the Gaussian codebooks of size 2^K the existence of a natural correspondence between these subsets of codevectors and the subsets of codewords of fixed Hamming weight. Using this idea we develop an index assignment method in the next section.

4. INDEX ASSIGNMENT

Let us first present the general idea of the proposed approach and then focus on its full description.

4.1. The guiding principle

Looking at (3), one can remark that the couples of codewords with mutual Hamming distance equal to one are of particular interest. Indeed, they correspond to one-bit error on the transmission channel. When the *BER* is much less than one, the signal's distortion is mainly due to this kind of error. To keep the average distortion low, we need to minimize the Euclidean distance between codevectors indexed by neighboring codewords. It is also important to keep in mind the trade off between minimizing this distance for a particular pair of codewords and minimizing the sum of all the Euclidean distances between codevectors having neighboring codeword assignments.

We address these two issues through a two stage algorithm. During a preprocessing stage we split the given codebook into zones according to the discrete model described in the previous section. This allows us to establish a correspondence between the subsets of the codewords of a given Hamming weight and the subsets of codevectors belonging to a given zone. Due to this correspondence the codewords that differ only in one bit (and therefore belonging to the subsets with Hamming weight difference equal to one) are associated to subsets of geometrically close codevectors of adjacent zones. This repartition of the codebook into zones allows to conceive the algorithm as the sequence of consecutive minimization programs, each of them treating only the codevectors of adjacent zones. Therefore the time complexity is significantly reduced compared to the methods searching the whole codebook (see, e.g., [14]). Besides, limiting the search for the minima to predefined zones insures that the first local optimizations will not be done to the detriment of the following ones.

To understand better the importance of this repartition into zones, let us consider the special case when the codevectors have equal occurrence probability $1/N$. Without loss of generality we consider that K is an odd integer (similar calculations can be conducted when K is even). As one can see it on Figure 5, the maximal Euclidean distances between two vectors from two adjacent zones is upper-bounded by the sum of the radiuses of the two corresponding d-balls. As a result, the distortion due to one-bit error between codewords of weight p and those of weight $p - 1$ is upper bounded by

$$\left(\sqrt[d]{\left|(K+1)/2 - p\right|} + \sqrt[d]{\left|(K+1)/2 - (p-1)\right|}\right) r. \tag{8}$$

There are $\binom{K}{p}$ codewords associated to the codevectors of zone p and each of them has p neighboring codewords assigned to codevectors of the previous zone. The contribution of the one-bit errors to the distortion that would occur if we assume this repartition into zones is therefore less or equal to

$$\mathcal{D} \le \frac{\epsilon}{N}\sum_{p=0}^{K}\binom{K}{p} p\left(\sqrt[d]{\left|(K+1)/2 - p\right|} + \sqrt[d]{\left|(K+1)/2 - (p-1)\right|}\right) r. \tag{9}$$

We would like to compare this upper-bound to the average distortion yielded by a purely random indexation scheme.

If we consider the codevectors as independent random variables distributed in $\mathbb{R}^d$ according to the centered Gaussian law of variance-covariance $(2r)^2 I$, the average distortion due to one-bit error for a random indexation is given by

$$E(\mathcal{D}) = \frac{\epsilon}{2N} \sum_{i=1}^{N} K E(d(c_1, c_2)), \tag{10}$$

where we took into account that each codeword has K neighbors. The coefficient 1/2 insures that we do not count twice the contribution from each couple of neighboring codewords. The average distance between two codevectors is computed using the Gaussian assumption

$$\begin{aligned} E(d(c_1, c_2)) &= \frac{1}{(2r\sqrt{2\pi})^d} \int_{\mathbb{R}^d \times \mathbb{R}^d} d(c_1, c_2) \\ &\quad \times \exp\left\{ -\frac{|c_1|^2}{8r^2} \right\} \exp\left\{ -\frac{|c_2|^2}{8r^2} \right\} dc_1\, dc_2, \\ E(d(c_1, c_2)) &= \frac{2r}{(\sqrt{2\pi})^d} \int_{\mathbb{R}^d \times \mathbb{R}^d} d(c_1, c_2) \\ &\quad \times \exp\left\{ -\frac{|c_1|^2}{2} \right\} \exp\left\{ -\frac{|c_2|^2}{2} \right\} dc_1\, dc_2. \end{aligned} \tag{11}$$

The ratio ρ of the upper-bound (9) to the average distortion (10) is only a function of the length of the codewords K and the dimension d, namely,

$$\begin{aligned} &\rho(K, d) \\ &= \frac{\sum_{p=0}^{K} \binom{K-1}{p-1} \left(\sqrt[d]{\left| \frac{(K+1)}{2} - p \right|} + \sqrt[d]{\left| \frac{(K+1)}{2} - (p-1) \right|} \right)}{\left(\frac{2^K}{(\sqrt{2\pi})^d} \right) \int_{\mathbb{R}^d \times \mathbb{R}^d} d(c_1, c_2) \exp\left\{ -\frac{|c_1|^2}{2} \right\} \exp\left\{ -\frac{|c_2|^2}{2} \right\} dc_1 dc_2}. \end{aligned} \tag{12}$$

Figure 6 plots this ratio as a function of d for $K = 5, 7, 9$. We see that the improvement of the repartition into zone scheme with respect to a random indexation becomes more and more substantial as the dimension of the vector quantizer increases.

In practical situations, codebooks may not be strictly symmetric Gaussian. Hence, the partition of the codebook into zones we adopt will not exactly respect the specific geometrical bounds of the regions of the model. Rather, these bounds are adapted such that each zone accommodates the necessary number of codevectors and that the maximal distance between two vectors of adjacent zones is minimized. This is achieved by defining the radius of each d-ball as the distance from the codebook's mean point of the most remote codevector belonging to it. Also we split each d-ball by hyperplanes orthogonal to the principal direction of the codebook, since the dispersion of the projection of the codevectors along this axis is the most significant. For each ball, an appropriate hyperplane is chosen so its splitting results in two subsets of codevectors of equal cardinality. The method we proposed in the following sections can thus be applied to any codebook. However, for the codebooks having the codevectors mainly concentrated around the mean point, this partition appears to be better justified for the model continues to be a good approximation.

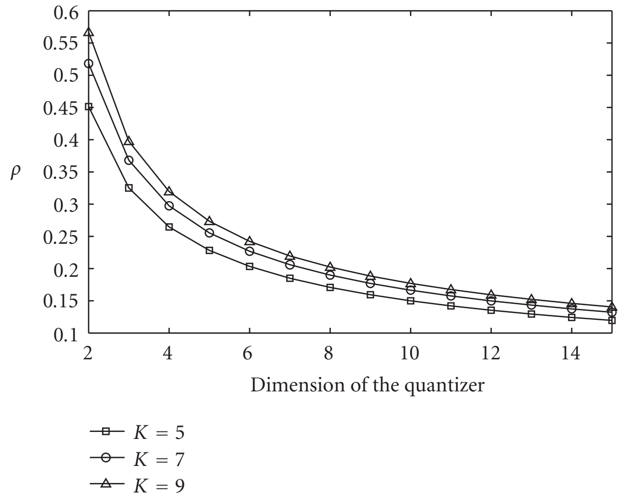

FIGURE 6: ρ as a function of the dimension for various values of K.

4.2. *Repartition of the codebook into zones*

The repartition into zones is performed according to the following procedure. Transfer the center of coordinates to the mean point of the codebook $\mathcal{C}$ and use the eigenvectors of the covariance matrix (computed by assuming equal weight for each point) of $\mathcal{C}$ as a new basis. Choose any of the possible orientations for this new system of coordinates. Select the principal axis, that is, the eigenvector corresponding to the greatest eigenvalue. Define m_0 as the median value of the coordinates of the codevectors on this axis and M_0 as the hyperplane orthogonal to the principal axis that contains m_0. The hyperplane M_0 splits the d-dimensional space into two subspaces with $N/2$ points of $\mathcal{C}$ each. Then, we sort all the codevectors of $\mathcal{C}$ in an ascending order with regard to their distances to the new center of coordinates, that results in an ordered set π:

$$\pi = \{c_{i_0} < c_{i_1} < \cdots < c_{i_{N-1}}\}. \tag{13}$$

Among the codevectors located to the left of M_0 (i.e., having a coordinate value on the principal axis less than m_0) select the one maximizing the Euclidean distance to zero. Similarly, among the codevectors to the right of M_0 select the one satisfying the same criterion. These two points constitute zones Z_0 and Z_K of our codebook. Numbering of zones is chosen in such a way, that the number i of a zone Z_i corresponds both to the cardinality $\binom{K}{i}$ of subset Z_i and the number of codewords with Hamming weight i to be assigned to it later on.

The other zones are built in an iterative fashion. Suppose that we already created $2p$ zones and that the ordered set

$$\pi \setminus \{Z_0 \cup \cdots \cup Z_{p-1} \cup Z_K \cup \cdots \cup Z_{K-p+1}\} \tag{14}$$

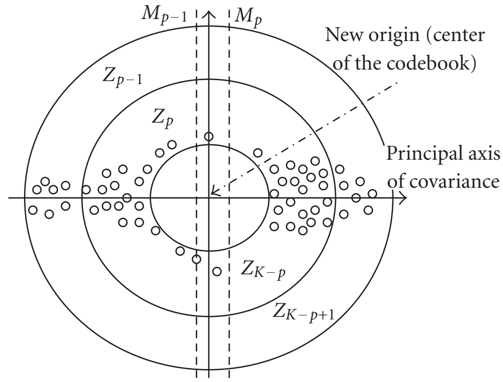
M_{p-1}
M_p
New origin (center of the codebook)
Z_{p-1}
Z_p
Principal axis of covariance
Z_{K-p}
Z_{K-p+1}

Figure 7: Iterative construction of the zones (only codevectors belonging to zones Z_{p-1}, Z_p, Z_{K-p+1}, and Z_{K-p} are shown).

is still not partitioned. In this ordered set consider the subset of the last $\binom{K}{p}+\binom{K}{K-p}$ codevectors (if K is even and only two last zones are still not defined consider the last $\binom{K}{p}$ codevectors). Find the median coordinate m_p of this subset with respect to the principal axis of the basis and, as well, M_p the median hyperplane. Then zone Z_p is defined as the set of codevectors of the considered subset located to the left of M_p, and the other half of the codevectors constitute zone Z_{K-p}. The process of repartition of the codebook $\mathfrak{C}$ into zones (creation of zones Z_p and Z_{K-p}) is illustrated by Figure 7.

4.3. Iterative assignment

We use the partition of $\mathfrak{C}$ into zones to construct a recursive procedure of indexation on which our combinatorial method relies.

Step 1. We assign the codeword $\overbrace{00\cdots0}^{K}$ to the codevector of zone Z_0 and the codeword $\overbrace{11\cdots1}^{K}$ to the codevector of zone Z_K.

Step 2. We proceed by assigning the indices to codevectors of zones Z_1 and Z_{K-1}, the next ones on the way to zero and both having cardinality K. We assign randomly all the K codewords of Hamming weight 1 to the codevectors of zone Z_1, and all the K codewords of Hamming weight $K-1$ to the codewords of zone Z_{K-1}.

Step 3 ($2 \le p \le \lfloor (K-1)/2 \rfloor$). Suppose now that we have already made p steps resulting in assignment of indices to all the codevectors within the first $2p$ zones $Z_0,\dots,Z_{p-1}$ and $Z_K,\dots,Z_{K-p+1}$. We proceed by assigning all the codewords of Hamming weight p to the codevectors of zone Z_p. For each such binary codeword b_j^p $\left(j \in 1,\dots,\binom{K}{p}\right)$ we consider the codevectors c_i^{p-1} of zone Z_{p-1} such that $d_H(b(c_{p-1}^i), b_j^p) = 1$, that is, the codevectors that were previously indexed with a codeword one-bit different from b_j^p. Then we look for the codevector c_j^p belonging to the subset of zone Z_p containing only codevctors that have not been yet indexed (further denoted Z_p^*) minimizing the contribution to the distortion

$$c_j^p = \arg\min_{c\in Z_p^*} \sum_{\substack{c_i^{p-1}\in Z_{p-1}\\ d_H(b(c_i^{p-1}),b_j^p)=1}} p(c_i^{p-1})\,d(c_i^{p-1},c). \tag{15}$$

Finally, we assign the binary codeword b_j^p to this codevector (i.e., $b(c_j^p) = b_j^p$). In other words, we look for the codevectors minimizing the distortion due to one-bit error during the transmission of the codewords of the previous zone. We proceed in a similar fashion to index the vectors in zone Z_{K-p}.

4.4. Time complexity

Let us now estimate the time complexity of the algorithm described above. The repartition into zones being neglegible with respect to the sequence of minimization programs we focus on this last point. For each program of minimization or equivalently each index assignment, the number of elementary operations performed is proportional to the size of the searched set of codevectors. When assigning the ith codeword of the pth zone of an $N = 2^K$ vector quantizer, this minimizing point is searched among $\binom{K}{p} - i$ codevectors. The global complexity C_N is then given by

$$\begin{aligned} C_N &= O\left\{\sum_{p=0}^{K}\sum_{i=0}^{\binom{K}{p}-1}\binom{K}{p}-i\right\}\\ &= O\left\{\sum_{p=0}^{K}\frac{1}{2}\binom{K}{p}\left(\binom{K}{p}+1\right)\right\}\\ &= O\left\{\frac{1}{2}\left(\binom{2K}{K}+2^K\right)\right\} = O\left\{\binom{2K}{K}\right\}. \end{aligned} \tag{16}$$

Now using Sterling formula gives a simple asymptotic equivalent to the algorithm complexity, namely,

$$\begin{aligned} C_N &= O\left\{\frac{(2K)^{2K}\sqrt{4\pi K}\left(e^K\right)^2}{e^{2K}\left(K^K\sqrt{2\pi K}\right)^2}\right\}\\ &= O\left\{\frac{(2)^{2K}}{\sqrt{\pi K}}\right\} = O\left\{\frac{N^2}{\sqrt{\log_2 N}}\right\}. \end{aligned} \tag{17}$$

For comparison purposes we compute the complexity of Wu and Barba's method. We notice that this algorithm follows a similar fashion of sequential minimization programs. However the minimizations are not performed within zones but in the set of all the codevectors remaining unindexed. The complexity of this algorithm is given by

$$C_N^{\text{WuBarba}} = O\left\{\sum_{i=0}^{N-1}(N-i)\right\} = O\left\{\frac{N(N+1)}{2}\right\} = O\{N^2\}. \tag{18}$$

As a result, our algorithm has a complexity that can be neglected at asymptote with respect to the Wu and Barba's method.

5. SIMULATION RESULTS

5.1. Simulations on real Gaussian codebooks

We conducted numerical simulations on three four-dimensional real codebooks provided by our industrial partner. For each of these codebooks, statistics of occurrence of the codevectors in real life mobile communications were provided as well. Five-second samples of real life conversation from 200, 000 different speakers were gathered to derive these statistics. Speech signals were encoded through an extended CELP process (similarly as in CDMA) based on each of the provided codebooks for prediction errors quantization and coding. Using these statistics we computed the expected output SNRs yielded by various indexation schemes as functions of the channel bit error rate on a typical mobile communication speech signal (having the same statistics). Figure 8 presents these output SNRs resulting from the following index assignment schemes: random index assignment, Wu and Barba's algorithm, the proposed approach. In order to illustrate the importance of the codebook repartition into zones, we also plotted the results of a random indexation of the codevectors in each zone by the codewords of corresponding Hamming weight.

One can see that our approach achieves better results than the Wu and Barba's algorithm on all the three provided codebooks. It is very interesting as well to see that the repartition into zones stage contributes for most of the improvement of our indexation scheme with respect to a random one.

5.2. CPU times comparison

As claimed earlier in this paper, our approach allows lower computation time than Wu and Barba's method to achieve the presented results. We drew at random a large number of 4 to 10 dimensional Gaussian codebooks of various sizes and measured the CPU time needed by the two methods to process each of these codebooks. Figure 9 depicts the ratio of the mean (averaged over dimensions and the codebook drawings) CPU times required by Wu and Barba's approach to the mean CPU times required by our algorithm as a function of the codebook size. We see that for very small set of codevectors, the time required by our method is higher because of the preprocessing stage. However when the set of codevectors has a cardinality above 16, simulations demonstrate that our method is less time consuming.

5.3. Simulations on bigger Gaussian codebooks

We applied our algorithm to bigger simulated codebooks of higher dimension. We drew at random 10, 000, 25-dimensional codebooks with 1024 codevectors according to

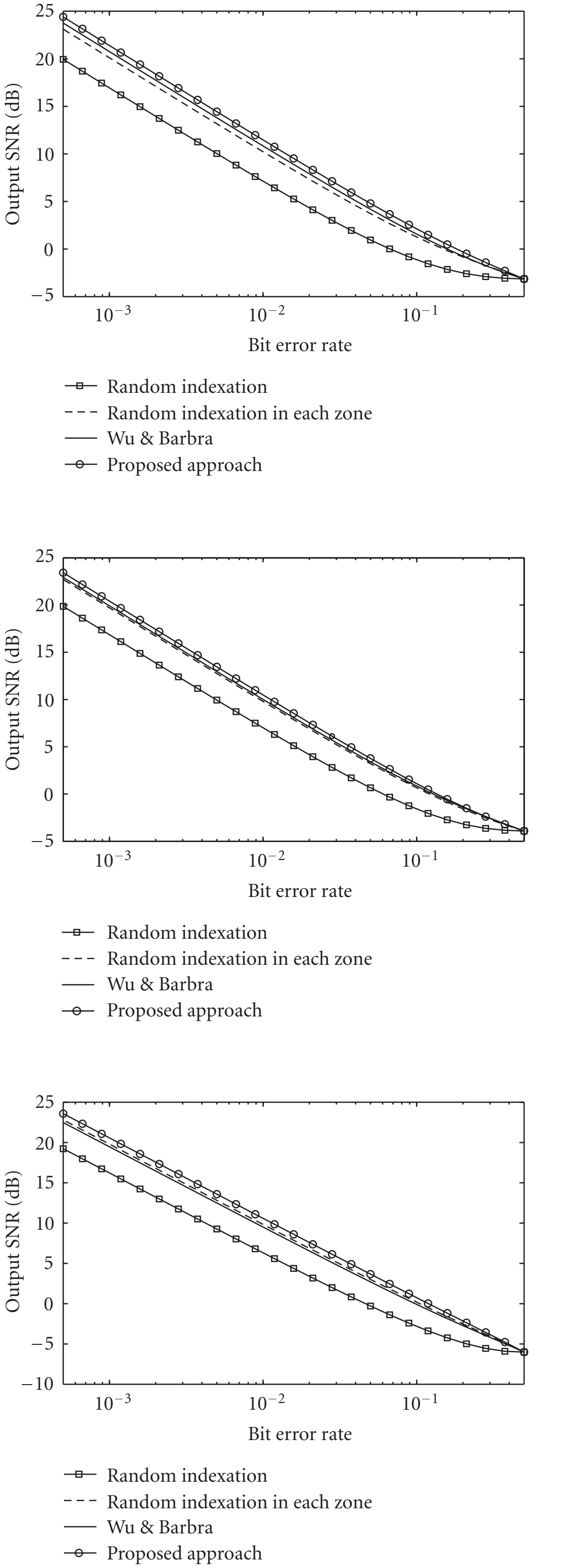

Figure 8: Output SNR yielded by various indexation schemes on two 256-vector codebooks (up and middle) and a 64-vector codebook (down).

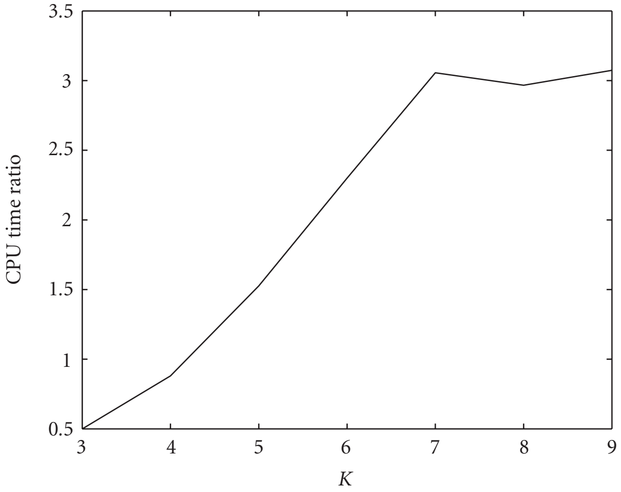

Figure 9: CPU time ratio of Wu and Barba's method to the proposed approach.

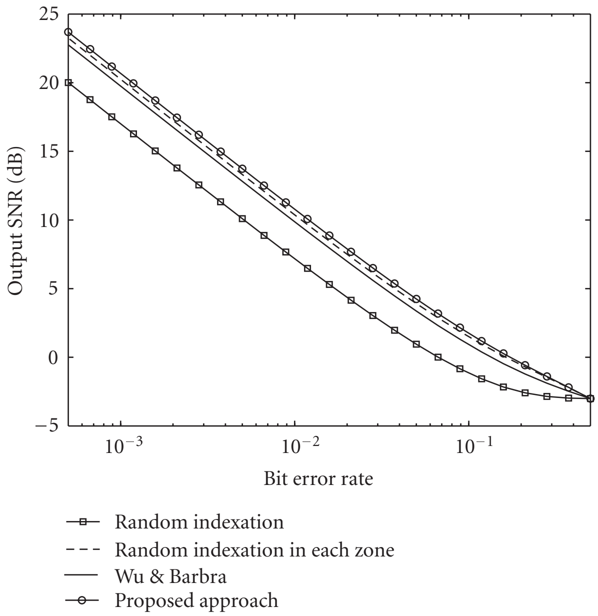

Figure 10: Average output SNR yielded by various indexation schemes on 1024-vector 25-dimensional codebooks.

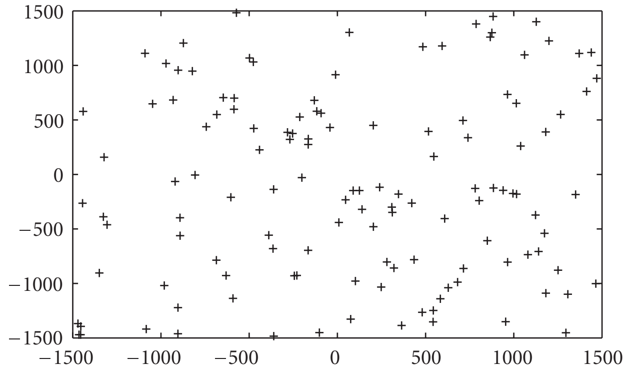

Figure 11: A uniform codebook.

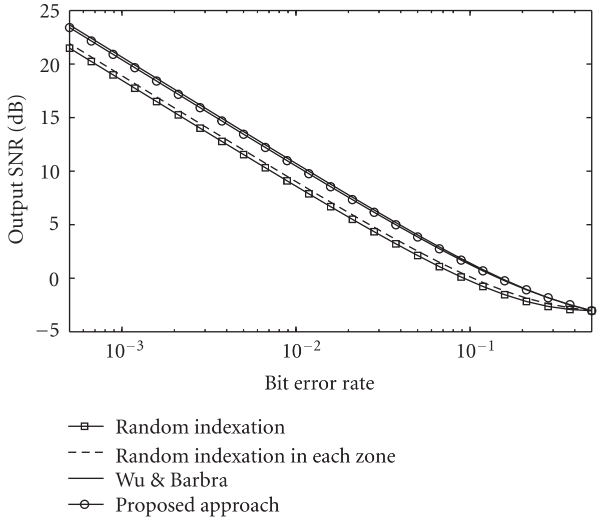

Figure 12: Output SNR yielded by various indexation schemes on a uniformly distributed codebook.

the normal distribution. We then multiplied the obtained codebooks by randomly drawn 25 × 25 matrices with coefficients uniformly drawn between −1 and 1 to simulate the fact that codebooks are in general not represented within the basis of the principal axis of their covariance matrice and to prevent them from being perfectly symmetric. Probability of occurence of the codevectors are uniformly drawn at random as well. Figure 10 shows the average expected output SNR recorded for these bigger codebooks. We see that increasing the size and the dimension of the codebooks did not changed much the performance patterns observed.

5.4. Extension to non-Gaussian codebooks

While the Gaussian hypothesis is the base for the general intuition of our approach, it turns out that the proposed method can be applied to any codebook. We generated an artificial four-dimensional codebook of 128 codevectors uniformly distributed within a given hypercube (see Figure 11 for a projection on the two first coordinates of the codebook). We applied the same indexation scheme as in the previous section to this codebook. The proposed approach performs less efficiently on this codebook. One can remark for example that the repartition into zone stage alone achieves a very little improvement with respect to a random indexation. We will notice however that the result of the whole algorithm is of similar order (even if a bit worse) as Wu and Barba's method (see Figure 12).

6. CONCLUSION

In this paper we proposed a new combinatorial method for index assignment of vector quantizers specially designed for Gaussian codebooks. In this case, our approach displays better performance than the Wu and Barba algorithm. In general the proposed method achieves an important reduction in time complexity. This suggests the idea of using our algorithm when time is a critical factor. We can imagine embedded systems with low computational capacities for very small networks (local) where quantization would be adapted in real time to the voice of the users. In this case, assignment could be rapidly adapted as well. We might think as well of statistical studies in voice coding that would require fast generation of a great number of codebooks with good index assignment. Besides, it can be employed as an initial assignment for general optimization methods in the field ([12] or [20]). To this extent, the repartition into zones together with a random indexation in each zone is very valuable since its time complexity can be neglected and the improvement with respect to a fully random assignment is quite substantial. Some applications might be relevant as well beyond voice coding as vector quantization is a widely spread method for lossy data compression (e.g., image and video coding).

REFERENCES

[1] R. M. Gray, "Vector quantization," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 1, no. 2, pp. 4–29, 1984.

[2] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1551–1588, 1985.

[3] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '85)*, vol. 10, pp. 937–940, Tampa, Fla, USA, April 1985.

[4] Y. Gao, A. Benyassine, J. Thyssen, H. Su, and E. Shlomot, "eX-CELP: a speech coding paradigm," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '01)*, vol. 2, pp. 689–692, Salt Lake City, Utah, USA, May 2001.

[5] Y. Gao, E. Shlomot, A. Benyassine, J. Thyssen, H. Su, and C. Murgia, "The SMV algorithm selected by TIA and 3GPP2 for CDMA applications," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '01)*, vol. 2, pp. 709–712, Salt Lake City, Utah, USA, May 2001.

[6] J. Makinen, P. Ojala, and H. Toukomaa, "Performance comparison of source controlled GSM AMR and SMV vocoders," in *Proceedings of International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS '04)*, pp. 151–154, Seoul, Korea, November 2004.

[7] J. R. B. De Marca and N. S. Jayant, "An algorithm for assigning binary indices to the codevectors of a multi-dimensional quantizer," in *Proceedings of IEEE International Conference on Communications*, pp. 1128–1132, Seattle, Wash, USA, 1987.

[8] J. A. Freeman and D. M. Skapura, *Neural Networks—Algorithms, Applications and Programming Techniques*, Addison Wesley, Reading, Mass, USA, 1992.

[9] R. Hagen and P. Hedelin, "Robust vector quantization in spectral coding," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '93)*, vol. 2, pp. 13–16, Minneapolis, Minn, USA, April 1993.

[10] P. Knagenhjelm, "How good is your index assignment?" in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '93)*, vol. 2, pp. 423–426, Minneapolis, Minn, USA, April 1993.

[11] P. Knagenhjelm and E. Agrell, "The hadamard transform—a tool for index assignment," *IEEE Transactions on Information Theory*, vol. 42, no. 4, pp. 1139–1151, 1996.

[12] K. Zeger and A. Gersho, "Pseudo-Gray coding," *IEEE Transactions on Communications*, vol. 38, no. 12, pp. 2147–2158, 1990.

[13] N.-J. Cheng and N. K. Kingsbury, "Robust zero-redundancy vector quantization for noisy channels," in *Proceedings of IEEE International Conference on Communications (ICC '89)*, vol. 3, pp. 1338–1342, Boston, Mass, USA, June 1989.

[14] H.-S. Wu and J. Barba, "Index allocation in vector quantisation for noisy channels," *Electronics Letters*, vol. 29, no. 15, pp. 1317–1319, 1993.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications Systems*, vol. 28, no. 1, pp. 84–95, 1980.

[16] R. Hagen and P. Hedelin, "Robust vector quantization by a linear mapping of a block code," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 200–218, 1999.

[17] R. Totty and G. Clark Jr., "Reconstruction error in waveform transmission," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 336–338, 1967.

[18] L. C. Potter and D.-M. Chiang, "Minimax nonredundant channel coding," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 804–811, 1995.

[19] J.-H. Chen, G. Davidson, A. Gersho, and K. Zeger, "Speech coding for the mobile satellite experiment," in *Proceedings of IEEE International Conference on Communications*, pp. 756–763, Seattle, Wash, USA, 1987.

[20] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 799–809, 1990.

E. A. Vassilieva holds a position of researcher in the French National Center for Scientific Research (CNRS) and is working at the Laboratory of Computer Science (LIX) in Ecole Polytechnique in Paris. Graduated in 1997 from the Mechanical Mathematical Department of Moscow State University (MS equivalent in mathematics and applied mathematics), she was awarded in 2000 a Ph.D. degree in symbolic computations and effective algorithms in noncommutative algebraic structures. During the two years that followed, Ekaterina Vassilievia occupied several postdoctorial research and teaching positions (NATO, Ecole Polytechnique, University Paris 7) in Paris before her current position in 2002. Her research interests lie mainly in algebraic combinatorics and applications of combinatorial methods in symbolic computation, telecommunications, and various fields of theoretical computer science like graph and map theory.

D. Krob is a Senior Researcher at the French National Center for Scientific Research (CNRS) and Professor at Ecole Polytechnique, where he is in charge of the Ecole Polytechnique-Thales teaching and research chair "Engineering of Complex Systems." He is a former student of Ecole Normale Supérieure. He received his Ph.D. degree and his second thesis (habilitation) in Computer Science from University Paris 7, respectively, in 1988 and 1991. His research interests cover algebraic and enumerative combinatorics, mobile telecommunications, formal languages theory, system modeling, information systems, theoretical computer science, and theoretical marketing. Nowadays his research focuses on formal modeling of complex systems. He is the Author of around 80 scientific papers, 4 books, and 2 patents. He directed two main national research groups (involving around 100 French research teams) at the border between mathematics and computer science. He was Founder and Head of the "Laboratoire d'Informatique Algorithmique Fondements et Applications" (CNRS & University Paris 7) during six years, before joining Ecole Polytechnique. He was also First Editor-in-Chief of the scientific international e-journal "Discrete Mathematics & Theoretical Computer Science" and Head of the steering committees of two main international scientific conferences (FPSAC and STACS).

J. M. Steyaert is Professor and Chair of the Computer Science Department at Ecole Polytechnique. His recent research interests include formal languages and grammars, complexity theory, as well as bioinformatics.