Enhanced Dynamic Codebook Reordering for Advanced Quantizer Structures

Jani Nurminen

Multimedia Technologies laboratory
Nokia Research Center, Tampere, Finland
jani.k.nurminen@nokia.com

Abstract

Conventional dynamic codebook reordering can often significantly enhance the achievable compression efficiency in simple one-stage vector quantization. When applied to more advanced quantizer structures, such as multi-stage vector quantizers, the performance of the technique becomes worse. This paper describes in detail an enhanced approach for dynamic codebook reordering that improves the performance by taking into account the whole quantizer structure. The significant efficiency improvements provided by the proposed approach are demonstrated in practical experiments. Though the results presented in this paper relate to a speech storage system, the proposed approach can also be employed more widely in compression applications that keep the encoder and the decoder in synchrony.

Index Terms: vector quantization, codebook reordering

1. Introduction

Vector quantization (VQ) is one of the most efficient and powerful tools in the field of data compression. In fact, no other memoryless coding scheme that maps a signal vector into one of N binary words can outperform vector quantization as there always exists a vector quantizer with codebook size N that provides at least the same accuracy [1]. When the quantizer is allowed to utilize memory, the compression performance can be improved using many different techniques. Examples of such techniques include predictive vector quantization (PVQ) [1], finite-state vector quantization (FSVQ) [1], and dynamic codebook reordering (DCR) [2] [3] [4].

In dynamic codebook reordering, when implemented as in [3] and in [4], the main idea is to dynamically reorder the codebook at each compression pass based on the selected code vector. Without reordering, the probabilities for the different indices are quite similar, making lossless coding relatively inefficient. The reordering exploits the correlation between successive vectors to make the probability distribution less flat, which in turn reduces the entropy and makes the lossless compression of the indices more efficient. The conventional dynamic codebook reordering approach works usually very well in the case of simple memoryless vector quantization, provided that the synchrony between the encoder and the decoder can be assured, but the technique becomes less efficient when applied to more advanced quantizer structures.

In this paper, the aim is to improve the performance of the dynamic codebook reordering technique in advanced quantizer structures. Here, the term advanced structure refers to cases where the structure also includes other parts in addition to a single full-search vector quantizer. Multi-stage vector quantization (MSVQ) [1] offers a typical example of such a structure. The proposed enhancements are described mainly from the viewpoint of MSVQ and predictive multi-stage vector quantization (PMSVQ) but the same ideas can be extended to other quantizer structures as well. The target application for the proposed technique is a speech storage system that operates on error-free bit streams.

The rest of this paper is organized as follows. First, the basic principles and the practical usage of the conventional dynamic codebook reordering technique are discussed in Section 2. The proposed enhanced dynamic codebook reordering approach is then introduced in Section 3. Section 4 describes experimental results that demonstrate the superior performance of the proposed approach in the speech coding related application. Finally, some concluding remarks are presented in Section 5.

2. Dynamic codebook reordering

2.1. Background information

Let $\mathbf{x}$ be the input vector to be quantized and let $\mathbf{C}$ be the codebook that is available both at the encoder and at the decoder. In conventional vector quantization, the encoder finds from the codebook $\mathbf{C}$ the code vector that best represents the input vector, i.e. the code vector that is closest to the vector $\mathbf{x}$ according to some distortion measure (e.g. weighted squared error). The compressed representation is then given by a digital symbol i that represents the index of the code vector that was selected by the encoder. During decoding, the dequantized version of the vector can be reconstructed simply by taking from $\mathbf{C}$ the code vector that corresponds to the received codebook index.

Since the number of code vectors in the codebook is always limited, the symbol i belongs to a limited set of symbols, $i \in \zeta$. If the number of code vectors, or equivalently the number of symbols in set ζ, is denoted as N, the number of bits needed to represent a single symbol i can be computed simply as $\lceil \log2(N) \rceil$. Through the use of lossless compression techniques, it may be possible to decrease the average number of bits per symbol and thus to achieve lower rate for the same distortion level. If it is possible to know or to estimate a probability $p(i)$ for every $i \in \zeta$, the theoretical entropy bound for lossless compression efficiency can be computed as

$$H = -\sum_i p(i) \log_2 \left(p(i) \right) \quad (1)$$

where the sum is computed over all symbols $i \in \zeta$. Without any reordering, the probabilities for the different symbols are usually quite similar, resulting in high entropy. Consequently, in this case, the lossless compression of the indices can only bring very limited benefits.

2.2. Description of the technique

The conventional dynamic codebook reordering technique was originally designed to facilitate the compression of the codebook indices [2]. It reduces the entropy of the index data by exploiting the correlation between adjacent vectors. Following the detailed description presented in [4], the encoding at time instant t is performed as follows. First, the codebook search is performed by finding the code vector $\mathbf{c}_k(t)$ that best represents the input vector $\mathbf{x}(t)$ similarly as described in Section 2.1. Then, the symbol $i(t) = \psi(k,t)$ corresponding to the code vector $\mathbf{c}_k(t)$ is stored or transmitted to the decoder. $\psi(k,t)$ is a simple dynamic index map [4] that relates the physical code vector index to its reordered index at time instant t. The dynamic index map is initialized as $\psi(l,0) = l$ for $l = 0, 1, ..., N-1$, and it is updated during each encoding pass after storing or transmitting the selected index. The updated order is obtained by sorting the values

$$\delta(l,t) = d(\mathbf{c}_l, \mathbf{c}_k(t)) \text{ for } l = 0, 1, ..., N-1, \tag{2}$$

in increasing order so that

$$\delta(l_0,t) \le \delta(l_1,t) \le ... \le \delta(l_{N-1},t) \tag{3}$$

where $l_0, l_1, ..., l_{N-1}$ belong to the set of integers between 0 and $N-1$. The dissimilarity measure $d(\cdot)$ in Equation 2 can be chosen freely as long as the computation can be performed both at the encoder and at the decoder. After the reordering, the updated dynamic index map for the next encoding pass can be obtained simply as

$$\psi(n,t+1) = l_n \text{ for } n = 0, 1, 2, ..., N-1. \tag{4}$$

The decoder performs the reordering similarly as the encoder. The reordering is carried out during each decompression pass after accessing the codebook. The mapping between the received symbol and the physical code vector index is obtained using an inverse dynamic index map $k = \psi^{-1}(i(t),t)$ [4]. As all the information is readily available both at the encoder and at the decoder, the dynamic index maps can be kept in synchrony without any side information, provided that the channel used for conveying the indices is error-free. Since usually the data to be compressed is not completely uncorrelated, the dynamic reordering makes the probability distribution less flat, which in turn reduces the entropy.

2.3. Usage in advanced quantizer structures

In multistage vector quantization, the input vector is quantized in two or more additive stages. The objective in encoding is to find a code vector combination, in other words a sum of the selected code vectors at different stages, that minimizes the resulting distortion. The selections are performed using some search algorithm. The simplest technique is to use sequential search, i.e. the selection is first performed for the first stage and then for the second stage, and so on. A better approach is to utilize joint selection, using e.g. the *M-L* tree search algorithm [5]. According to the descriptions given in [4], the correct way to use the DCR algorithm in the case of MSVQ is to apply it separately to each stage codebook. Each reordering is performed in the order of increasing dissimilarity to the code vector selected from that codebook. This approach is very simple and straightforward but the improvements gained over the case without reordering are usually rather small. This problem was very recently discussed in [4], where the authors correctly stated that after the first stage the vectors are more decorrelated, which makes the technique less efficient. The proposed enhancements described in Section 3 allow clearly better reordering in the case of multi-stage vector quantization, which significantly improves the performance especially at latter stages.

In predictive vector quantization, a prediction of the input vector is calculated using information about a finite number of previously quantized vectors, using e.g. auto-regressive (AR) prediction and/or moving average (MA) prediction. Then, the prediction error vector (also referred to as the prediction residual) is quantized instead of the original input vector. Finally, in decoding, the output is computed by adding together the prediction and the quantized prediction error. Since both PVQ and DCR exploit the correlation between adjacent vectors, the benefit achieved using both of them at the same time cannot be expected to be very high. Nevertheless, it is possible to apply dynamic codebook reordering on the indices of PVQ, using a direct approach similar to the one proposed for MSVQ in [4]. In this straightforward application of the conventional DCR technique, the reordering is performed simply using the dissimilarity to the selected codebook entry as the sorting criterion.

3. Enhanced DCR approach

The proposed enhanced dynamic codebook reordering approach improves the efficiency of DCR by taking into account the whole quantizer structure and all the pieces of information that are available from the other parts of the structure. To maximize the amount of available information, each reordering is delayed to the last possible moment, in contrast to the conventional DCR where the reordering is performed immediately after each quantization. In this section, the enhanced approach is presented from the viewpoint of MSVQ and predictive quantization. However, using the same ideas, it is possible generalize the proposed approach to other quantizer structures as well. For example, it would be possible to extend the technique for the predictive multi-mode matrix quantizer structure presented in [6].

In the proposed approach, the dynamic index map for each stage is first initialized similarly as described in Section 2.2. Then, the encoding at time instant t is performed as follows:

Step 1. Prediction (optional); Let $\mathbf{p}(t)$ denote the predicted vector and $\mathbf{e}(t)$ denote the prediction residual vector, i.e. $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{p}(t)$.

Step 2. Perform the codebook search for the vector $\mathbf{e}(t)$ for all the stages of the K-stage quantizer structure. Let $\mathbf{c}_{k,j}(t)$ denote the code vector selected at stage j ($j = 1, 2, ..., K$).

Step 3. Set $j = 1$.

Step 4. Dynamic codebook reordering for stage j using for $l = 0, 1, ..., N-1$ the measure

$$\delta_j(l,t) = d\left(\sum_{n=1}^{j-1} \mathbf{c}_{k,n}(t) + \mathbf{c}_{l,j} + \mathbf{p}(t), \hat{\mathbf{x}}(t-1)\right) \quad (5)$$

where $\hat{\mathbf{x}}(t-1)$ can be computed using

$$\hat{\mathbf{x}}(t) = \sum_{n=1}^{K} \mathbf{c}_{k,n}(t) + \mathbf{p}(t) \quad (6)$$

together with the sorting stragy similar to that given in equations (2), (3), and (4). The outcome is the updated dynamic index map $\psi_j(k_j,t)$ for stage j. Here, k_j denotes the physical index of the code vector selected at stage j.

Step 5. Dynamic index mapping for stage j: $i_j(t) = \psi_j(k_j,t)$.

Step 6. Transmit or store the digital symbol $i_j(t)$. The lossless compression can be performed during this step or later jointly for several symbols at the same time.

Step 7. Set $j = j + 1$. If $j \le K$, go to Step 4. Otherwise, exit.

If the lossless compression is not carried out in Step 6, it must be handled later, e.g. after processing the whole vector or after processing a block of vectors, or even after processing all input vectors. The optimal timing of the lossless compression depends on the selected compression algorithm. For example, with Huffman coding [1] the compression can be done in Step 6 but with arithmetic coding [1] the compression should be done for a larger block of symbols at the same time.

The operation of the decoder is modified in a similar manner. The decoding at time instant t is performed as follows:

Step 1. Prediction (optional); Set $j = 1$.

Step 2. Dynamic codebook reordering and dynamic index map update for stage j similarly as in Step 4 of encoding.

Step 3. Inverse mapping for stage j: $k_j = \psi_j^{-1}(i_j(t),t)$.

Step 4. Set $j = j + 1$. If $j \le K$, go to Step 2. Otherwise, continue to Step 5.

Step 5. Reconstruction of the dequantized vector $\hat{\mathbf{x}}(t)$ using Equation (6). Exit.

As in the case of conventional DCR, all the necessary information is readily available both at the encoder and at the decoder. Thus, there is no need to include any side information, unless the channel is erroneous. However, an occasional reset of the dynamic index maps can be included to allow more convenient access to the bit stream.

The above descriptions are applicable to both MSVQ and PVQ, and to the combination of them. If the quantizer structure does not include prediction, p(t) can be set to zero vector for all t. If there is only one stage (K = 1) and no prediction, the proposed technique reduces to the conventional DCR approach (with different timing of the reordering). As will be shown in Section 4, the proposed modifications make favourable changes to the probability distributions, which in turn reduces the entropy and significantly improves the compression efficiency. This performance advantage can be achieved at a very low cost: with careful implementation, the additional complexity compared to conventional DCR is only one add operation per stage.

4. Experimental results

As mentioned in the introduction, the target application for the proposed technique is a speech storage system that always operates on error-free bit streams. To demonstrate the benefits of the proposed approach, some experiments were carried out in this framework, more specifically in the quantization of the line spectral frequency (LSF) vectors derived from the time-varying linear prediction coefficients. Since many speech coders utilize linear prediction, this quantization task is one of the most common tasks in speech coding, and hence a good example application for the proposed technique. Due to space limitations, only a selection of interesting experiments is presented here. In the experiments, the training set contained 180,000 10-dimensional LSF vectors and the test set included a separate set of another 180,000 vectors, all estimated at 10 ms intervals from an internal speech database containing 8-kHz speech signals. The speech database consisted of active speech containing full sentences from many female and male speakers in several languages.

Table 1 illustrates the results from an experiment involving the application of the proposed approach in MSVQ. The studied quantizer structure included 4 stages with 64, 64, 32 and 32 code vectors in their codebooks. The training was performed using the multi-stage VQ simultaneous joint design algorithm [5]. Weighted squared error was used both as the distortion measure in the M-L tree search (with M = 8) and as the dissimilarity measure in the reordering. The weights were computed similarly as in [7] in such a manner that during the codebook search the weights were derived based the corresponding unquantized LSF vector whereas in the reordering the weights were computed based on the latest quantized LSF vector. Table 1 contains a comparison of the theoretical bit rates achievable using the same codebooks in three cases: using lossless compression without reordering, using conventional DCR, and using the proposed enhanced approach. As can be seen from these theoretical numbers obtained through empirical entropy (Equation (1)), the proposed approach clearly outperforms conventional DCR. The reason for the improved performance can be seen from Figure 1 that depicts the empirical index probabilities for the fourth stage, using the test data and the three different approaches (the same as in Table 1). The proposed technique produces the least flat distribution and consequently the lowest entropy. The difference compared to the conventional DCR is significant, especially in the latter stages.

To illustrate the effect that the number of stages has for the performance, results from two alternative designs are presented in Table 2 and Table 3. More specifically, Table 2 contains results obtained using a 3-stage quantizer structure. The performance advantage gained using the proposed approach is a bit smaller than in the case of the 4-stage quantizer but the gap to the conventional DCR is still a clear 260 bps. Table 3 depicts the theoretical bit rates achievable in the case of a 5-stage quantizer. Here, the increased number of stages makes the difference between the conventional DCR and the enhanced DCR larger (now 340 bps).

In Table 4, the 5-stage quantizer structure of the previous experiment was complemented with a predictor. The prediction was performed using a first order auto-regressive predictor with a diagonal predictor matrix. The predictor and the codebooks were jointly optimized using a combination of the multi-stage VQ simultaneous joint design algorithm [5] and the asymptotic closed loop design approach [8]. The PMSVQ structure contains properties that make it very challenging from the viewpoint of dynamic codebook reordering. Nevertheless, the proposed approach still clearly outperforms conventional DCR but the improvement of 100

Table 1. *Theoretical bit rates achievable using a 4-stage MSVQ for LSFs (originally 2200 bps).*

	No reorder	DCR	Proposed
Stage 1	566	368	325
Stage 2	582	531	407
Stage 3	497	482	397
Stage 4	497	491	426
Total	2142	1873	1555

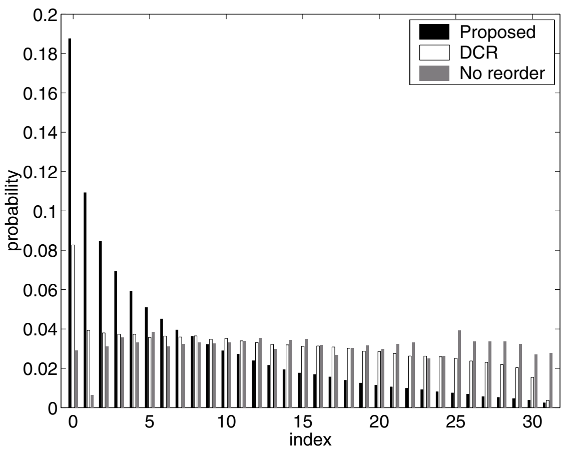

Figure 1. *Index probabilities at the last stage of the 4-stage quantizer.*

bps is smaller than in the structure that did not contain prediction (see Table 3).

It should be noted that the results presented in this paper were selected from a larger set of experimental results. Consequently, the results shown here do not represent the best possible performance gain achievable using the proposed approach but instead they offer typical examples of the advantages. In all the experiments that have been conducted on the proposed approach, it has constantly outperformed the conventional DCR technique without any exceptions.

5. Concluding remarks

This paper has presented enhancements to the conventional dynamic codebook reordering technique. The proposed enhancements improve the performance of the technique in advanced quantizer structures by taking into account the whole quantizer structure. The enhanced DCR technique has been described in detail and the clearly superior performance of the proposed approach has been demonstrated using experimental results. Though the practical results reported in this paper were obtained in a single speech coding application, the proposed technique can also be utilized in other applications that require data compression, provided that the encoder and the decoder are kept in synchrony.

6. References

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

Table 2. *Theoretical bit rates achievable using a 3-stage MSVQ for LSFs (originally 2200 bps).*

	No reorder	DCR	Proposed
Stage 1	774	509	457
Stage 2	687	642	519
Stage 3	694	680	593
Total	2155	1830	1570

Table 3. *Theoretical bit rates achievable using a 5-stage MSVQ for LSFs (originally 2200 bps).*

	No reorder	DCR	Proposed
Stage 1	464	300	266
Stage 2	482	440	328
Stage 3	397	379	301
Stage 4	399	390	327
Stage 5	400	395	342
Total	2142	1904	1564

Table 4. *Theoretical bit rates achievable using a 5-stage PMSVQ for LSFs (originally 2200 bps).*

	No reorder	DCR	Proposed
Stage 1	448	431	419
Stage 2	476	471	434
Stage 3	398	398	376
Stage 4	399	398	383
Stage 5	400	400	387
Total	2120	2099	1999

[2] DeNatale, F. G. B., Fioravanti, S., and Giusto, D. D., "DCRVQ: A new strategy for efficient entropy coding of vector-quantized images", IEEE Transactions on Communications, Vol. 44, No. 6, pp. 696-706, 1996.

[3] Sridharan, S. and Leis, J., "Two novel lossless algorithms to exploit index redundancy in VQ speech compression", in Proc. ICASSP '98, pp. 57-60, 1998.

[4] Krishnan, V., Barnwell III, T. P., and Anderson, D. V., "Using dynamic codebook re-ordering to exploit inter-frame correlation in MELP coders", in Proc. Eurospeech '05, pp. 2717-2720, 2005.

[5] LeBlanc, W. P., Bhattacharya, B., Mahmoud, S. A., and Cuperman, V., "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding", IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 4, pp.373-385, 1993.

[6] Nurminen, J., "Multi-mode quantization of adjacent speech parameters using a low-complexity prediction scheme," in Proc. Eurospeech '03, pp. 1069-1072, 2003.

[7] Paliwal, K. K. and Atal, B. S., "Efficient vector quantization of LPC parameters at 24 bits/frame", IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 1, pp. 3-14, 1993.

[8] Khalil, H. and Rose, K., "Robust predictive vector quantizer design," in Proc. Data Compression Conference 2001, pp. 33-42, 2001.