15'th IEEE convention of Electrical & Electronic Engineers in ISRAEL Apr 1987

IEEE - Israel 1987

VECTOR QUANTIZATION OF SPEECH SIGNALS IN SUBBANDS

R. Arad D. Malah

Electrical Engineering Department
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

ABSTRACT

This paper introduces a speech coding system, which is based upon subband coding (SBC) and vector quantization (VQ). Unlike conventional subband coding systems, which apply scalar quantization and produce poor-quality speech under 16 Kbps, the proposed system yields good output quality even at 9.6 Kbps.

The paper begins with a brief introduction of subband coding, a common frequency-domain speech coding method, and of the emerging technique of vector quantization. Next, the problems of vector extraction and system adaptation are discussed, as these problems arise when the two methods are combined. Results of experiments with several types of systems are then presented, with detailed comparisons. The recommended system is finally introduced, and its practical traits are discussed.

INTRODUCTION

Subband coding is a speech compression scheme that, like transform coding, is considered to be a frequency-domain method [1]. A subband coding system consists of two parts - a filter bank and a coding mechanism. The speech signal is first filtered resulting in its division into several frequency bands. The signal in each band is down-sampled to its Nyquist rate and then coded, in order to obtain the required rate reduction. The coded information is transmitted through the channel, to be decoded and reconstructed into the output signal, using an appropriate synthesis filter bank. A typical SBC system is depicted in figure 1.

The subband coder system design is typically separated into two independent tasks: that of designing the filter bank and that of the coder design. The filter bank is usually of the Quadrature-Mirror type [2], having the important advantage of aliasing cancellation. As this work deals mainly with the coding part of the system, we will not elaborate any further upon the filter bank design.

The coders in a conventional subband coding scheme are adaptive scalar uniform quantizers within a prediction loop. The number of bits allocated to the coder in each band is chosen according to its relative importance, in terms of energy, as compared to the other bands. With such a coding scheme one can achieve good subjective speech quality down to about 16 Kbps, and still not pay too much in complexity (see, for example,]3]). Further bitrate reduction, however, results in a swift degradation. To prevent this, one should resort to other coding methods; we have chosen for this purpose to employ vector quantization.

Vector quantization consists of the following procedure: source samples are grouped together, to form vectors, and these are coded using a so-called codebook of precomputed code-vectors. For each input vector, the closest matching code-vector, according to a pre-defined distortion measure, is found. The code-vector's index is sent over the channel, and the receiver uses it to retrieve and output this vector, using its own copy of the codebook.

Theoretically speaking, vector quantization is not a new concept. Still, its implementation was impractical for a long time, This was due to the lack of a quantizer design algorithm on one hand, and of a technology suitable for surmounting the coder's complexity, on the other. Of these problems, the former was solved in 1980, with the publication of a method for quantizer design, by Linde, Buzo and Gray [4]. Technological advancement, aided by simplified schemes devised to decrease the coder's complexity, have solved the latter.

The algorithm proposed by Linde, Buzo and Gray is based on a classical design method for scalar quantizers, commonly known as the Lloyd-Max algorithm. This scheme usually works upon a large training sequence, representing the source to be quantized, and uses an optimization procedure that converges to a locally optimal solution. As already mentioned, suboptimal variants of this algorithm are typically employed, permitting faster coding and smaller codebook storage needs. Of these, the most widely used are tree-structured codebooks and cascades, or multistage, systems, A thorough review of vector quantization can be found in [5].

VQ IN SBC SYSTEMS

The use of vector quantization in subband coding systems has already been discussed in the literature. The first two papers concerning this subject, [6] and [7], appeared in 1984, and a third paper, [8], in 1986. Each of those papers presents a specific system, but does not contain a thorough comparative study, and many interesting details concerning SBC-VQ systems in general are left out. Therefore, we proceed by highlighting several prominent questions, concerning the incorporation of vector quantization into subband coding systems.

When vector quantization is performed on "raw" speech samples, vectors are formed artificially by grouping several consecutive samples. When used in SBC systems, the situation is slightly different. The filter bank outputs a two-dimensional stream consisting of trains of consecutive samples, appearing in several frequency bands. The question of how to extract vectors from this data for the coding process, now becomes less trivial.

The best choice would be, of course, one involving both frequency- and time-domain properties of the signal: several spectral components, each one containing several consecutive samples, will then be used. Unfortunately, this method (which may be called "matrix quantization") is rather demanding in complexity. Two degenerate cases thereof are then to be considered: "vertical" (or frequency-wise) coding, and "horizontal" (or time-wise) coding.

The "vertical" system is thus characterized by coded vectors, having in each component a sample from a different frequency-band of the signal, all occurring simultaneously, as depicted in figure 2. Such a method is reminiscent of coding a source with memory, by using techniques from memoryless source coding, on a transformed version of the source. The decomposition into subbands, being a crude DFT, helps in a similar way to decorrelate the signal before the actual coding. A "vertical" coding scheme is also, in a sense, more "natural", as the vectors formed are not artificial, but represent a contained

entity - the short-time spectrum.

"Horizontal" coding, on the other hand, works separately on each band, grouping consecutive samples of that band into vectors to be coded, as depicted in figure 3. Unlike "vertical" coding, where the codebook is intended to take care of sharing the resources among the bands, we have here to take care of this task explicitly. The easiest method to do so is by using a codebook allocation procedure, similar to the bit-allocation scheme used in the scalar case. Each band would then be assigned a codebook of size proportional to its relative importance, and the "horizontal" system is thus a vector extension of a scalar subband coder.

Another prominent question, arising when VQ is to be employed in a subband coding system, is that of adaptation. As mentioned above, scalar SBC systems usually employ adaptive coders, aiming to account for the non-stationarity of the speech signal. Unfortunately, adapting a vector quantizer's codebook in real time is a most burdensome task. An easier alternative, first proposed in [9], is to employ a vector AGC mechanism. Instead of adapting the codebook to follow the properties of the input, the vectors to be coded are "normalized" to fit into a reduced dynamic range, more easily tackled by the coder.

A completely different approach to adaptation is dynamic resource sharing. This method can, of course, be employed only in the "horizontal" case, where it is easily implemented by a dynamic codebook allocation. The sizes of the codebooks allocated to the bands are dynamically varied, to fit at each given moment to the input's properties. This, again, is an extension of a method commonly used with scalar subband coders.

As mentioned above, the papers dealing with SBC-VQ published so far, do not give a complete comparative study of the issues mentioned. We thus proceed by a presentation of our own results, based upon thorough experimentation and detailed comparisons.

RESULTS OF EXPERIMENTS AND COMPARISONS

Before experimenting with SBC-VQ, a scalar SBC system was designed, to serve as a benchmark for comparison. The best results were obtained with a system which uses an eight-band filter bank, and backward-adaptive uniform scalar quantizers without prediction. This system employs dynamic bit-allocation, according to the optimal allocation formula (see, for example, [10]). As this allocation does not take into account the non-negative integer constraint on the number of bits, we had to amend the resulting allocation slightly. The allocation's adaptation is forward, i.e. sent as side-information. Static allocations and backward-adaptive allocations were tried and discarded, being less satisfactory.

At 16 Kbps, the "scalar" system gives an SNR of 18.3 dB and a segmental SNR of 19.4 dB. The produced speech quality is very good, with a slight "ringing" in the background. At 9.6 Kbps and 8 Kbps, on the other hand, SNR values are about 13 and 11 dB respectively, and the quality is very much degraded. Loud ringing and a high-pass filtering effect are felt, and are very annoying at those rates.

To the above "scalar" system we compared "vertical" and "horizontal" SBC-VQ systems. The basic "vertical" system codes eight-dimensional vectors (a sample for each band) and has codebooks suitable for working at 8 Kbps and 16 Kbps. At 8 Kbps the codebook contains 256 vectors of dimension 8 each, whereas at 16 Kbps a two-stage codebook is used, having the 8 Kbps codebook as its first stage and an equal size codebook for the second. Its output SNR is higher than 21 dB, both in regular and segmental measures, at 16 Kbps, and about 15 dB at 8 Kbps. Unfortunately, the system did not perform as well at listening tests - no better than the "scalar" system at 16 Kbps, and not good enough at 8 Kbps.

Trying to improve the subjective results, we first used the vector-AGC mechanism proposed in [9] as a measure for adaptation. A gain factor was computed, corresponding to the average norm of sixteen consecutive vectors. It was used to normalize the vectors, and then was sent as side-information. The codebook design is now modified to take into account the gain normalization of the input vectors in the following way: the optimal code-vector assigned to each "region" in the vector space is not its average member, but a scaled version thereof, normalized by the average gain factor computed for this "region". Following the changes, the results were improved, but not drastically.

We then tried noise-shaping, by which the error is "colored" to match the input's spectral properties, thus supposedly improving the subjective quality. Noise shaping is commonly used in speech coding, either in the time-domain, by using a coloring filter, or in the frequency-domain, by using a weighted bit-allocation method, where the energy of each band is multiplied by a spectral weighting factor, In the context of "vertical" WQ, the error to be colored is actually the average distortion achieved in the codebook design. We thus define this distortion as weighted by the same spectral weights, and expect a noise-shaping effect. As in the case of the gain-adaptation, the codebook design is modified again: this time the scaling is done by the spectral weights instead of the gain. Unfortunately, the improvement achieved here was also small, and this method too did not lead to a major breakthrough.

A "horizontal" system was then designed, based on four-dimensional vectors. The codebook-size allocation to the bands was constant, based on long-term statistics, and this system achieved SNR's of about 18 dB at 16 Kbps and about 12 dB at 8 Kbps. Subjectively, the results were not very promising either. But for this system, as opposed to the "vertical" system, the incorporation of a similar AGC-type adaptation mechanism into the scheme proved to be a great success. Although the SNR values did not change drastically, there was a major improvement in output speech quality, which now sounded almost natural at 16 Kbps and only slightly distorted at 9.6 Kbps and even 8 Kbps.

Trying to improve further these results, we made the codebook allocation dynamic, using the same algorithm as in the "scalar" case. The adaptation here does not add considerable complexity, as the allocation can be based on using gain factors, already calculated, as energy estimators. However, here a disappointingly small improvement was achieved, which prompted us to discard this scheme.

In order to reduce the system's complexity, we then tried using suboptimal tree-structured codebooks, which permit logarithmic, instead of linear, codebook search time. There is of course a trade-off, as the codebook storage space is doubled, to provide for the search-tree constructed, and the quality is a little degraded, due to the non-exhaustive nature of the search. In our case, however, the degradation was barely discernible. The tree-structured codebooks give us then a system which is much simpler to implement, at almost no additional cost in quality and a certain cost in storage.

Being more complex than its scalar counterpart, the "horizontal" SBC-VQ system is not recommended for use at 16

Kbps, where the scalar system's performance is quite good. At 9.6 Kbps, on the other hand, it has no competitors among the other systems we designed. We thus proceed by presenting, in a more detailed fashion, our recommended system's structure and some of its practical traits.

THE RECOMMENDED SYSTEM AND ITS PROPERTIES

The system we recommend for coding speech at 9.6 Kbps is thus a "horizontal" SBC-VQ system with a vector-AGC adaptation mechanism. It has eight subbands, each one coded separately using a gain-controlled vector quantizer. In each band the samples are grouped into four-dimensional vectors, and each vector's gain is estimated as the average norm of a four-vector block. The gain is logarithmically quantized and sent as side-information. The normalized vectors are coded using a (tree-structured) optimal codebook, designed according to a gain-adaptive version of the Linde, Buzo and Gray algorithm, presented in [9] (see also in the previous section).

The codebook size allocation is static, and given as follows: the first band has two cascaded codebooks of 256 and 64 vectors respectively; the following bands have codebooks for which the number of vectors is 256, 8, 4, 4, 2, 1, 1 respectively; the following bands have codebooks for which the number of vectors is 256, 8, 4, 4, 2, 1, 1 respectively (i.e. the two highest bands are not coded at this rate). The total rate is thus 7.5 Kbps for the main information. For the side information we use 0.25 Kbps for each active band (i.e. except for the highest two) and thus 1.5 Kbps in total. This leaves us with 0.6 Kbps for error control. The system results in an SNR of 14 dB and a segmental SNR of 14.4 dB. In informal listening the quality was judged to be very good.

Calculating exactly the system's complexity is quite tiresome, but is important for its real-time implementation. Storage requirements can be approximated at about 5K numbers, which consists mainly of the code-vectors, but also of the analysis-synthesis filter coefficients. Time complexity is almost equally divided between the filtering and the coding operations, and several time-reducing measures can be taken, such as using polyphase filter structure and tree-structured codebooks. The "time complexity" can be approximated at about 50 multiply-add operations per sample, or 0.4 Mflops. It may be noted that these requirements can be met by today's technology.

We tested the recommended system for its immunity to noise. First, we tried a version without error-control (i.e. working at 9 Kbps), and at a bit-error-rate of 1E-4 the performance was found to be without noticeable degradation. We then incorporated into the system an error-correcting code applied only to the side-information, at an additional rate of 0.6 Kbps. This improved further the results: the system can now be used at a bit-error-rate of 1E-3 without it being noticed. Tests were also made about using the system on speech corrupted by additive white Gaussian noise. The effect is only a coloring of the noise, without a noticeable change in the noise-power or in the intelligibility of the output speech.

We then checked the system's immunity to tandeming, i.e. the degradation in quality of a signal passing through several identical systems, connected in series. The degradation was found to be subjectively smooth, although the objective results show a degradation of about 4 dB after the second system, and of another dB for every additional system connected. The general effect is not of an increased background noise but of high-pass filtering, which is less annoying.

Finally, we checked the system's response to voice-band-data signals. We used a computer simulated 2400 bps QPSK modem with an 1800 Hz carrier. Unfortunately, even an input without any noise (digital or analog) was decoded with a bit-error-rate of 4E-3, and so we did not even test our system with noisy channels. These results are probably due to the static bit-allocation used in the system, which is tailored to speech signals, and is not appropriate for data. We propose therefore that for data transmission purposes, different quantizers should be designed and used for data and speech, aided by a data/speech classifier. This direction of research is, however, not within the scope of this work.

CONCLUSION

In this paper a system for coding speech at 9.6 Kbps, employing subband coding and vector quantization, is recommended. This system is found to outperform other SBC systems that were designed and compared to it in this work, using both scalar and several vector coding schemes. It is based upon "horizontal" VQ, and employs a simple AGC-mechanism to account for the speech signal's non-stationarity. Its properties, detailed in the previous section, qualify it as a serious candidate for good-quality medium-rate speech coding, at reasonable complexity.

REFERENCES

[1] J. M. Tribolet and R. C. Crochiere, "Frequency Domain Coding of Speech", IEEE Trans. ASSP, Vol. ASSP-27, pp. 512-530, Oct. 1979.

[2] D. J. Esteban and C. Galand, "Application of Quadrature Mirror Filters to Split Band Voice Coding Schemes", Proc. 1977 Int'l IEEE Conf. on ASSP, pp. 191-195.

[3] C. Galand and D. J. Esteban, "16 Kbps Real-Time QMF Subband Coding Implementation", Proc. 1980 Int'l IEEE Conf. on ASSP, pp. 332-335.

[4] Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. Comm., Vol. COM-28, pp. 84-95, July 1980.

[5] R. M. Gray, "Vector Quantization", IEEE ASSP Magazine, Apr. 1984, pp. 4-29.

[6] H. Abut and S. A. Luse, "Vector Quantization for Subband Coded Waveforms", Proc. 1984 Int'l IEEE Conf. on ASSP, No. 10.6.

[7] A. Gersho, T. Ramstad and I. Versvik, "A Fully Vector Quantized Subband Coder with Adaptive Codebook Allocation", Proc. 1984 Int'l IEEE Conf. on ASSP, No. 10.7.

[8] G. Mensa, R. Montagna and F. Rusina, "Comparison between Vector and Scalar Quantization in Variable Rate Subband Coders", Proc. 1986 Int'l IEEE Conf. on ASSP, pp. 2383-2385.

[9] J. - H. Chen and A. Gersho, "Gain Adaptive Vector Quantization for Medium-Rate Speech Coding", Proc. 1985 Int'l IEEE Conf. on Comm., pp. 1456-1460.

[10] N. S. Jayant and P. Noll, "Digital Coding of Waveforms - Principles and Applications to Speech and Video", Englewood Cliffs, NJ : Prentice-Hall, 1984. Chapters 11 and 12.

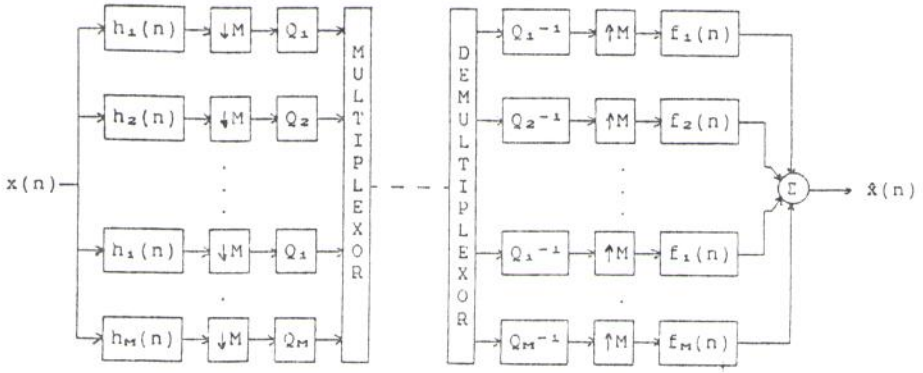

Fig. 1: Block diagram of an SBC system.

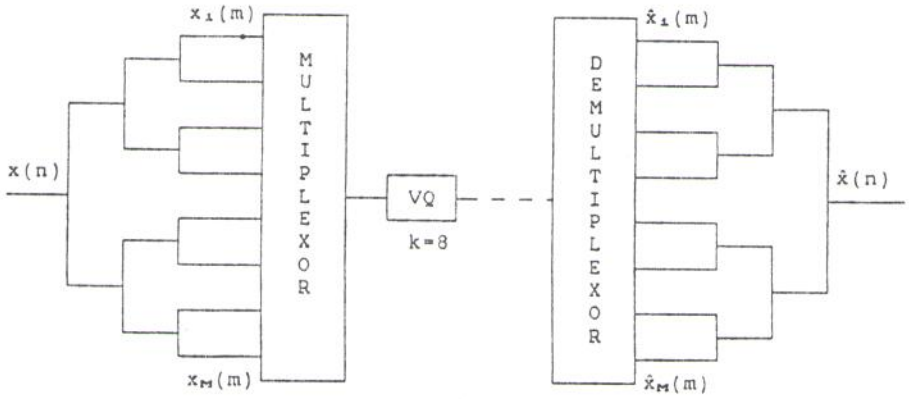

Fig. 2: SBC system with "vertical" vector quantization.

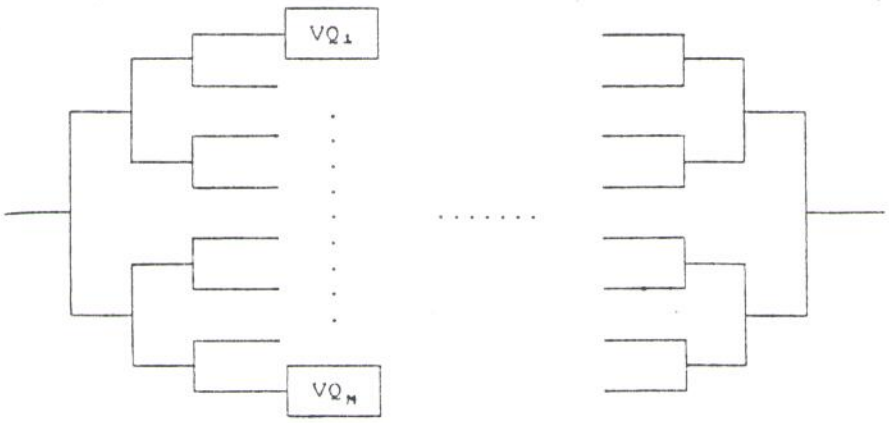

Fig. 3: SBC system with "horizontal" vector quantization.