Vector Quantization: A Pattern-Matching Technique for Speech Coding

ALLEN GERSHO AND VLADIMIR CUPERMAN

VECTOR QUANTIZATION (VQ), a new direction in source coding, has recently emerged as a powerful and widely applicable coding technique. It was first applied to analysis/synthesis of speech, and has allowed Linear Predictive Coding (LPC) rates to be dramatically reduced to 800 b/s with very slight reduction in quality, and further compressed to rates as low as 150 b/s while retaining intelligibility [1,2]. More recently, the technique has found its way to waveform coding [3-5], where its applicability and effectiveness is less obvious and not widely known.

There is currently a great need for a low-complexity speech coder at the rate of 16 kb/s which attains essentially "toll" quality, roughly equivalent to that of standard 64-kb/s log PCM codecs. Adaptive DPCM schemes can attain this quality with low complexity for the proposed 32 kb/s CCITT standard, but at 16 kb/s the quality of ADPCM or adaptive delta modulation schemes is inadequate. More powerful methods, such as subband coding or transform coding, are capable of producing acceptable speech quality at 16kb/s but have a much higher implementation complexity.

The difficulty is further compounded by the need for a scheme that can handle both speech and voiceband data at the 16 kb/s rate. These two types of waveforms occupy the same bandwidth in the subscriber loop part of the telephone network, yet they have a widely different statistical character. Effective speech coding at this rate must be geared to the specific character of speech and must exploit our knowledge of human hearing. On the other hand, a waveform that carries data must be coded and later reconstructed so that a modem can still extract the data with an acceptably low error rate. This is purely a signal processing operation not involving human perception. Vector quantization appears to be a suitable coding technique which caters to this dual requirement. VQ may become the key to 16 kb/s coding; it may also lead to improved quality waveform coding at 8 or 9.6 kb/s.

In this paper, we review recent results obtained in waveform coding of speech with vector quantization and describe some new developments not previously published. Although VQ has not yet become competitive, extrapolation of some of the results reviewed here suggests that it will indeed become a technique of major importance in waveform coding of speech. For a broader and more comprehensive review of VQ, see Gray [19].

This work was supported in part by the Air Force Office of Scientific Research under grant AFOSR 82-008.

This is an expanded version of a paper presented by the authors at GLOBECOM '83.

Vector PCM and Pattern Matching

The starting point and benchmark for all speech coding schemes is conventional PCM, the direct digitization of the speech waveform, sample by sample. PCM is based on the separate digital coding of each sample with linear, logarithm, or perhaps some optimized spacing of quantization levels. Vector quantization, when applied to a block of samples of a waveform, offers a direct generalization of PCM to what might be called "Vector PCM" or VPCM. In VPCM, a block of consecutive samples forms a vector that is treated as one entity. The vector is encoded with a binary word, and an approximation to the original vector is generated using only this binary word. The essential operation here is the quantization of a random vector. In other applications of VQ, the components of the vector may be parameters extracted from a signal rather than the signal samples themselves. In VPCM, the vector specifically represents a block of consecutive samples.

Regardless of the nature of the vector, the VQ coding process is a pattern matching technique. Each vector is encoded by comparison with a set of stored reference vectors, known as codevectors or *patterns*. Each pattern will be used to represent input vectors that are somehow identified as "similar" to this pattern. The best matching pattern in the *codebook*, the set of stored reference patterns, is selected by the encoding process according to a suitable fidelity measure, and a binary word is used to identify this pattern in the "codebook" of patterns. An example of a VPCM codebook for speech waveform coding is illustrated in Fig. 1, where each of 64 patterns consists of six samples joined by straight line segments and the various patterns are superimposed. Note that the patterns describe a variety of possible shapes for a waveform segment of this length. The patterns were optimized to allow the lowest mean-square error matching to

0163-6804/83/1200-0015 $01.00 © 1983 IEEE

ZTE EXHIBIT 1016

speech waveforms subject to the limitation that only 64 patterns could be included in the codebook. The codebook size (number of patterns) is a critical parameter which determines the encoding complexity needed for searching through the codebook, the memory required to store the codebook in both transmitter and receiver, and the bit rate for digital transmission.

Let N be the number of patterns in the codebook, k the dimension of the vector, and r the (coded) transmission rate in bits/sample. A sampling rate of 8 kHz is assumed throughout this article. These three parameters are related by:

$$r = \frac{\log_2 N}{k}$$

Thus, for the codebook of Fig. 1, $k = 6$, $N = 64$, and $r = 1$ bit/sample. The signal-to-quantization noise ratio (SNR) for VPCM in the high rate region is given by the relation:

$$\text{SNR} = 6\,\frac{\log_2 N}{k} + C_k \qquad (1)$$

where C_k is a constant depending on the dimension k, and SNR and C_k are expressed in dB units. This relation is based on the asymptotic theory when the number of patterns, N, is sufficiently large [6].

According to (1), the SNR for VPCM increases approximately at the rate of $6/k$ dB for each doubling of the codebook size (for each additional bit used to code the entire vector), where k is the dimension of the vector. Note that (1) includes as a special case the well-known formula for scalar (one-dimensional) quantization ($k = 1$) [7]. By considering SNR as a function of bit-rate, we see that the same trade-off occurs as in scalar PCM, with a 6 dB increase in SNR per unit increase in rate (bits/sample). However, the advantage of VPCM is that its constant term, C_k, is higher since the *correlation* between components of the vector is now exploited. This, in fact, is the chief virtue of VQ: it exploits, in an optimal manner, correlations within a vector of data values. *The constant term C_k increases with the dimension*, approaching the ultimate rate-distortion limit for a given source. The fact that VQ can, in principle, approach the ultimate performance limit is a consequence of Shannon's coding theorem, which is based on the concept of a block source code. Of more practical interest is the fact that a significant gain over scalar PCM is obtained in low dimensions. For example, from experiments with speech signals, we have found that C_2 is larger than C_1 by more than 3 dB and C_8 is larger than C_1 by more than 7 dB. These observations agree with earlier results reported in [8]. VPCM is clearly an attractive improvement over standard PCM. It appears to offer the potential for robustness to changes in signal statistics including power level, but it needs a more complex implementation than scalar PCM.

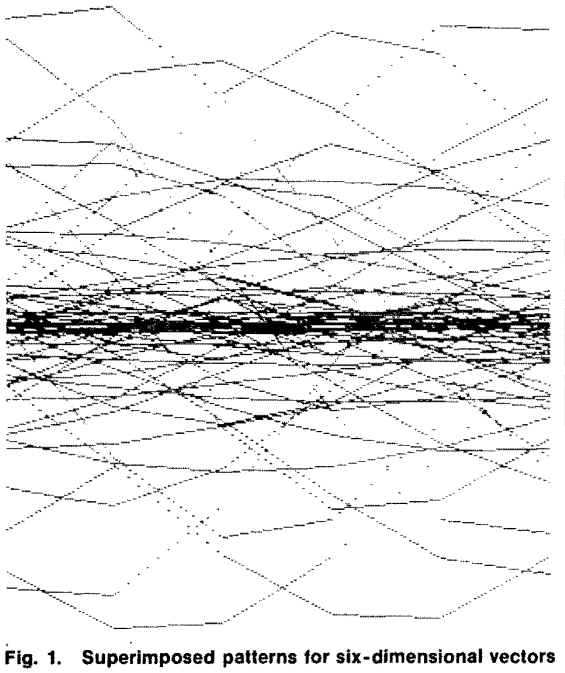

Fig. 1. Superimposed patterns for six-dimensional vectors at rate 1 bit/sample.

The Complexity Issue in Pattern Matching

Implementation complexity in signal processing hardware is typically estimated by the amount of computation or "number crunching" needed per unit of time. Computational complexity is highly processor dependent and is difficult to estimate without consideration of particular architectures. Nevertheless, we can make some preliminary assessments by assuming that multiplications are the most demanding basic operations needed. Although additions can often require equal execution times, the count of multiplications still give a useful order of magnitude estimate of complexity.

Another aspect of implementation complexity is memory. Although the capacity of read-only memories (ROM's) is increasing and their cost decreasing, the codebook storage requirements in VQ can still be very demanding. Thus, in addition to computational complexity, an important consideration is the memory size needed for a particular implementation.

The main obstacle to using VPCM for high dimensional vectors is the exponential growth of encoding complexity with the dimension for a given rate. A vector quantizer of dimension k for a rate of r bits/sample requires a number of computations per sample of the order 2^{kr} and a memory of $k\,2^{kr}$ words if a full search through the codebook is performed for each input vector. (Compare Figs. 2 and 3.)

For these reasons, the direct use of VQ of speech waveforms with optimal codebooks has been limited to dimension-rate products (kr) of 8 or less. For example, at 2 bits/sample and dimension 4, Abut et al. [1] obtained an SNR in the range of 12–14 dB, and at 1 bit/sample and dimension 8, SNR values in the range of 8–10 dB have been obtained. This performance is certainly better than what can be offered by scalar PCM at the same rate. Nevertheless, VPCM is not competitive with the performance achieved by

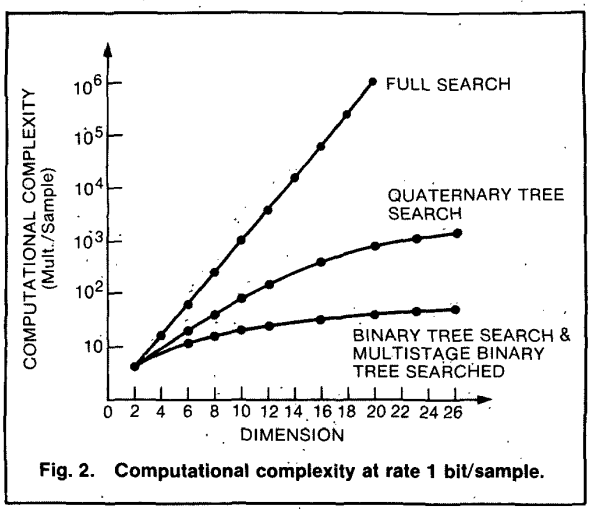

Fig. 2. Computational complexity at rate 1 bit/sample.

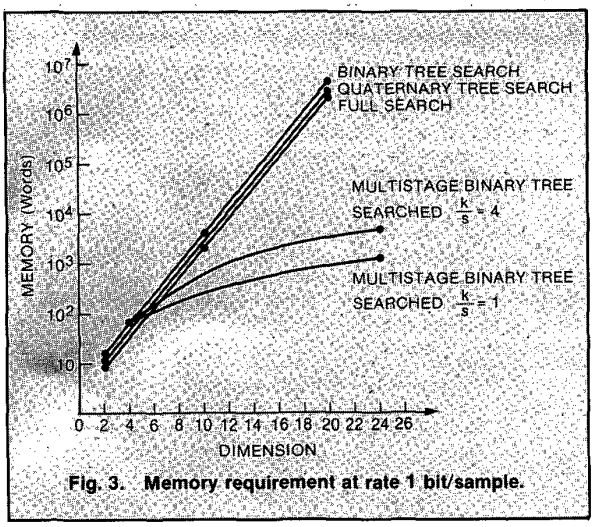

Fig. 3. Memory requirement at rate 1 bit/sample.

more sophisticated scalar speech compression methods such as subband coding, transform coding, or even ADPCM.

Several attempts have been made to circumvent the complexity problem. All are very recent developments and, based on the increasing activity in this area, we may expect to see important advances in the next few years. The approaches being taken fall into two general categories: The first is to use specially structured codebooks which reduce the search and/or memory complexity to make higher dimensional coding feasible. In the second approach, VQ is imbedded as a building block in some coding system which uses other redundancy reducing techniques that have proven effective prior to the advent of vector coding.

Reduced Complexity Vector Quantization Techniques

Several techniques have been proposed to circumvent the complexity problem yet retain the basic VPCM feature that each input vector is separately encoded, independently of prior or subsequent vectors. They generally allow higher dimensionality vector coding for the same complexity, but usually yield a performance inferior to what would be achieved with a direct optimal codebook design for the given dimension and bit rate.

Tree-Structured Codebooks

Tree-structured vector quantizers, first used for LPC [1], greatly reduce the encoding complexity at the expense of a need for more memory and a lower SNR performance than obtained with an optimal "full-search" codebook. The encoding process in a binary tree search is similar to the "successive approximation" method of analog-to-digital conversion which is routinely used in PCM encoders. Each intermediate decision supplies information which determines what test is performed at the next step.

In tree-structured VQ, a large part of the set of stored patterns in the codebook are used only to guide the search. For example, in a binary tree codebook of size 2^{kr}, the search consists of kr binary decisions; only at the last decision stage is the input vector compared with two patterns which are actual candidate codevectors to represent the input vector. The computational complexity for tree-structured codebooks increases only linearly with the dimension. On the other hand, the memory required may be up to twice as much as for standard full-search VQ.

The tree structure is not optimal because of the structural constraints imposed on the location of the pattern vectors in the final stage of the tree. Gray and Abut [8] compared the performance of standard and tree-structured VQ for speech waveforms and found a degradation of about 1 dB in SNR as a result of the tree structure. Increased memory is a major limitation. For example, in dimension 6 at 2 bits/sample, the memory is 48 000 words; in dimension 7 at the same rate, the memory increases to 224 000 words (assuming a binary tree in both cases).

Multistage Vector Quantizers

Multistage vector quantization, first proposed for quantization of LPC coefficients by Juang [9], is a technique that reduces memory as well as encoding complexity. Here, the encoding error of a vector quantizer is formed by taking the difference between the original and quantized vectors and then feeding it into a second-stage vector quantizer. This is illustrated in Fig. 4. Note that the actual binary word that must be transmitted consists of the concatenation of the binary words needed to identify the correct pattern in each

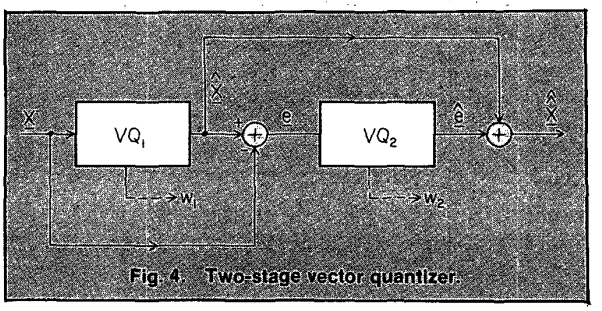

Fig. 4. Two-stage vector quantizer.

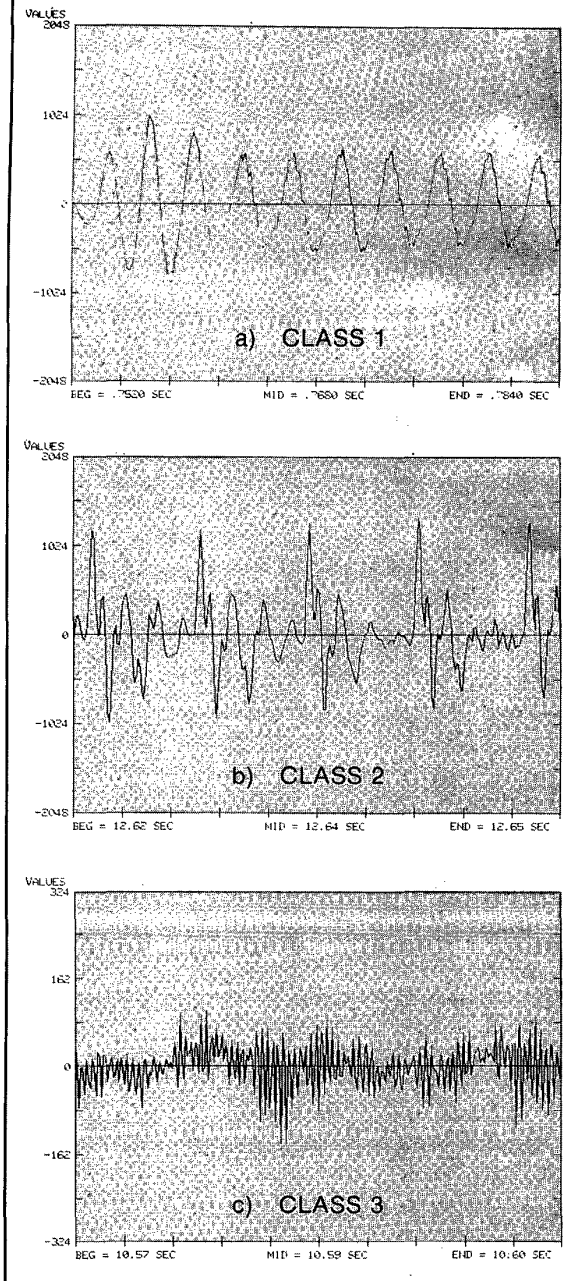

Fig. 5. Typical examples of speech waveforms for the three classes.

codebook. The process can be repeated by feeding the second stage error into a third stage vector quantizer, and so on. Note that this technique is equivalent to a single stage VQ whose codevectors are specially structured. For example, a two-stage VQ with N_1 and N_2 patterns in the first and second stages, respectively, is equivalent to a particular single stage VQ with a codebook of N_1N_2 patterns.

With this approach, both the complexity and memory can be greatly reduced in comparison with single stage vector quantization at the same rate. If s stages are used and the vector dimension is increased in proportion to the number of stages, thus keeping k/s constant, the complexity only increases quadratically with the dimension (that is, increasing as k^2) rather than exponentially as in one-stage VQ. The encoding complexity can be further reduced by using tree-structured codebooks for each stage.

Figures 2 and 3 present, respectively, the computational complexity and required memory versus dimension for different versions of suboptimal vector quantization and for one-stage optimal vector quantization, at a rate of 1 bit/sample. Note that the combination of multistage and binary tree searched coding requires the same encoding complexity but much less memory than the binary tree searched coder without use of multiple stages. One can see from these figures that binary-structured multistage VQ is the only procedure which achieves affordable complexity at higher dimensions.

We have compared the use of standard full-search vector quantizers with multistage vector quantizers for coding speech waveforms. The results show that performance may be improved by increasing the dimension for a given rate while holding the ratio k/s constant and hence keeping the complexity manageable. For example, at a rate of 1 bit/sample the SNR increases from 9.6 dB for standard full-search VQ with dimension 4 to 10.7 dB for multistage VQ with dimension 12.

The performance of a full-search vector quantizer can be improved via the multistage technique at the expense of only a moderate increase in complexity. However, the attainable improvement is quite limited since the increase in SNR with dimension saturates very quickly. This effect is due to a combination of two opposing forces. On the one hand, the performance of multistage VQ tends to decrease as more stages are used, due to greater structural constraints on the equivalent single-stage codebook. On the other hand, increasing the dimensionality tends to improve performance since more correlation in the waveform is being exploited to reduce redundancy.

In summary, tree-structured and multistage codebooks make it possible to implement higher dimensional vector quantizers at relatively low complexity. However, because of the constraints imposed on the codebook structures, they achieve only very limited improvements in performance for a given level of complexity.

Gain-Shape Vector Quantizers

Another way to reduce both memory and encoding complexity in waveform coding is to normalize the energy level ("gain") of a vector prior to encoding and to separately encode the gain with a scalar quantizer. The energy-normalized vector is called the "shape." This technique was studied by Sabin and Gray [5], who found an optimal way to jointly design the gain and shape quantizers. The effectiveness of gain-shape VQ depends on the degree to which the gain of a randomly selected speech vector is statistically independent of its shape. With this technique, higher dimensional vectors can be used with tolerable complexity but

the optimality is compromised. For example, at a rate of 1 bit/sample (8 kb/s) the highest "affordable" dimension for standard VQ is $k = 8$ and the SNR obtained is 9.7 dB. By using gain-shape VQ with dimension 12, an improvement of 0.7 dB was obtained at the same level of complexity. Again, only a very limited improvement is achieved. The complexity problem is a tough nut to crack!

Fast Nearest-Neighbor Search Algorithms

In standard full-search VQ, the optimal encoding procedure for a given codebook requires a search process through the codebook to determine the nearest codevector to the given input vector. Since the codebook is designed by a clustering algorithm such as the LBG algorithm [10], the codevectors are at irregularly located points in k-dimensional Euclidean space. It has generally been assumed that an *exhaustive* search through such an unstructured codebook is necessary to find the best matching pattern or so-called "nearest neighbor" to a given input vector. The resulting exponential growth in computational complexity with the dimension for a fixed rate is the main obstacle to approaching the Shannon limits in source coding. For a fixed rate, this implies that computation grows exponentially with k, thus preventing a close approach to Shannon's source coding bound.

In recent work, we have found a way to greatly reduce encoding complexity without degrading performance by generating a tree-structured search algorithm specifically tailored for a given optimal codebook. This gives a successive approximation method that will always lead to the best matching pattern, the so-called "nearest neighbor" to the input vector [11]. The approach, first introduced in [12], is based on an intensive one-time precomputation which results in an efficient nonuniform tree structure. This allows simple binary tree searching that eliminates large groups of codewords in each step and rapidly finds the nearest-neighbor codeword. Preliminary computational results confirm the efficiency of this approach. It is important to note that this approach does not compromise the optimality of the original codebook. An optimal "full-search" codebook is first designed and then the precomputation is used to find an efficient way to speed up the search process in performing the encoding operation. In contrast, the tree-structured codebook designs previously studied do, in fact, lead to suboptimal codebooks [1].

This work does not imply that we can now achieve Shannon limits in practice. A major remaining obstacle is the exponential growth in memory (with dimensionality) required for codebook storage. Some additional memory is also needed to store the binary decision tree for the encoding algorithm.

Imbedded Vector Quantization Methods

Just as other waveform coding systems such as DPCM use one dimensional quantizers as building blocks, so also can we use vector quantization in a coding system that exploits other methods for removing redundancy. The idea of using a vector quantizer as a building block in waveform coding systems may lead to new and powerful waveform coding devices for rates such as 16 kb/s and 9.6 kb/s with important applications to digital voice systems including secure voice and voice-mail.

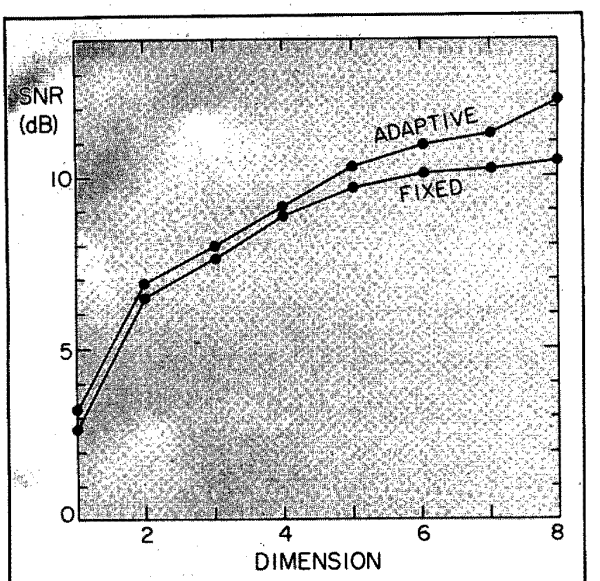

Fig. 6. Performance of fixed vs. adaptive vector quantization.

Adaptive Vector Quantization

The speech waveform is generally not stationary but local stationarity may be assumed within "frames" of duration 10–20 ms. Ideally, one would like to adapt the codebook to the changing statistics of the waveform by "learning" the particular character of the incoming speech waveform, and updating the patterns in the codebook to suit the current talker. Recently, one study was reported using this extremely complex approach for very-low-rate analysis/synthesis coding [13]. The problems related to use of adaptive codebooks for waveform coding are the large amount of side information needed to transmit the new codebook vectors to the receiver, and the complexity and processing delays related to estimation of the new codevectors.

We have experimented with a simpler adaptive scheme using a speech classifier which computes some simple statistics for each frame of speech and selects one of K classes to characterize the frame. The vectors that make up the frame are then coded with one of K codebooks corresponding to the selected classification. The class identification is transmitted to the receiver as side information once for each frame of speech. Hence, for reasonable frame lengths and a moderate number of classes, the increase in bit rate due to the side information is negligible.

We used a specific three-way classifier, described in [4], which computes the energy and the unit-lag autocorrelation and selects the appropriate class based on these two values. In the first class, the waveform is highly correlated, has high

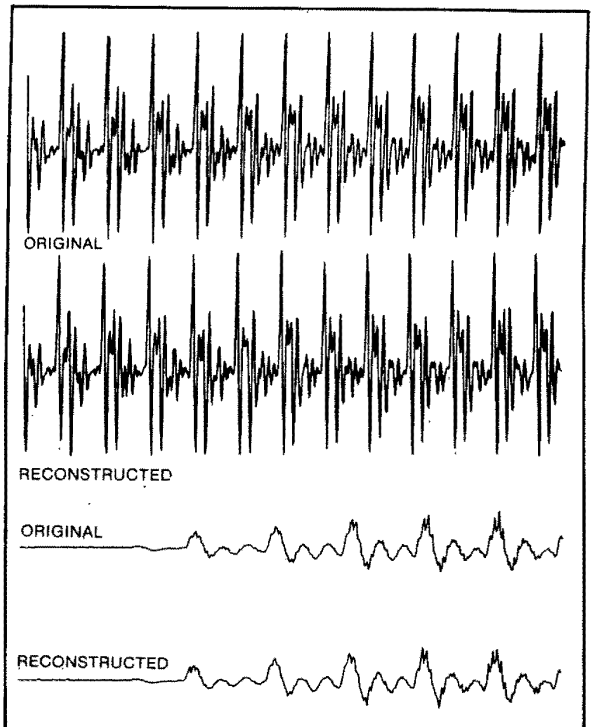

Fig. 7. Examples of original and reconstructed waveforms for adaptive vector predictive coding at 2 bits/sample.

energy, and generally contains voiced speech; it may also contain certain types of unvoiced sounds such as plosives. The second class consists of moderately correlated and high-energy segments. Voiced speech with a strong first formant is a good example of this class. Finally, the third class is made up of low-correlation or low-energy frames and generally consists of unvoiced sounds. Examples of waveforms for the three classes are shown in Fig. 5.

Some useful improvements in SNR (in one case as high as 2.5 dB) were obtained with a rate increase of only 0.03 bit/sample by using this form of adaptive VQ rather than standard VQ with the same dimension and rate. Figure 6 presents the performance of this simple adaptive VQ versus dimension at a rate of 1 bit/sample compared with standard VQ where a fixed codebook is used for every vector.

Adaptive Vector Predictive Coding

Another imbedded approach, AVPC, begins with a generalized DPCM coder by using vector prediction to remove the redundant information from each new input vector, and VQ to encode the unpredictable residual vector [4,14]. To improve performance, the system is made adaptive by classifying input frames of speech into categories and using an appropriate predictor and quantizer for each class. Examples of original speech segments and the reconstructed waveforms after AVPC coding at 2 kb/s are shown in Fig. 7. SNR performance in the range of 18–20 dB has been obtained with this technique for dimension 5 and rate 2 bits/sample (16 kb/s). These SNR values are substantially higher than those achieved by other VQ approaches and are competitive with the best known results for waveform speech coding at a similar level of complexity.

Subband Coding Using Vector Quantization

Subband coding has proven a powerful technique for medium rate speech coding. Its effectiveness is based on a strategy of allocating bits to the different frequency bands according to perceptual criteria. Performance can be further improved by adaptive bit allocation based on the instantaneous power in the subbands [15]. In one recent study, VQ was used to code the LPC spectral information allowing an efficient description of the side information for adaptive bit allocation [1]. Another recent development is a fully vector-quantized subband coder where the side information is generated by vector coding the short-term subband power levels, and VQ is used on each subband with codebooks of appropriate size allocated by the side information [17]. Vector quantization was found to cut in half the bit rate needed for side information, and to distinctly improve perceptual quality by vector coding the subband signals without increasing the bit rate of the main information. With the addition of Time Domain Harmonic Scaling (TDHS), a technique used effectively in reducing the rate of scalar quantized subband coding [18], this approach may lead to improved coders at 9.6 kb/s.

Conclusion

Vector quantization is an exciting new development that is still in its infancy as a coding technique for speech waveforms. We expect to see an increased research effort in the next few years in speech waveform coding based on VQ. Hardware development of competitive speech codecs at 16 kb/s employing this fundamental and versatile technique should not be very far away.

References

[1] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, and Sig. Process.*, ASSP-28, no. 5, pp. 562–574, Oct. 1980.

[2] S. Roucos, R. Schwartz, and J. Makhoul, "Vector quantization for very-low-rate coding of speech," *Conf. Rec.*, 1982 IEEE Global Communications Conf., pp. E6.2.1–E6.2.5, Miami, FL, Nov. 29–Dec. 2, 1982.

[3] H. Abut, R. M. Gray, and G. Rebolledo, "Vector quantization of speech and speech-like waveforms," *IEEE Trans. Acoust., Speech, and Sig. Process.*, ASSP-30, pp. 423–436, June 1982.

[4] V. Cuperman and A. Gersho, "Adaptive differential vector coding of speech," *Conf. Rec.*, 1982 IEEE Global Communications Conf., pp. E6.6.1–E6.6.5, Miami, FL, Nov. 29–Dec. 2, 1982.

[5] M. J. Sabin and R. M. Gray, "Product code vector quantizers for speech waveform coding," *Conf. Rec.*, 1982 IEEE Global Communications Conf., pp. E6.5.1–E6.5.5, Miami, FL, Nov. 29–Dec. 2, 1982.

[6] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, IT-25, no. 4, pp. 373–380, July 1979.

[7] A. Gersho, "Quantization," *IEEE Communications Magazine*, vol. 15, pp. 16–29, Sept. 1977. (Reprinted as "Principles of quantization," *IEEE Trans. Circuits Syst.*, CAS-25, no. 7, July 1978.)

[8] R. M. Gray and H. Abut, "Full-search and tree-searched vector quant. of speech waveforms," *Proc. Intl. Conf. on Acoust., Speech, and Sig. Process.*, pp. 593–596, Paris, 1982.

[9] B. H. Juang and A. H. Gray, Jr., "Multiple stage vector quantization for speech coding," *Proc. Intl. Conf. on Acoust., Speech, and Sig. Process.*, pp. 597-600, Paris, 1982.
[10] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, COM-28, pp. 84-95, Jan. 1980.
[11] A. Gersho and D.-Y. Cheng, "Fast nearest-neighbor search for Euclidean codes," Abstract presented at IEEE Intl. Symp. on Inform. Theory, Ste. Jovite, Canada, Sept. 1983.
[12] A Gersho, "On the structure of vector quantizers," *IEEE Trans. Inform. Theory*, IT-28, no. 2, pp. 157-166, March 1982.
[13] D. Paul, "A 500-800 b/s adaptive vector quantization vocoder using a perceptually motivated distance measure," *Conf. Rec.*, 1982 IEEE Global Communications Conf., pp. E6.3.1-E6.3.5, Miami, FL, Nov. 29-Dec. 2, 1982.
[14] V. Cuperman and A. Gersho, "Vector predictive coding of speech at 16 kb/s," to be published.
[15] T. Ramstad, "Subband coder with a simple adaptive bit-allocation algorithm: a possible candidate for digital mobile telephony?," *Proc. Int. Conf. on Acoust., Speech, and Sig. Process.*, pp. 203-207, Paris, May 1982.
[16] C. D. Heron, R. E. Crochiere, and R. V. Cox, "A 32-band subband/transform coder incorporating vector quantization for dynamic bit allocataion," *Proc. Intl. Conf. on Acoust., Speech, and Sig. Process.*, pp. 1276-1279, Boston, MA, April 1983.
[17] A. Gersho, T. Ramstad, and I. Versvik, "Fully vector-quantized subband coding with adaptive codebook allocation," to be published.
[18] R. E. Crochiere, R. V. Cox, J. D. Johnston, and L. A. Seltzer, "A 9.6-kb/s DSP speech coder," *Bell Syst. Tech. J.*, vol. 61, no. 9, November 1982.
[19] R. M. Gray, "Vector quantization," to be published in *IEEE ASSP Mag.*, 1984.

Allen Gersho is a professor of Electrical and Computer Engineering at the University of California, Santa Barbara, where his current research focuses on vector quantization theory and speech and image coding. He was at Bell Laboratories from 1963 to 1980, where he was engaged in research in digital communications, data transmission, and signal processing.

Dr. Gersho has served as editor of *IEEE Communications Magazine* and as associate editor of the *IEEE Transactions on Communications*. He received the Guillemin-Cauer Prize Paper Award from the Circuits and Systems Society in 1980. He is a member of the Board of Governors of the IEEE Communications Society. Dr. Gersho received his B.S. from MIT in 1960 and Ph.D. from Cornell University in 1963.

Vladimir Cuperman was born in Bucharest, Romania on April 9, 1938. He received the B.S. degree in electrical engineering (communications) from the Polytechnic of Bucharest in June, 1960 and the M.A. degree in mathematics from the University of Bucharest in June, 1972. He received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara in March 1981 and June 1983, respectively.

From 1963 to 1974, he worked at the Institute for Automation, Bucharest, on data transmission and error correcting codes. Since 1974, he has worked at Tadiran, Israel, on digital communications, speech, and image coding. From January 1981 to March 1983 he was teaching assistant, research assistant, and research associate at UCSB. ■

CALL FOR PAPERS

Special Issue on Optical Communications
IEEE Communications Magazine

The *IEEE Communications Magazine* will be publishing a special issue on Optical Communications in May 1985. The Guest Editors will be Dr. Alvin B. Glenn of Florida International University and Dr. Gerd Keiser of GTE.

Original papers, surveys, and tutorial articles concerning optical communications are solicited for this special issue. Possible topics include, but are not limited to:

- historical review
- optical fibers
- lightwave detectors
- modulation techniques
- measurement techniques
- performance evaluation
- commercial communications systems
- economic and performance comparisons
- technology growth projections

The papers should be in keeping with the general spirit of the *Magazine*, minimizing detailed theoretical derivations and analyses.

A 100-200 word summary/abstract should be sent to either Guest Editor at the addresses given below by July 1, 1984. This should be followed by five copies of the proposed article by October 15, 1984.

Professor Alvin B. Glenn
Department of Electrical Engineering
Florida International University
Tamiami Campus
Miami, FL 33199
Tel. (305) 554-3017/2807

Dr. Gerd Keiser
GTE Communications Systems Division
77 A Street
Needham Heights, MA 02194
Tel. (617) 449-2000, ext. 587