Apple Computer Inc. v. Burst.com, Inc. Doc. 108 Att. 13 Case 3:06-cv-00019-MHP Document 108-14 Filed 06/07/2007 Page 1 of 10

Exhibit K

Dockets.Justia.com

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-30, NO. 4, APRIL 1982

The Design of Trellis Waveform Coders

LAWRENCE C. STEWART, MEMBER, IEEE, ROBERT M. GRAY, FELLOW, IEEE, AND YOSEPH LINDE, MEMBER, IEEE

Abstract—New algorithms for the design of trellis encoding data compression systems are described. The main algorithm uses a training sequence of actual data from a source to improve an initial trellis decoder. An additional algorithm extends the constraint length of a given decoder. Combined, these algorithms allow the automatic design of a trellis encoding system for a particular source. The algorithms' effectiveness for random sources is demonstrated through performance comparisons with other source coding systems and with theoretical bounds. The algorithms are applied to the practical problem of the design of trellis and hybrid codes for medium-to-lowrate speech compression.

INTRODUCTION

In the past decade considerable progress has been made towards an understanding of tree and trellis encoding data compression systems. In these discrete-time or sampled-data systems (Fig. 1), a deterministic but possibly nonlinear decoding filter transforms the channel sequence into a reproduction sequence. The channel sequence is chosen by a tree or trellis search encoding algorithm to minimize the distortion between the reproduction sequence and the source sequence. Results in information theory have proved the existence of tree and trellis systems which operate close to the theoretical limits of performance, but these papers offer only existence proofs, without describing ways of actually constructing good codes [20], [15].

The study of encoding algorithms for tree and trellis codes has been of interest since the earliest work on convolutional channel coding. The Viterbi algorithm [11] is optimum for searching the trellis structures associated with finite-state decoders, but has computational cost exponential in the constraint length of the code. Some of the nonoptimal algorithms, such as the Fano and stack algorithms, have been derived from problems of decoding error correcting codes; others have been designed specifically for the tree source encoding problem. The M, L algorithm [21], for example, is simply a breadthfirst tree search. Other search algorithms are classed as depthfirst or metric-first (distortion-first) types [2], [3].

Because many effective encoding algorithms are known, the more difficult part of the design of a tree or trellis coding system lies in the design of the decoder. Decoders may be drawn

Manuscript received June 2, 1981; revised September 29, 1981. This paper was presented in part at the IEEE International Symposium on Information Theory, Santa Monica, CA, February 1981. This work was supported in part by the Joint Services Electronics Program under Contract DAAG-0047, by the U.S. Army Research Office under Contract DAAG29-80-K-0073, and by the Xerox Corporation.

L. C. Stewart is with the Information Systems Laboratory, Stanford University, Stanford, CA 94305, and the Computer Science Laboratory, Xerox Palo Alto Research Center, Palo Alto, CA 94304.

R. M. Gray is with the Information Systems Laboratory, Stanford University, Stanford, CA 94305.

Y. Linde is with the Codex Corporation, Mansfield, MA 02048.



from any of the traditional waveform coding techniques, such as predictive quantization. Such *plagiarized decoder* systems achieve improved performance by replacing their traditional encoders by a tree or trellis search algorithm. In the literature, various other methods for selecting decoders have been proposed. For speech sources, for example, attempts have been made at the design of tree decoders based on predictive quantizers which match the average correlation properties of speech [4]. Variational methods have been proposed to improve decoders based on predictive quantizers and Linde and Gray have used the notion of a fake process to suggest decoders [5], [24]. While most of these codes based on traditional systems incorporate structures similar to recursive digital filters and thus have such a large number of states that they generate tree codes, several investigators have used decoders incorporating either transversal digital filter approximations of recursive filters or decoders originally based on finite impulse response models [23]. These decoders have a relatively small number of states and, therefore, a trellis structure.

In this paper we describe new algorithms for the design of trellis encoding data compression systems. The main algorithm uses a training sequence of actual data from a source to improve an initial trellis decoder. An additional algorithm extends the constraint length of a given decoder. Combined, these algorithms allow the automatic design of a trellis encoding system for a particular source. The decoders designed with these techniques are locally optimal and, we conjecture, globally optimal for certain sources [17]. The algorithms' effectiveness for random sources is demonstrated through performance comparison with other source coding systems and with theoretical bounds. The algorithms are also applied to the practical problem of the design of trellis and hybrid codes for medium-to-low-rate speech compression. Additional information, including a Soundsheet with the speech coding results, may be found in [34].

The goal of this paper is not the presentation of particular coding results, but the presentation of design methods which offer the potential of improving the performance of any existing trellis coder—or of constructing a new coder for a source of entirely unknown character.

Page 3 of 10

STEWART et al : TRELLIS WAVEFORM CODERS

BLOCK SOURCE CODE DESIGN ALGORITHM

As a step towards the description of the trellis code design algorithms, we describe an algorithm for the design of block source codes, or vector quantizers, based on a long training sequence of symbols from a source. This algorithm is a multidimensional version of a quantizer design method of Lloyd [25] and is more completely reported by Linde *et al.* [22].

An N-level, k-dimensional quantizer is a function f that assigns to each input vector \mathbf{x} a reproduction vector \mathbf{x}' drawn from a finite alphabet $A = \{y_i: i = 0, \dots, N-1\}$. The function f returns the index of the selected reproduction vector or codeword. The quantizer is described by the reproduction alphabet and an encoding rule. For a particular input (training) sequence $\{x_j: j = 0, \dots, n-1\}$, the encoding rule induces a partition, $S = \{S_i: i = 0, \dots, N-1\}$, of the input sequence into the disjoint sets $S_i = \{j: f(x_j) = y_i\}$ of the time indexes of those input vectors mapping into the *i*th reproduction vector.

The fidelity of reproduction is measured by a nonnegative distortion measure d(x, x'). Many distortion measures have been proposed for various applications, but perhaps the best known is the squared-error measure

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i=0}^{k-1} (x_i - x_i')^2.$$

An obvious encoding rule maps each source vector into the reproduction vector giving minimum distortion. A tie breaking rule is necessary but since ties are generally low probability events, nearly any rule will do.

The initial conditions for the design algorithm are N, the desired number of reproduction vectors, A^0 , an initial quantizer, and $\{x_j\}$, a long training sequence of symbols from the source. The algorithm consists of the repetition of two steps: finding the best encoding of the training sequence for a given set of reproduction vectors, and finding the best set of reproduction vectors for the given encoding.

Encode: Given A^m , the reproduction alphabet of generation m, find the minimum distortion partition $S^m = P(A^m)$ by mapping each element of the training sequence to the minimum distortion reproduction vector:

$$f(\mathbf{x}) = i: d(\mathbf{x}, \mathbf{y}_i) \le d(\mathbf{x}, \mathbf{y}_i), \quad \text{for all } j$$

with some tie breaking rule.

j

Update Reproduction Alphabet: Given a partition S^m , find the minimum distortion reproduction alphabet for generation $m + 1, A^{m+1} = A(S^m) = A(P(A^m))$, by setting y_i^{m+1} , the *i*th reproduction vector of the new alphabet, to the value giving the minimum average distortion over the training sequence vectors indexed by elements of S_i^m . This value will be the generalized centroid, or center of gravity under the distortion measure, of those training sequence values which were reproduced by the value y_i^m .

In the case of the squared-error distortion measure, this center of gravity calculation is just the sample average over a partition:

$$y_i^{m+1} = \frac{1}{||S_i^m||} \sum_{j \in S_i^m} x_j.$$

The new codeword is the average of those training sequence samples encoded by the old codeword.

Repeated application of these two steps must result in decreasing, or at least nonincreasing, sample average distortion over the training sequence. Since the average distortion is nonnegative and decreasing, it must eventually reach a limit. Although the limit may not be the global optimum, it is at least a local optimum. The algorithm may be stopped either when a fixed point is reached (when no changes occur in the reproduction vectors or partitions), or when the reduction in average distortion per iteration falls below some threshold.

If the vector source $\{x_j\}$ is ergodic and stationary, the quantizer designed by this algorithm will work as well on data from outside the training sequence as from the training sequence itself. However, operation of the design algorithm does not depend on either stationarity or ergodicity.

TRELLIS CODE DESIGN ALGORITHM

The most general case of a trellis decoder consists of a finite-state machine driving a table-lookup codebook of decoder output values. Symbols arriving from the channel drive the finite-state machine, which in turn selects reproduction symbols from the codebook. In what follows, we specialize the finite-state machine to the tradional case of a shift register containing the most recent k channel symbols (Fig. 2). The contents of the shift register are used as a table index to select the decoder output codeword. The trellis code design algorithm seeks to fill in the contents of the codebook.

Suppose that we have a table-lookup trellis decoder, a single symbol additive distortion measure, and an optimal encoding (search) algorithm such as the Viterbi algorithm which will find the trellis encoding of a source sequence giving the lowest possible sample average distortion. The symbols of the channel sequence, or path map, chosen by the encoding algorithm associate the sequence of source symbols with the reproduction sequence of codewords. This association can be used as the partition function in a trellis adaptation of the block quantizer design algorithm. (See Table I.)

The initial conditions for the design algorithm are a finitestate decoder with initial codebook C^0 (such as the shift register decoder described above), a single symbol distortion measure (such as squared-error), and $\{x_j\}$, a long training sequence of symbols from some source. The algorithm consists of two key steps: finding the best encoding of the training sequence for a given codebook, and finding the best codebook for the given encoding. Executed alternately, these steps provide an iterative design algorithm for improving the initial trellis decoder.

Find Encoding: Given the trellis codebook for generation m, find the minimum average distortion encoding of the training sequence. This encoding induces a partition on the training sequence so that the elements of the partition cell correponding to a certain codeword are the time indexes of those elements of the training sequence which were reproduced by that codeword.

Find Codebook: Given the partition for generation m, find the minimum distortion codebook for generation m + 1. The updated value for a certain codeword will be the value giving

704

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-30, NO. 4, APRIL 1982

Shift Register Implementation (constraint-length 9)



Fig. 2. Shift register implementation of a one bit per symbol trellis decoder.



- (0) Initialization: Given a distortion threshold $x \ge 0$, a q-ary noiseless channel, an R state decoder, an initial codebook C^0 with cardinality $||C^0|| = N = qR$, and a training sequence $\{x_j\} = \{x_j: j = 0, \dots, n-1\}$, set m = 0.
- (1) Given $C^m = \{y_i^m: i = 0, \dots, N-1\}$, the codebook for generation m, find the minimum distortion trellis encoding $\{\hat{x}_j: j = 0, \dots, n-1\}$ of the training sequence. This encoding induces a partition on the training sequence $\{S_i^m: i = 0, \dots, N-1\}$ with $S_i^m = \{j: \hat{x}_j = y_i^m\}$. Each set $S_i^m = P(C^m, \{x_j\})$ contains the time indexes of those elements of the training sequence which were encoded by codeword y_i^m .
- (2) Compute the average distortion $\Delta^m = n^{-1} \sum_{i=0}^{n-1} d(x_i, \hat{x}_i)$.
- (3) If at least one full iteration has been completed (m > 0) and the decrease in distortion has fallen below the threshold ε, (Δ^m Δ^{m-1})/Δ^{m-1} ≤ ε, then halt with C^m as the final codebook. Otherwise go to step (4).
- (4) Find the optimal codebook C^{m+1} for generation m + 1 as $C^{m+1} = C(P(C^m, \{x_j\}))$ with $C^{m+1} = \{y_i^{m+1}: i = 0, \dots, N-1\}$ and $y_i^{m+1} = y: \Sigma_j \in S_{im} d(x_j, y) \le \Sigma_j \in S_{im} d(x_j, y')$, for all y'. Replace m by m + 1 and go to step (1).

the minimum average distortion over those elements of the training sequence indexed by the partition cell corresponding to that codeword.

The encoding function—the trellis search—does not necessarily map a source symbol into the minimum distortion codeword, but maps the entire training sequence into the minimum average distortion *sequence* of codewords.

Each iteration of this procedure can only result in decreasing, or at least nonincreasing, average distortion. Because the encoding algorithm is optimal, the new encoding can result in distortion no worse than the distortion due to using the old path with the new codewords—which, in turn, is at least as good as the distortion due to using the old path with the old codewords. If the same path is chosen twice, the algorithm has reached a fixed point and will proceed no further.

Each new codeword has the value which minimizes the average distortion over its partition cell. Each codeword in generation m + 1 will be the centroid of those elements of the training sequence which were encoded by the corresponding codeword of generation m. Some rule must be adopted to cover the case of an empty partition cell. Some approaches to this problem are to retain the old codeword, to copy the value of another codeword, or to use the centroid of the entire training sequence as the new codeword.

If the distortion measure is squared-error, $d(x, x') = (x - x')^2$, then the centroid computation for the updated codewords is particularly simple:

$$y_i^{m+1} = \frac{1}{||S_i^m||} \sum_{j \in S_i^m} x_j.$$

This expression does not work when a partition cell is empty. As mentioned, there are many ways to handle this situation; the goal of the method selected must be to select a value that will be *used* by the next pass of the encoder. Once the partition cell is no longer empty, the design algorithm will take care of adjusting the value of the associated codeword.

EXTENSION

In this section, we describe a method of constructing a decoder based on a shift register of length k + 1 from a decoder of length k. Our method has the advantage of constructing a decoder with sample average distortion over the training sequence at least as low as that of the starting (shorter) decoder. We call this method *extension* because the new, longer decoder is constructed by adding an additional stage to the shift register of the starting decoder.

Again, the decoder of a trellis encoding system is implemented by a shift register driving a table-lookup codebook. Arriving channel symbols are shifted into the least significant end of the register and the contents of the register act as the table address or codeword index to generate the decoder output. For a q-ary channel and constraint-length k decoder, let the register contents be r. With the arrival of symbol u from the channel, the new register contents will be $(qr + u) \mod q^k$ and the decoder will produce codeword $y_{(qr+u) \mod qk}$. Now suppose the shift register is extended by one stage to length k+1; the codebook must increase in size from q^k entries to q^{k+1} entries. We fill in the new codeword values by duplicating the old codebook q times so that the symbol stored in the new register stage does not affect the decoder output. Let the old codebook contain codewords $\{y_i: i = 0, \dots, q^k - 1\};$ then the extended codebook will contain codewords y':

$$y'_{i} = y'_{i+qk} = \cdots = y'_{i+(q-1)qk} = y_{i};$$

 $i = 0, \cdots, q^{k} - 1.$

This procedure produces a decoder of constraint length k + 1 whose behavior is identical to the behavior of the original constraint-length k decoder. This new decoder can be used as the initial guess for the decoder improvement algorithm.

From an initial constraint-length 1 decoder, alternate application of the decoder improvement algorithm and the extension algorithm permits the design of successively longer codes (Table II).

Extension can be illustrated by considering its application to a one bit per symbol (q = 2) system using squared error distortion.

Let the initial codebook for the constraint-length 1 decoder include codewords -1 and +1. The trellis code design algorithm for this constraint length selects a locally optimal twolevel scalar quantizer for the training sequence. Assuming that the mean of the training data is 0, the optimum partition for the final constraint-length 1 decoder collects all the samples in the training sequence with positive values in one partition cell and all the samples with negative values in the other. We have $S_0 = \{j | x_j \ge 0\}$ and $S_1 = \{j | x_j < 0\}$; ties are broken by assigning the associated sample to the first partition. The two

Case 3:06-cv-00019-MHP [

STEWART et al.: TRELLIS WAVEFORM CODERS

TABLE II			
COMBINED EXTENSION	AND	IMPROVEMENT	ALGORITHM

Let C_k^m refer to a constraint-length k codebook at the mth iteration of the decoder improvement algorithm.

- (1) Design an initial codebook C_1^0 by finding a good q-level quantizer for the training sequence.
- (2) Given C_k^0 , an initial code of a certain constraint length, use the trellis code design algorithm to improve it. The extent of this procedure will be governed by the value of ϵ in the trellis code algorithm.
- (3) If the constraint-length k is sufficiently large, or the distortion provided by C_k^m sufficiently low, then halt.
- (4) Use the extension algorithm to produce the codebook C_{k+1}^{0} from C_k^{m} and return to step (2).



Filed 06/07/2007

Fig. 3. Performance of fake process codes and trained random codes on the memoryless Gaussian source.

codeword values will be the means, respectively, of the positive and negative training sequence samples. The first extension of this decoder—to constraint length 2—contains four codewords. The original partition is refined in a very intuitive way: the partition cell containing the positive valued samples of the training data is divided into two cells containing, respectively, the positive samples which were preceded in the training data by other positive samples, and those which were not. $S_{00} = \{j | x_j \ge 0, x_{j-1} \ge 0\}$ and $S_{10} = \{j | x_j \ge 0, x_{j-1} < 0\}$ are the two subcells created from S_0 (using binary notation for the partition cell indexes).

CODING RANDOM SOURCES

The basic trellis code design algorithm guarantees only two things: it will eventually halt and the average distortion obtained over the training data will be nonincreasing with each successive iteration. These guarantees are of considerable theoretical interest, but the algorithm would be of little practical value if it either failed to produce improved codes or converged slowly. In practice, its convergence is almost always extremely rapid—a few iterations usually suffice—and performance improvements are usually obtained.

The Gaussian i.i.d. and Gauss-Markov (autoregressive) sources are common yardsticks of source coding. In this section, one bit per symbol trellis codes designed by the iterative algorithm are compared with the fake process trellis codes of Linde-Gray, with some recent results by other researchers, and with the rate distortion limits [18]. Since existing tree and trellis codes already perform quite close to the theoretical limits, there is not a lot of room for improvement, but the design algorithm is able to obtain 0.2-0.4 dB performance gains. Since the design may be accomplished off-line, these gains are obtained without additional operating complexity.

Rate-Distortion Functions

A first-order Gauss-Markov source is the output of a first-order digital filter driven by Gaussian i.i.d. symbols:

$$x_i = a x_{i-1} + \omega_i, \qquad |a| < 1.$$

where ω is distributed $N(0, \sigma^2)$. For $\sigma^2 = 1$ and sufficiently small distortion, the Shannon lower bound for this source with squared error distortion is

$$R(D) = \frac{1}{2} \lg \frac{1-a^2}{D}, \qquad D \le \frac{1-a}{1+a}$$

where D is the average squared error and R is the information rate in bits per symbol. In this formula, R is the theoretical minimum information rate required for *any* data compression system with average distortion no greater than D [8].

Memoryless Gaussian Sources

Fig. 3 compares decoders designed by the iterative design algorithm with the rate distortion bound, with the best one-bit scalar quantizer (the Lloyd-Max quantizer [31]), and with the Linde-Gray scrambling function decoder [22]. Pearlman [32] reports squared error performance of 0.303 and 0.301 for length 9 and 10 decoders, respectively, designed using a new theory for source coding with constrained alphabets. These results are very close to ours. Pearlman's decoders use only four discrete codeword values, but a fairly complex function is used to map the decoder register contents to the four codewords. Fehn and Noll [10] have obtained similar results using randomly populated trellis.

The trained decoders of Fig. 3 were designed using the Viterbi encoding algorithm for ten iterations on a training sequence of 20 000 samples. Table-lookup shift register decoders with random codewords were used as the "initial guess" for the design algorithm. The results of Fig. 3 represent the trained decoder performance on data from outside the training sequence.

From theory [8, ch. 4] we know that the reproduction values of a data compression system should have a distribution yielding the appropriate distortion-rate function (this technique is used in [10]). Since we would eventually like to use the trellis code design algorithm for sources with an unknown distribution, we may not be able to calculate the best R-D distribution for the initial codebook. For this reason, we have repeated the approach of [22] and used initial codewords drawn from the sample distribution of the source, based on the training sequence.

First-Order Gauss-Markov

Fig. 4 presents performance results of trained decoders for the first-order Gauss-Markov source with correlation coefficients between 0.35 and 0.95. Shift register decoders of constraint lengths 5 and 6 were designed by a combination of the extension algorithm and the decoder improvement algorithm starting from initial constraint-length 1 decoders with codewords +1 and -1. The resulting decoders are compared with the rate-distortion function and with the *truncated predictive quantization decoders* described in [23]. The TPQD decoders

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-30, NO. 4, APRIL 1982



Fig. 4. Performance of TPQD codes and extension codes on first-order Gauss-Markov sources.

are shift register trellis decoders obtained by truncating the impulse response of a predictive quantizer. When used with a trellis search algorithm as the encoder, the TPQD system can outperform ordinary predictive quantization. As demonstrated here, the iterative design algorithm can obtain still better performance. We note that tree-searched predictive quantizers [10] can obtain similar results for highly correlated sources.

Additional research of Gray and Linde for one bit per symbol block codes designed using the block code design algorithm yields results comparable to those reported here. For a = 0.85, a 128 codeword block code has been designed which yields an average distortion of 0.101. For a = 0.90, an average distortion of 0.073 has been obtained [18].

CODING SPEECH SOURCES

In this section, we turn to the problem of designing tree and trellis encoding systems for speech. We discuss trellis codes for the original speech waveform, trellis codes for the residual signal of linear predictive coding systems, and a tree encoding system using a hybrid decoder for the speech waveform. The results of the hybrid tree codes have been very encouraging, as these codes can provide a good quality speech at rates of one bit per sample through an automatic design procedure.

Previous Tree and Trellis Codes for Speech

The use of tree and trellis techniques for speech coding has a long history. Most of the early work was in the area we have called plagiarized decoders-the application of a tree search algorithm to the encoding problem in a standard coding system. In the literature, this technique is usually referred to as delayed decision [7], [35]. By using a tree encoder, which delays the encoding process by a number of samples in order to observe the consequences of particular encodings, delayed decision systems are able to achieve improved performance. The general consensus has been that such methods can yield improvements in signal-to-noise ratio, but that the improvements are not usually audible [12]. A second class of approaches to tree coding of speech has been the design of tree decoders based on short- or long-term correlation functions of speech [4], [19], [30]. Systems both with fixed and with adaptive decoders have been built [10], [26], [37]. More recently, a third approach has been the use of an optimization algorithm to improve an initial decoder [5]. The technique uses a variational method and a training sequence to adapt the tap weights of a transversal filter. This approach is similar in spirit to the techniques of this paper, but considers only linear decoder structures.

Digital speech data sampled at 6500 Hz were made available by Signal Technology, Inc. of Santa Barbara, CA. Since the residual excited and hybrid coding systems discussed later are associated with linear predictive coding techniques, we will also refer to the speech data as broken into LPC frames of 128 samples each—corresponding to a rate of about 50 frames/s. The code design efforts discussed below used two segments of 600 frames or about 12 s each. One segment was used as the training sequence for the design algorithms. The second segment, by a different speaker, was used as a test sequence to check the performance of designed decoders on different data. While the speech segments were not phonetically balanced or otherwise selected, they were carefully recorded in a low noise environment and with good gain control.

The speech coding systems discussed here use the M, L algorithm for both decoder design and code operation. The algorithm parameters were set so that the algorithm maintained 20 path sequences as possible encodings with an encoding delay of 31 symbols. It is widely held [12], [35] that less intensive search will achieve most of the benefits of delayed decision encoding, but our primary purpose was the *design* of speech codes, rather than a search for the most efficient encoding algorithm. Trellis decoders were designed using the extension method from small initial codebooks. At each constraint length, the design algorithm was run for at most six iterations or halted when the change in signal-to-noise ratio measured in decibels fell below 1 percent. Decoders were designed for data rates of 1/2 bit/speech sample, 1 bit/sample, and 2 bits/sample.

Speech Waveform Trellis Codes

The simplest way to apply trellis encoding to speech is to build a trellis decoder for the original speech waveform. 3250 bits/s were required for the rate 1/2 code, 6500 bits/s were required for the rate 1 code, and 13 000 bits/s were required for the 2 bits/sample code.

Fig. 5 contains signal-to-noise ratios in decibels for the various speech waveform trellis codes designed, measured outside the training sequence. The information is presented as a function of the base two logarithm of the size of the codebook. This value is equivalent to the number of address bits required by a memory implementing the codebook.

The decrease in performance of both the rate 1/2 code and the rate 1 code for large codebooks outside the training sequence indicates that training sequence was of insufficient length for the design of codebooks with more than about 512 codewords. If the training sequence is too short, the decoder becomes specialized to the particular sequence rather than the properties of the source. Audibly, as might be expected, the 3250 bit/s system is very noisy, although it is intelligible. The rate 1 systems are an improvement, although still quite noisy, and the rate 2 systems, at 13 000 bits/s, are still sufficiently noisy to be classed as "communications quality"—not quite up to the standards of long distance telephony.

707

STEWART et al.: TRELLIS WAVEFORM CODERS



Fig. 5. Speech waveform trellis code performance. Figures are decibel signal-to-noise ratio. Curve a is for the 1/2 bit per sample code outside the training sequence. Curve b is for the rate 1 code, and curve c is for the 2 bits per speech sample code. Log N is the base-2 log-arithm of the codebook size.

Linear Predictive Coding and Vector Quantization

Linear predictive coding of speech has been very successful [27]. The basic principle of the method is a decomposition of the speech signal into an excitation function, e.g., the vocal cords, and an all-pole linear filter model of the vocal tract. LPC systems operate by breaking up the speech signal into segments or frames, estimating the filter and excitation parameters for each frame, and transmitting quantized versions of the parameters to the decoder, which uses the values to operate a synthesizer. The length of a frame is typically 10-50 ms. While there are many methods for estimating the filter parameters, all seek to minimize the power of the error or residual signal [14]. The speech residuals, the result of passing the original speech waveform through the LPC analysis filter A(z) or $A(z)/\sigma$, play a role in the various methods for estimating the excitation function and are sometimes themselves quantized and transmitted.

There has been much work done on the best way to encode the LPC parameters for transmission [13]. Recent research into applications of the block quantizer design algorithm has led to vector quantization methods of encoding the gain and model parameters simultaneously [9], [33]. This method is illustrated by Fig. 6. During the design phase of the vector quantizer, a finite collection of gain/model combinations typical of speech are selected by using the block code design algorithm together with a distortion measure between the waveform segment and the filter coefficients. During operation, for each frame a parallel test of each filter is made in search for the one giving the minimum residual energy. The *index* of the filter is then transmitted.

The most difficult part of LPC encoding has been the estimation of the excitation parameters, which determine whether a speech frame is voiced or unvoiced and, if voiced, determine the pitch. While there are a plethora of methods, none has been completely satisfactory. Part of the difficulty is that



Fig. 6. Vector quantization of LPC.

there are some speech sounds like "v" that are partly voiced and partly unvoiced, and part is due to the desire that a speech coding system continue to operate in a reasonable manner in the presence of nonspeech background noise or with multiple simultaneous speakers.

In response to the problems of estimating the excitation part of the LPC decomposition, several systems have been proposed which encode either the actual LPC residual signal or some associated signal. Adaptive predictive coding (APC), the voice excited vocoder (VEV), and residual excited LPC (RELP) systems all fall into this category [6], [7], [36]. The major difficulty in each of these systems lies in the encoding of the spectrally flat residual signal while retaining a low transmission rate. A typical method used is to band limit the residuals and then to use some form of adaptive PCM to transmit the resulting signal. At the receiver, a nonlinearity is used to restore some energy to higher frequencies and the resulting signal is used to excite a standard LPC synthesis filter. Tree and trellis encoding offers the potential of transmitting the residual signal at low rates without resorting to these downsampling and spectral extension techniques.

LPC with Trellis Encoded Residuals

Transmission of a coded form of the residuals, combined with the model or model and gain portions of an LPC system, should have several advantages. First, since standard LPC systems successfully use either white noise or an impulse train as excitation functions, it is evident that errors in the details of the residual waveform or synthesizer driving process may not result in much perceptual deterioration of the speech at the synthesizer output. Second, coding of the *actual* residuals should tend to reduce the effects of failures of the voiced/unvoiced model and the effects of background noise.

For these systems, a vector quantized LPC system was used, operating with a codebook of 512 tenth-order filters for a rate of 9 bits per LPC frame or about 450 bits/s. The LPC system supplied combined gain and model, so that the residual signal, generated by passing the original speech through the filter $A(z)/\sigma$, was normalized by the LPC gain. The LPC filter set was designed by Rebolledo using the block code design algorithm and the Itakura-Saito distortion measure [16], [33].

Using 450 bits/s of side information for the gain and model parameters combined with an original sampling rate of 6500 Hz, trellis RELP speech coding systems were produced operatIEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-30, NO. 4, APRIL 1982

ing at 3700 bits/s, 6950 bits/s, and 13 450 bits/s for the 1/2, 1, and 2 bits/symbol cases. As in the case of the waveform coding experiments, the trellis decoders for the speech residuals were designed using the extension method, starting from constraint-length 1 decoders. The distortion measure used was the simple squared-error measure.

The most evident artifact in the RELP speech was a growllike sound at the LPC frame rate. In our judgment, the RELP system offered improved quality over the waveform coder of Fig. 5 at equivalent rate.

Hybrid Tree Codes

In the trellis-coded RELP systems just discussed, the rateconstrained residual signal is reproduced according to the trellis code distortion measure (squared-error) without regard for the eventual waveform after the receiver passes the decoded residuals through LPC synthesis filter. It was with some surprise, therefore, that we observed that the output waveform of the RELP system was close to that of the original speech signal. As a result of this observation, we decided to test a composite decoder consisting of the $\sigma/A(z)$ LPC synthesis filter combined with a trellis decoder front end. This "hybrid tree code" system is shown in Fig. 7. The only difference between the RELP system and this new system is that the hybrid system encoder seeks a channel sequence which will, at the decoder output, match the original speech waveform while the RELP system encoder seeks a channel sequence which will, at the output of the trellis decoder, match the speech residuals.

As before, the LPC portion of the decoder consists of one of 512 tenth-order filters selected by a vector quantizer. The filter selection is made independently on each frame of speech. The trellis portion of the hybrid decoder is in fact the same decoder as was used in the RELP systems—it was designed using the trellis code design algorithm on the speech residuals from the training sequence.

Fig. 8 gives performance data for hybrid tree encoding systems operating at 1/2, 1, and 2 bits/sample for the driving process, plus 450 bits/s for the LPC portion of the system. As before, the figure gives signal-to-noise ratios in decibels for the entire 10 s speech segment. The combined channel rates for these codes are 3700, 6950, or 13 450 bits/s. Although these rates are the same as for the RELP system, the hybrid tree system produces considerably better perceptual quality.

In the usual predictive coding systems (including adaptive systems), the prediction filter is driven by a variable step-size quantizer, which is not a particularly good model of speech residuals. In the hybrid tree coder, the trellis front end has been specifically designed for speech residuals. It makes intuitive sense that the closer the driving process of an APClike system is to residuals, the better the overall system will perform. Although the vector quantized LPC filters used in this paper are different from those used in most APC systems, the predictor from an APC system could be easily used. Descriptions of trellis implementations of otherwise conventional APC systems may be found in [10].

In [30], a collection of hybrid decoders consisting of LPC filters with a fixed trellis front end are used for universal cod-



Fig. 8. Hybrid tree code performance. Performance figures are in decibel SNR for the original speech waveform. Curve a is for the 1/2 bit per sample code outside the training sequence. Curve b is for the rate 1 code, and curve c is for the rate 2 code.

ing of a speech waveform. This system differs from the hybrid system described in this section in two ways: only 16 LPC filters are used, rather than the much larger set of 512 filters considered here, and a fake process trellis decoder matched to a Laplacian distribution is used, rather than a trellis decoder designed explicitly for speech residuals. Matsuyama and Gray do suggest in [30] the use of a spectral distortion measure for filter selection in place of universal coding and this approach is taken in [29]. The same encoding method is used in [1], although the decoders considered do not include a trellis component and the encoder is a single path search such as in standard APC. In an informal subjective comparison, the hybrid tree encoding system described in this section has, for a given channel rate, given speech quality significantly improved over that of the similar system of [29].

SUMMARY

We have presented algorithms for the design of trellis encoding data compression systems based on a training sequence of data. The algorithms have been tested on random sources and applied to the problem of speech compression. The relatively poor performance of the fixed-trellis (nonadaptive) speech waveform coder, together with the observed waveform tracking ability of the RELP systems, suggest that it is advantageous to design a composite, or hybrid, decoder which makes use of knowledge about speech. In the hybrid tree system, a spectral distortion measure is used on a frameby-frame basis to select that filter from a finite collection of LPC filters which best matches the short-term speech spectrum. The chosen filter is used together with a fixed trellis decoder designed to emulate speech residuals. The combined decoder is used as the code generator for a classic squarederror tree encoding system.

The design methods presented in this paper have two main applications. First, any existing coding system whose decoder can be described at least in part by a trellis structure can potentially be improved by the algorithms. Second, the algorithms can construct new trellis decoders from scratch for sources with unknown characteristics. Most other code construction methods depend upon special knowledge of the source.

REFERENCES

- J.-P. Adoul, J. L. Debray and D. Dalle, "Spectral distance measure applied to the design of DPCM coders with L predictors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, Apr. 1980, pp. 512-515.
- [2] J. B. Anderson, "Effectiveness of sequential tree search algorithms in speech digitization," in *Proc. Int. Conf. Commun.*, Boston, MA, June 1979, pp. 8.1.1–8.1.6.
- [3] —, "Recent advances in sequential encoding of analog waveforms," in *Proc. Nat. Telecommun. Conf.*, Birmingham, AL, Dec. 1978, pp. 19.4.1–19.4.5.
- [4] J. B. Anderson and J. B. Bodie, "Tree encoding of speech," IEEE Trans. Inform. Theory, vol. IT-21, pp. 379–387, July 1975.
- [5] J. B. Anderson and C. W. Law, "Real-number convolutional codes for speech-like quasi-stationary sources," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 778–782, Nov. 1977.
- [6] B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech signals," *Bell Syst. Tech. J.*, vol. 49, pp. 1973–1986, Oct. 1970.
- [7] D. W. Becker and A. J. Viterbi, "Speech digitization and compression by adaptive predictive coding with delayed decision," in *Proc. IEEE Nat. Telecommun. Conf.*, New Orleans, LA, 1975, pp. 46/18-46/23.
- [8] T. Berger, Rate Distortion Theory, A Mathematical Basis for Data Compression. Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [9] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based on vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 562–574, Oct. 1980.
 [10] H. G. Fehn and P. Noll, "Tree and trellis coding of speech and
- [10] H. G. Fehn and P. Noll, "Tree and trellis coding of speech and stationary speech-like signals," in *Proc. IEEE Int. Conf. Acoust.*, *Speech, Signal Processing*, Denver, CO, Apr. 1980, pp. 547–550.
- [11] G. D. Forney, Jr., "The Viterbi algorithm," Proc. IEEE, vol. 61, pp. 268–278, Mar. 1973.
- [12] J. D. Gibson, "Tree coding versus differential encoding of speech," presented at the Int. Conf. Inform. Theory, Santa Monica, CA, Feb. 1981.
- [13] A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Comparison of optimal quantizations of speech reflection coefficients," IEEE

Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 9-23, Feb. 1977.

- [14] A. H. Gray and D. Y. Wong, "The Burg algorithm for LPC speech analysis/synthesis," *IEEE Trans. Acoust., Speech, Signal Proc*essing, vol. ASSP-28, pp. 609-615, Dec. 1980.
- [15] R. M. Gray, "Time-invariant trellis encoding of ergodic discretetime sources with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 71-83, Jan. 1977.
- [16] R. M. Gray, A. Buzo, A. H. Gray, Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 367–376, Aug. 1980.
- [17] R. M. Gray, J. C. Kieffer, and Y. Linde, "Locally optimal block quantizer design," *Inform. Contr.*, vol. 45, pp. 178–198, May 1980.
- [18] R. M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Commun.*, vol. COM-30, pp. 381-389, Feb. 1982.
- [19] N. S. Jayant and S.A. Christensen, "Tree-encoding of speech using the (M, L)-algorithm and adaptive quantization," *IEEE Trans.* Commun., vol. COM-26, pp. 1376–1379, Sept. 1978.
- [20] F. Jelinek, "Tree encoding of memoryless discrete time sources with a fidelity criterion." *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 584-590, Sept. 1969.
- [21] F. Jelinek and J. B. Anderson, "Instrumentable tree encoding of information sources," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 118-119, Jan. 1971.
- [22] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84– 95, Jan. 1980.
- [23] Y. Linde and R. M. Gray, "The design of tree and trellis data compression systems," Inform. Syst. Lab. rep., Stanford Univ., Stanford, CA, SEL Tech. Rep. 6504-2, Feb. 1978.
- [24] —, "A fake process approach to data compression," IEEE Trans. Commun. vol. COM-26, pp. 840-847, June 1978.
- [25] S. P. Lloyd, "Least squares quantization in PCM's," Bell Labs. Paper, Murray Hill, NJ, 1957; also, *IEEE Trans. Inform Theory*, vol. IT-28, Mar. 1982.
- [26] J. W. Mark, "Adaptive trellis encoding of discrete-time sources with a distortion measure," *IEEE Trans. Commun.*, vol. COM-25, pp. 408-417, Apr. 1977.
- [27] J. D. Markel and A. H. Gray, Linear Prediction of Speech. New York: Springer-Verlag, 1976.
 [28] Y. Matsuyama, "Process distortion measures and signal proc-
- [28] Y. Matsuyama, "Process distortion measures and signal processing," presented at the Int. Conf. Inform. Theory, Santa Monica, CA, Feb. 1981.
- [29] —, "Speech compression systems using a set of inverse filters," Ph.D. dissertation, Inform. Syst. Lab., Stanford Univ., Stanford, CA, Aug. 1978.
- [30] Y. Matsuyama and R. M. Gray, "Universal tree coding for speech," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 31–40, Jan. 1981.
- [31] J. Max, "Quantizing for minimum distortion," IRE Trans. Inform. Theory, vol. IT-6, pp. 7–12, Mar. 1960.
- [32] W. A. Pearlman, "A sliding block code for small user alphabets with performance near the rate-distortion limit," presented at the Int. Conf. Inform. Theory, Santa Monica, CA, Feb. 1981.
- [33] G. Rebolledo, "Speech and waveform coding based on vector quantization," Ph.D. dissertation, Inform. Syst. Lab., Stanford Univ., Stanford, CA, 1982.
- [34] L. C. Stewart, "Trellis data compression," Ph.D. dissertation, Inform. Syst. Lab., Stanford Univ., Stanford, CA, June 1981; available as Tech. Rep. L905-1.
- [35] J. Uddenfeldt and L. H. Zetterberg, "Algorithms for delayed encoding in delta modulation with speech-like signals," *IEEE Trans. Commun.*, vol. COM-24, pp. 652–658, June 1976.
- [36] C. K. Un and D. T. Magill, "The residual-excited linear prediction vocoder with transmission rate below 9.6 kbits/s," *IEEE Trans. Commun.*, vol. COM-23, pp. 1466–1474, Dec. 1975.
 [37] S. G. Wilson and S. Husain, "Adaptive tree encoding of speech at
- [37] S. G. Wilson and S. Husain, "Adaptive tree encoding of speech at 8000 bits/s with a frequency weighted error criterion," *IEEE Trans. Commun.*, vol. COM-27, pp. 165–170, Jan. 1979.
- [38] D. Y. Wong, F. B. Juang, and A. H. Gray, "An 800 bps speech compression system based on vector quantization," presented at the Int. Conf. Inform. Theory, Santa Monica, CA, Feb. 1981; also, *IEEE Trans. Acoust., Speech, Signal Processing*, to be published.



Lawrence C. Stewart (S'74–M'80) received the S.B. degree from the Massachusetts Institute of Technology, Cambridge, in 1972 and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1977 and 1981, respectively, all in electrical engineering.

Since 1977, he has been with the Xerox Palo Alto Research Center, Palo Alto, CA, joining the Computer Science Laboratory there in 1981. His current interests are in data compression, packet radio, computer communications, and computer systems.

Dr. Stewart is a member of Eta Kappa Nu and Tau Beta Pi.

★



Robert M. Gray (S'68–M'69–SM'77–F'80) was born in San Diego, CA, November 1, 1943. He received the B.S. and M.S. degrees from the Massachusetts Institute of Technology, Cambridge, in 1966 and the Ph.D. degree from the University of Southern California, Los Angeles, in 1969, all in electrical engineering.

Since 1969 he has been with the Information Systems Laboratory and the Department of Electrical Engineering, Stanford University, Stanford, CA, where he is currently a Professor

engaged in teaching and research in communication and information theory and in speech compression.

Dr. Gray was a member of the IEEE Information Theory Group Board of Governors, and was corecipient of the Group's 1976 Prize Paper Award. He was Associate Editor for Shannon Theory and is currently Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY. He is the recipient of a 1981 Fellowship from the Japan Society for the Promotion of Science and a 1982 Fellowship from the John Simon Guggenheim Memorial Foundation. He is a member of Sigma Xi, Eta Kappa Nu, the Society for Industrial and Applied Mathematics, the Institute of Mathematical Statistics, the Association for Computing Machinery, and the Societé des Ingénieurs et Scientifiques de France.



Yoseph Linde (S'76–M'78) was born in Poland on June 14, 1947. He received the B.Sc. degree from the Technion—Israel Institute of Technology, Haifa, in 1970, the M.S. degree from Tel-Aviv University, Tel-Aviv, Israel, in 1975 and the Ph.D. degree from Stanford University, Stanford, CA, in 1977, all in electrical engineering.

From 1970 to 1975 he was with the Signal Corps, where he was involved in research and development of communication systems. In

1976–1977 he was a Research Assistant at the Information System Laboratory at Stanford University, involved in research in data compression systems. He is currently with the Codex Corporation, Mansfield, MA, where he is involved in developing architectures for VLSI implementation of digital signal processing.