Smoothed DPCM Codes

Wen-Whei Chang, *Member, IEEE,* and Jerry D. Gibson, *Senior Member, IEEE*

Abstract—**Rate distortion theory promises that autoregressive sources can be encoded optimally at small distortions (high rates) by a source coder with infinite encoding delay and zero delay at the decoder. However, for instrumentable systems with finite encoding delay and an unmatched code generator or for operation at low rates, decoding delay may provide a performance increment. The alphabet constrained approach to data compression allows delay at both the encoder and the decoder, and Sethia and Anderson incorporate delay in a tree coder code generator by combining a weighted linear interpolation scheme with DPCM. This system, called interpolative DPCM (IDPCM), was shown to outperform DPCM at rate 1 b/sample for several synthetic source models. In the present work, we use minimum mean squared error (MMSE) fixed-lag smoothing in conjunction with DPCM to develop a code generator employing delayed decoding. This smoothed DPCM (SDPCM) code generator is compared to DPCM and IDPCM code generators at rates 1 and 2 b/sample for tree coding several synthetic sources and to a DPCM code generator at rate 2 b/sample for speech sources. The (M,L) algorithm, which retains only the M best paths to depth L, is used for tree searching, and SDPCM outperforms IDPCM and DPCM at rate 2 b/sample for the synthetic sources with $M = 1, 4, 8,$ and 12, and at rate 1 b/sample with $M \geq 4$. For speech, SDPCM provides a slight improvement in MSE over DPCM codes that is also evident in sound spectrograms and informal subjective listening tests. The models upon which the fixed-lag smoother is based must be chosen appropriately to achieve good SDPCM performance.**

I. Introduction

THE use of encoding delay to improve the performance of differential pulse code modulation (DPCM) systems was motivated by both rate distortion theory [1], [4] and intuition [2], [3]. As noted by Sethia and Anderson [5], optimal encoding of autoregressive (AR) sources with small distortion can be achieved by a system with infinite delay at the encoder and no delay at the decoder [1]. However, for finite encoding delays, suboptimal code generators (such as DPCM), and larger distortions, delay at the decoder may prove useful. Gibson [6] proposed the incorporation of delay at the decoder, leaving the encoder code generator unchanged, and investigated the performance improvement available using a fixed-lag smoother for first-order AR sources. The alphabet constrained approach to data compression formulated by Gibson and Fischer [7] allows delay at both the encoder and decoder, and Sethia and Anderson [5] propose and evaluate one such system based upon DPCM. In particular, Sethia and Anderson [5] modify the code generator to take advantage of delay by using a weighted linear interpolation of nearby outputs, and evaluate the performance of these interpolated DPCM codes at rate 1 b/sample for various correlated source models.

In the present paper we study delayed decoding by using a fixed-lag smoother in a code generator based upon DPCM, and investigate the performance of these codes for tree coding at rates 1 and 2 b/sample for synthetic sources and at rate 2 b/sample for speech. The (M, L) tree search algorithm [4], the squared error distortion measure, and the incremental path map symbol release rule are employed exclusively. For synthetic sources, the performance indicator is segmental signal-to-noise ratio (SNRSEG), while for speech, we use SNRSEG, sound spectrograms, and informal subjective listening tests.

II. DPCM-Based Tree Codes

Block diagrams of a DPCM system transmitter and receiver are shown in Fig. 1. Classical DPCM systems operate without delay in the sense that for an input sample at time instant n, only data at times $j \leq n$ are used in the encoding process. Tree coders attempt to improve on DPCM by delaying the encoding decision for a few samples, say L, which allows the input samples at time instants $j \leq n + L$ to be used to encode the input sample at time n. Slight delays are often not critical to the operation of communication systems, and this delay allows all possible encoding sequences through time $n + L$ to be examined for a best fit. Each different encoding sequence is called a path, and hence, tree coding is a multipath search procedure, whereas DPCM exhibits a single path search.

That a DPCM system implements a single path search is evident from its mode of operation. From the source sample at time instant n, denoted $x(n)$, a DPCM system subtracts a predicted value $\hat{x}(n \mid n-1)$ to generate the prediction error $e(n)$. The prediction error is then quantized (by Q) to obtain $e_q(n)$, which is converted to a binary sequence for transmission to the receiver. At the receiver, the binary sequence is decoded to produce $e_q(n)$, which is then used to generate the reconstructed output sample $\hat{x}(n)$ by adding in the predicted value $\hat{x}(n \mid n-1)$. The same operations are performed at the transmitter. At both the transmitter and receiver, the $\{\hat{x}(n)\}$ sequence is used to obtain the next predicted value. A new source sample appears, and the process is duplicated.

Paper approved by the Editor for Quantization, Speech/Image Coding of the IEEE Communications Society. Manuscript received March 7, 1989; revised May 11, 1990. This work was supported in part by the National Science Foundation under Grant NCR-8914496. This paper was presented in part at the 1990 IEEE International Symposium on Information Theory, San Diego, CA, January 14–19, 1990 and at the 1989 IEEE Global Telecommunications Conference, Dallas, TX, November 27–30, 1989.

W.-W. Chang is with the Department of Communication Engineering, National Chiao-Tung University, Taiwan, Republic of China.

J. D. Gibson is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843.

IEEE Log Number 9100317.

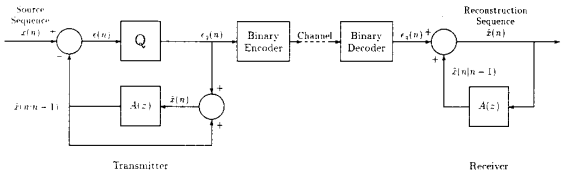


Fig. 1. DPCM system block diagram.

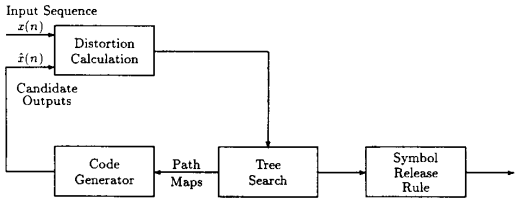


Fig. 2. Functional diagram of a tree coder.

A functional block diagram of a tree coder (transmitter only) is shown in Fig. 2. The distortion between the source sequence and each possible reconstructed sequence to some depth L in the tree is calculated, and the path through the tree with the smallest distortion (to depth L) is selected as the best path. Path map digits corresponding to this path (or some portion thereof) are then released as encoder output digits and sent to the decoder or receiver for reconstruction. The path map digits defining the minimum distortion path are also provided to the code generator at the encoder. The optimum or minimum distortion path is then extended to depth L, and the process is repeated. The source sequence is reconstructed (to some fidelity) at the receiver by applying the encoder output digits to the code generator input. Design of a tree coder consists of selecting a code generator, a distortion measure, a tree search algorithm, and a path map symbol release rule. Here we describe a tree coder based upon DPCM.

Although DPCM is a single path search procedure, a DPCM system can be used as the basis for a tree coder design. The part of a DPCM system transmitter which emulates the DPCM receiver can function as a code generator, and by delaying the transmission of $e_q(n)$, and basing the decision as to which $e_q(n)$ value to send on the distortion between $x(j)$ and $\hat{x}(j)$, $j \leq n + L$, we obtain a tree coder. A DPCM based tree coder transmitter is illustrated in Fig. 3. The receiver is unmodified.

The basic equations describing the code generator operation are

$$\hat{x}(n) = \hat{x}(n \mid n-1) + e_q(n) \tag{1}$$

and

$$\hat{x}(n \mid n-1) = \sum_{i=1}^{p} a_i \hat{x}(n-i). \tag{2}$$

The blocks labeled Q in Fig. 1 and "adaptive quantizer-based decoder" in Fig. 3 represent a quantizer with step size $\Delta(n)$, which together with (1) and (2) allow the reconstructed signal to be calculated from a given path map sequence.

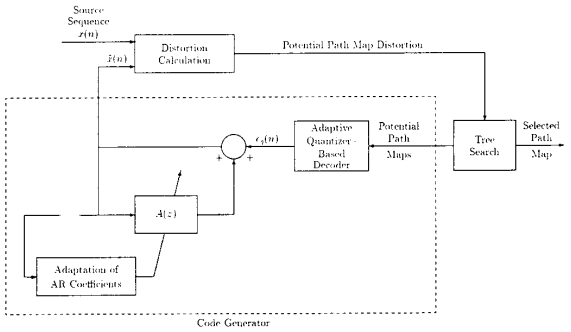


Fig. 3. Tree coder with a DPCM code generator.

Other components of a tree coder are the distortion measure, the tree search algorithm, and the path map symbol release rule. Here we use the single letter, sum of the squared errors distortion measure given by

$$d(x, \hat{x}) = \frac{1}{L} \sum_{i=1}^{L} [x(i) - \hat{x}(i)]^2 \tag{3}$$

where the $x(i)$ and $\hat{x}(i)$ in (3) refer to the values currently at depth i in the tree. The tree search algorithm employed is the (M, L) algorithm [4], [5], and the path map symbols are released according to the incremental, single symbol release rule. For given initial conditions, the (M, L) algorithm finds the M best paths to depth L and discards all of the other paths to this depth. For incremental, single symbol release, only the first symbol in *the* best path is sent, and the M or fewer paths with this root node are again extended to depth L, and the process is repeated. This algorithm has a complexity proportional to $M\|C\|$, where $\|C\|$ is the number of symbols/sample (or branches/node), rather than $\|C\|^L$, as is the case for exhaustive searching of the tree. Specific values of the several tree coder parameters are described in Section V for each source of interest.

III. Interpolative DPCM

Sethia and Anderson [5] proposed and studied interpolative DPCM (IDPCM) at rate 1 b/sample for several AR source models, including the McD(1), LAW(2), McD(3), and LAW(3) sources. The designations McD and LAW were used in [5] and are short for McDonald [14] and Law [15], respectively, the authors who first used these sources. We also investigate the BWH(3) source, which was constructed from the correlations of a two-pole Butterworth filter by Anderson and Bodie [4]. Given the standard DPCM system output sequence $\{\hat{x}(n)\}$, the IDPCM system calculates the code generator output as a weighted linear combination of local DPCM outputs or available previously interpolated values. In particular, for the McD(1), LAW(2), and McD(3) source models, the IDPCM code generator outputs $\{\hat{x}_b(n \mid n+1)\}$ are calculated from

$$\hat{x}(n+1) = a_1 \hat{x}(n) + a_2 \hat{x}(n-1) + a_3 \hat{x}(n-2) + e_q(n+1) \tag{4}$$

TABLE I
PREDICTOR AND INTERPOLATOR COEFFICIENTS: McD(1), LAW(2), McD(3)

	(a_1, a_2, a_3)	$(b_1, b_{-1}, b_{-2}, b_{-3})$
McD(1), $R = 1$	(0.900, 0.000, 0.000)	(0.497, 0.497, 0.000, 0.000)
LAW(2), $R = 1$	(1.725, −0.781, 0.000)	(0.429, 0.801, −0.221, 0.000)
McD(3), $R = 1$	(1.748, −1.222, 0.301)	(0.443, 0.963, −0.454, 0.093)

and

$$\hat{x}_b(n \mid n+1) = b_1\hat{x}(n+1) + b_{-1}\hat{x}_b(n-1 \mid n) + b_{-2}\hat{x}_b(n-2 \mid n-1) + b_{-3}\hat{x}_b(n-3 \mid n-2) \quad (5)$$

where the predictor $\{a_i\}$ and interpolator $\{b_i\}$ coefficients are listed in Table I [5]. For the LAW(3) and BWH(3) source models, the IDPCM code generator equations are

$$\hat{x}(n+1) = a_1\hat{x}(n) + a_2\hat{x}(n-1) + a_3\hat{x}(n-2) + e_q(n+1) \quad (6)$$

and

$$\hat{x}_b(n \mid n+1) = b_1\hat{x}(n+1) + b_0\hat{x}(n) + b_{-1}\hat{x}(n-1) + b_{-2}\hat{x}(n-2) \quad (7)$$

where the predictor and interpolator coefficients are as specified in Table II. This IDPCM code generator is a variation on those presented by Sethia and Anderson [5], [8]. We leave further development of these source models and the description of the computation of the interpolator coefficients to the references [5], [8]. Note, however, from (5) and (7) that these are single lag interpolators based upon past interpolated values and a "future" DPCM output or simply based upon DPCM outputs, respectively.

IV. SMOOTHED DPCM

In our smoothed DPCM (SDPCM) system, we compute code generator outputs using minimum mean squared error (MMSE) fixed-lag smoothing algorithms based upon specified message and observation (or measurement) models. The fixed-lag smoothing algorithms are relatively straightforward but they are described briefly in Appendix A for completeness and notational purposes.

To apply the fixed-lag smoothing algorithms in Appendix A to SDPCM, it is necessary to develop both the message and the measurement models which describe the dynamics of the source and the DPCM system. Consider a pth-order autoregressive source. Its current source sample $x(n)$ can be expressed as a weighted sum of past source samples plus an excitation term $\varepsilon(n)$, where $\varepsilon(n)$ is a zero-mean, white Gaussian process with variance σ_ε^2. Letting $\{a_i\}$ denote the autoregression coefficients, which are the same as the short-term predictor coefficients used in the DPCM system, we have, for a third order $AR(3)$ source, the process model

$$\boldsymbol{x}(n+1) = \boldsymbol{\Phi}(n)\boldsymbol{x}(n) + \boldsymbol{v}(n) \quad (8)$$

TABLE II
PREDICTOR AND INTERPOLATOR COEFFICIENTS: LAW(3), BWH(3)

	(a_1, a_2, a_3)	$(b_1, b_0, b_{-1}, b_{-2})$
LAW(3), $R = 1$	(1.526, −0.773, 0.101)	(0.3489, 0.5081, 0.1145, 0.0552)
BWH(3), $R = 1$	(1.231, −0.625, 0.119)	(0.3006, 0.6512, 0.0789, 0.0190)
LAW(3), $R = 2$	(1.526, −0.773, 0.101)	(0.1475, 0.7871, 0.0806, 0.0061)
BWH(3), $R = 2$	(1.231, −0.625, 0.119)	(0.1229, 0.8550, 0.0553, −0.001)

where

$$\boldsymbol{x}(n+1) = \begin{bmatrix} x(n+1) \\ x(n) \\ x(n-1) \end{bmatrix} \qquad \boldsymbol{v}(n) = \begin{bmatrix} \varepsilon(n) \\ 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \end{bmatrix} \qquad \boldsymbol{\Phi}(n) = \begin{bmatrix} a_1 & a_2 & a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (9)$$

and the excitation vector $\boldsymbol{v}(n)$ has the covariance matrix

$$\boldsymbol{Q}(n) = E\left[\boldsymbol{v}(n)\boldsymbol{v}^T(n)\right] = \begin{bmatrix} \sigma_\varepsilon^2 & 0 & 0 \\ 0 & \sigma_\varepsilon^2 & 0 \\ 0 & 0 & \sigma_\varepsilon^2 \end{bmatrix}. \quad (10)$$

For a DPCM system, there always exists a distortion between the source sample $x(n)$ and the reconstruction sample $\hat{x}(n)$ by the amount of quantization noise $q(n)$, $\hat{x}(n) = x(n) + q(n)$. Assuming that $q(n)$ is zero-mean, white noise with variance σ_q^2, the corresponding measurement model is

$$y(n) = \boldsymbol{c}(n)\boldsymbol{x}(n) + q(n) \quad (11)$$

where

$$\boldsymbol{c}(n) = [1 \quad 0 \quad 0]. \quad (12)$$

Given the process and measurement models in (8)–(12), the Kalman filtering algorithm described in (A.5)–(A.8) can be applied to obtain the filtered estimate vector $\hat{\boldsymbol{x}}_f(n \mid n)$ given $\hat{x}(n)$. The resulting filtered estimate vector can then be used together with the preceding filtered estimate vector $\hat{\boldsymbol{x}}_f(n-1 \mid n-1)$ to calculate the smoothed estimate sample $\hat{x}_s(n-1)$ from (A.9)–(A.11).

The block diagram of a smoothed DPCM code generator is shown in Fig. 4. Fig. 5 shows the code tree generated by a smoothed DPCM system. It indicates that the SDPCM code tree in dashed lines follows the DPCM code tree in solid lines. The code tree of SDPCM can be viewed as constructed in two steps. First, the secondary code tree (solid line) is generated by conventional DPCM; then superimposed on it, the code tree (dashed line) populated by the output of the fixed-lag smoother. Note that each branch of the code tree in solid lines is associated with both the reconstruction sample $\hat{x}(n)$ and the filtered estimate vector $\hat{\boldsymbol{x}}_f(n \mid n)$, while each branch of the code tree in dashed lines is populated with the smoothed estimate sample $\hat{x}_s(n)$. It is evident that the SDPCM system with binary $e_q(n)$ results in twice as many branches at each level of the tree as those generated by conventional DPCM, which is also true of IDPCM.

TABLE III
PERFORMANCE FOR AR SOURCES IN SNRSEG (dB): DPCM

	$M = 1$	$M = 4$	$M = 8$	$M = 12$	DRF
McD(1), $R = 1$	8.40 ±0.14	8.75 ±0.14	8.75 ±0.14	8.75 ±0.14	13.23
LAW(2), $R = 1$	14.53 ±0.13	16.98 ±0.14	17.06 ±0.14	17.07 ±0.13	
McD(3), $R = 1$	8.33 ±0.11	10.82 ±0.09	10.86 ±0.09	10.88 ±0.09	14.3[a]
LAW(3), $R = 1$	9.07 ±0.11	10.68 ±0.10	10.74 ±0.10	10.76 ±0.10	
BWH(3), $R = 1$	6.26 ±0.07	7.29 ±0.08	7.31 ±0.08	7.31 ±0.08	11.19[b]
LAW(3), $R = 2$	15.08 ±0.09	15.82 ±0.10	15.85 ±0.10	15.85 ±0.10	21.45
BWH(3), $R = 2$	11.31 ±0.08	11.98 ±0.08	12.01 ±0.08	12.01 ±0.08	17.52

[a]From [5].
[b]From [8].

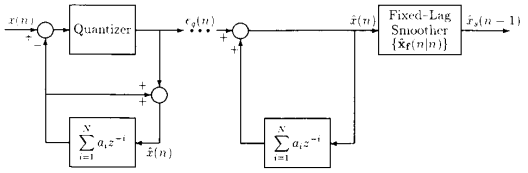


Fig. 4. SDPCM encoder/decoder.

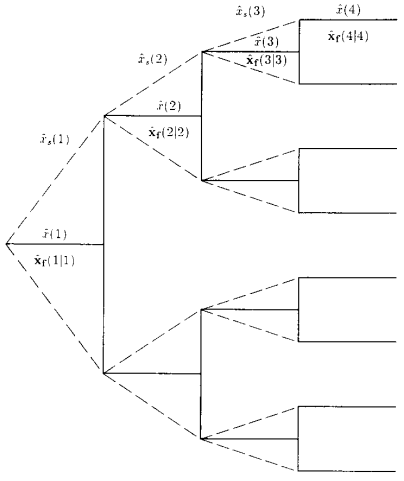


Fig. 5. SDPCM code tree.

V. PERFORMANCE COMPARISONS FOR SYNTHETIC SOURCES

We present comparative performance results for tree encoding of pth-order Gaussian autoregressive sources $(AR(p))$ using DPCM, interpolative DPCM (IDPCM), and smoothed DPCM (SDPCM) as code generators at rates 1 and 2 b/sample. The encoding performance is evaluated over a range of synthetic $AR(p)$ sources in terms of the segmental SNR (SNRSEG) (see Appendix B) with 1000 samples per segment. The tree coder uses the squared error distortion measure, the (M, L) search algorithm, and the single symbol release rule. The rate 1 b/symbol quantizer used here has an output step size of 1.8, which is in generally good agreement with the best choice in [5], although the optimum value can vary with code parameters. At rate 2 b/symbol, we use a nonuniform quantizer designed to minimize the MSE for a memoryless Gaussianly-distributed input with variance $(1.8)^2$. The resulting scale factor of 1.8 is a nominal value that provides good performance based on our experiments over a range of sources and codes. An $AR(p)$ source $x(n)$ is generated by passing a zero-mean, unit-variance, white Gaussian excitation sequence $\varepsilon(n)$ through a coloring filter with transfer function $1/(1 - \sum_{i=1}^{p} a_i z^{-i})$ such that its successive samples satisfy the difference equation $x(n) = \sum_{i=1}^{p} a_i x(n-i) + \varepsilon(n)$. We use 1000 samples per experiment, and each experiment is repeated 20 times, each time using an independent realization of $\varepsilon(n)$.

Tables III–V show the performance in terms of segmental SNR (along with 68% confidence intervals) for tree coding of $AR(p)$ sources using DPCM, IDPCM, and SDPCM code generators, respectively, all with search depth $L = 10$ and $M = 1$, 4, 8, and 12. The 68% confidence intervals indicate that the ensemble average segmental SNR values shown in these tables fall within $\pm\hat{\sigma}/\sqrt{N}$, where $\hat{\sigma}$ = sample standard deviation and N = number of experiments, of the true SNR value with probability 0.68. The 68% confidence intervals are used for consistency with [5]. The SNR values corresponding to the distortion rate function (DRF) shown in the tables were taken from [5] and [8], respectively, for the McD(3) source with $R = 1$ and the BWH(3) source with $R = 1$. The other three values presented were calculated using the approach of Bunin [16], which only holds for general AR sources when $R \geq R_{\min}$. The rate $R_{\min}$ is the minimum rate for which the rate distortion function of the AR source $\{x(n)\}$ equals the rate distortion function of the source driving term $\{\varepsilon(n)\}$ [1], [16]. At rate $R = 2$ b/sample, both IDPCM and SDPCM substantially outperform DPCM for both sources and all M. The lower rate $R = 1$ results still indicate a performance advantage for IDPCM and SDPCM over DPCM with $M \geq 4$, however, the IDPCM and SDPCM performance is very close. If the number of retained paths M is only 1 at rate $R = 1$, IDPCM and SDPCM performance remains close for the McD(1) source and both outperform DPCM. However, for the LAW(2) and McD(3) sources, the ranking of performance from best to worst is IDPCM, DPCM, and SDPCM.

The reason for this degraded performance of SDPCM is that a two-level $(R = 1)$ quantizer together with single path searching produces reconstruction error samples, which are measurement noise samples, that are too highly correlated

TABLE IV
PERFORMANCE FOR AR SOURCES IN SNRSEG (dB): IDPCM

	$M=1$	$M=4$	$M=8$	$M=12$	DRF
McD(1), $R=1$	11.09 ±0.17	11.53 ±0.14	11.63 ±0.14	11.70 ±0.14	13.23
LAW(2), $R=1$	15.08 ±0.20	18.80 ±0.18	19.20 ±0.14	19.46 ±0.13	
McD(3), $R=1$	8.96 ±0.14	12.17 ±0.07	12.48 ±0.09	12.68 ±0.09	14.3[a]
LAW(3), $R=1$	10.74 ±0.12	12.48 ±0.10	12.77 ±0.10	12.92 ±0.11	
BWH(3), $R=1$	7.95 ±0.07	8.96 ±0.08	9.08 ±0.07	9.14 ±0.07	11.19[b]
LAW(3), $R=2$	16.59 ±0.09	17.57 ±0.10	17.70 ±0.10	17.81 ±0.11	21.45
BWH(3), $R=2$	12.32 ±0.08	13.21 ±0.08	13.26 ±0.08	13.34 ±0.08	17.52

[a]From [5].
[b]From [8].

TABLE V
PERFORMANCE FOR AR SOURCES IN SNRSEG (dB): SDPCM

	$M=1$	$M=4$	$M=8$	$M=12$	DRF
McD(1), $R=1$	11.27 ±0.15	11.63 ±0.15	11.66 ±0.14	11.74 ±0.14	13.23
LAW(2), $R=1$	13.96 ±0.21	18.88 ±0.16	19.20 ±0.15	19.46 ±0.13	
McD(3), $R=1$	7.63 ±0.13	12.26 ±0.10	12.58 ±0.09	12.75 ±0.10	14.3[a]
LAW(3), $R=1$	9.70 ±0.16	12.60 ±0.11	12.97 ±0.11	13.12 ±0.10	
BWH(3), $R=1$	7.65 ±0.10	9.64 ±0.10	9.73 ±0.10	9.75 ±0.10	11.19[b]
LAW(3), $R=2$	17.12 ±0.12	19.21 ±0.11	19.45 ±0.12	19.58 ±0.13	21.45
BWH(3), $R=2$	13.33 ±0.12	15.29 ±0.10	15.46 ±0.09	15.56 ±0.08	17.52

[a]From [5].
[b]From [8].

and therefore do not match the white measurement noise assumption used in developing the fixed-lag smoothing algorithms. Increasing the coding rate R or multipath searching tends to reduce the statistical dependence in the reconstruction error (measurement noise) so that the assumed model is more accurate. Since the IDPCM interpolator is not based on an explicitly stated model, its performance is not as severely affected at $R=1$ with $M=1$. IDPCM is not able to perform as well as SDPCM at $R=2$ for the same reason, however.

To obtain a clearer view of relative performance, the SNRSEG is plotted for the three code generators at $R=1$ for the LAW(3) and BWH(3) sources in Figs. 6 and 7, respectively, and for the same two sources at rate $R=2$ in Figs. 8 and 9. From Figs. 6 and 7, we see that at rate $R=1$, multipath searching improves the performance of SDPCM over DPCM and IDPCM as M increases, and the rate 2 results in Figs. 8 and 9 show a 1.77 to 2.22 dB performance gain for SDPCM over IDPCM with $M=12$ and a 3.55 to 3.73 dB gain for SDPCM relative to DPCM.

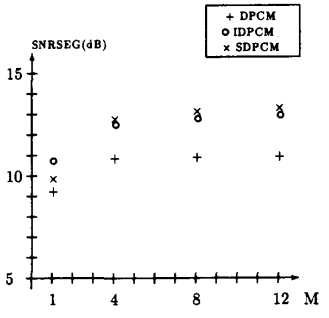


Fig. 6. Improvement versus retained paths—LAW(3), $R=1$.

VI. SDPCM PERFORMANCE FOR SPEECH

Using the well-known linear prediction model for speech, fixed-lag smoothing algorithms can be written for speech analogous to those described in Section V. This section presents simulation results for tree coding of speech at $R=2$ b/sample (16 kb/s) using DPCM and SDPCM code generators. The tree coders use a four-level robust Jayant quantizer [12], the squared error distortion measure, the (M,L) tree search algorithm, and the single symbol release rule. The SNR, SNRSEG, sound spectrograms, and informal subjective listening tests are used to evaluate coder performance for the five sentences in Appendix C. The SNRSEG is calculated on nonoverlapping 20 ms segments. Only 8th-order predictors are considered, and the coefficients are adapted using the least-squares lattice algorithm described in Reininger and Gibson [13].

Tables VI and VII show SNR/SNRSEG in decibels for DPCM and SDPCM codes, respectively, for five sentences using the (M,L) search algorithm with depth $L=10$ and $M=1$, 4, 8, and 12. The results show that SDPCM has a higher SNRSEG than DPCM for all five sentences, but the difference is not nearly so substantial as it is for the synthetic autoregressive sources. Narrowband spectrograms of the original sentence 2 (first 10 000 samples), the tree coder output with the DPCM code generator $(M,L)=(4,10)$, and the tree coder output with the SDPCM code generator $(M,L)=(4,10)$ are shown in Figs. 10, 11, and 12, respectively. These spectrograms indicate that SDPCM yields a better reproduction of the original spectrogram than DPCM-

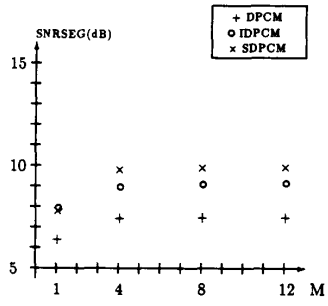


Fig. 7. Improvement versus retained paths—BWH(3), $R = 1$.

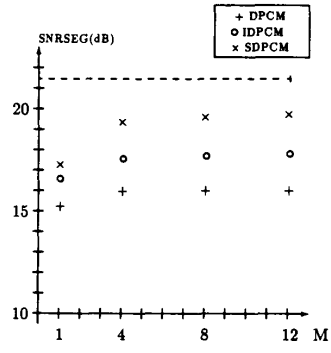


Fig. 8. Improvement versus retained paths—LAW(3), $R = 2$.

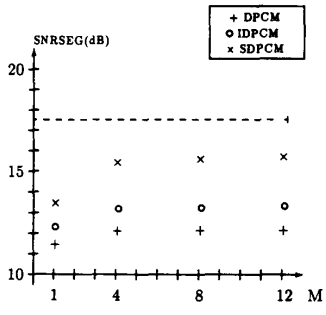


Fig. 9. Improvement versus retained paths—BWH(3), $R = 2$.

based codes, particularly during the first 2000 samples and in the 1500–2000 Hz band around 5000 samples. Informal subjective listening tests reveal a slight preference for the SDPCM code generator.

The success of the smoothing algorithms depends heavily on the accuracy of the models and model parameters in (8)–(12). For speech, the AR source model does not account for the long-term redundancy present due to the pitch pulse. Thus, the smoother performance may be poor in time intervals around the occurrence of a pitch pulse. Further, the smoothing algorithms used here are based upon the assumption of white measurement noise in (11). Since the measurement noise is actually the quantization noise, and we are using only a four-level quantizer at 2 b/sample, this white noise assumption may be violated. The extensive simulations required to verify these two conjectures have not been performed. However, the relatively slight performance increment for SDPCM over DPCM here is likely a result of these two factors.

TABLE VI
PERFORMANCE FOR SPEECH SOURCES IN SNR/SNRSEG (dB): DPCM

	$M = 1$	$M = 4$	$M = 8$	$M = 12$
sent1	16.65/16.97	19.80/19.57	20.26/20.09	20.39/20.17
sent2	17.31/16.73	19.79/19.25	20.67/19.69	21.15/19.90
sent3	16.15/15.18	18.96/17.77	19.79/18.25	19.96/18.49
sent4	14.26/15.77	16.91/17.96	17.15/18.24	17.53/18.41
sent5	14.09/15.70	16.58/18.14	17.33/18.65	17.49/18.88

TABLE VII
PERFORMANCE FOR SPEECH SOURCES IN SNR/SNRSEG (dB): SDPCM

	$M = 1$	$M = 4$	$M = 8$	$M = 12$
sent1	16.16/17.10	19.84/20.12	20.23/20.41	20.54/20.49
sent2	17.73/17.35	20.77/19.86	21.24/20.04	20.84/20.38
sent3	17.03/15.87	19.26/18.21	19.83/18.50	19.50/18.61
sent4	14.19/16.10	16.44/18.03	16.91/18.43	17.08/18.61
sent5	14.33/16.04	16.37/18.47	17.40/18.78	17.02/19.03

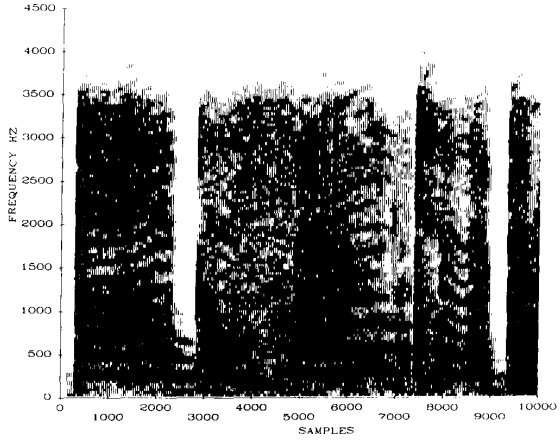


Fig. 10. Spectrogram (sent2, original).

VII. CONCLUSIONS

Smoothed DPCM codes employing an MMSE fixed-lag smoother have been introduced and shown to outperform IDPCM and DPCM based code generators for tree coding several synthetic sources at rate $R = 2$ b/sample, $M = 1$, 4, 8, 12, and $R = 1$ b/sample with $M \geq 4$. With $M = 1$ at rate 1, SDPCM has a smaller SNRSEG than both IDPCM and DPCM, which is likely because the whiteness assumption on the measurement noise is violated, thus leading to poorly matched smoothing algorithms. Rate 2 b/sample tree coding of speech with adaptive DPCM and SDPCM code generators

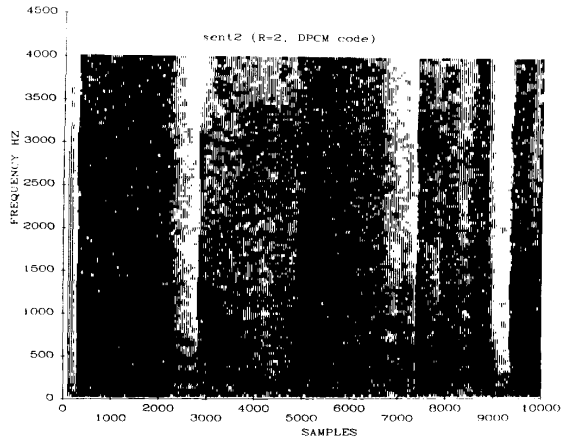


Fig. 11. Spectrogram (sent2, DPCM).

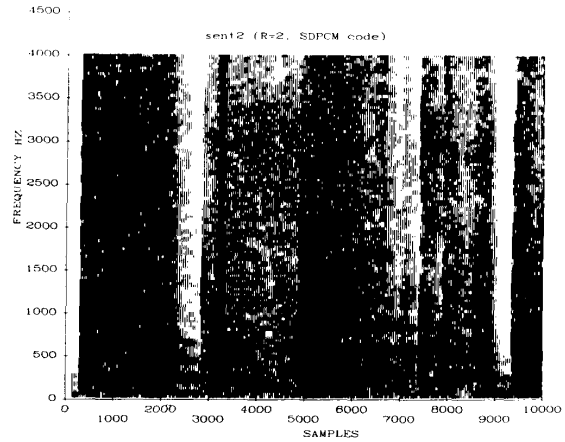


Fig. 12. Spectrogram (sent2, SDPCM).

show that SDPCM has slightly higher SNRSEG values, and spectrogram analyses reveal that SDPCM provides output speech that more closely emulates the original speech than does DPCM. It is evident that the source and measurement models used in developing the smoothing algorithms must be accurate and are extremely important to the success of the SDPCM codes.

Appendix A
Fixed-Lag Smoothing Algorithms

Consider a linear, discrete-time dynamic system described in state-space form. Hereon, we will use lower-case italic letters for scalars, lower-case boldface letters for vectors, and capital boldface letters for matrices. At each time instant n, $\boldsymbol{x}(n)$ denotes the p-dimensional system state vector and $y(n)$ denotes the noisy measurement sample. The system model can be described by both the process and the measurement equations. The former specifies the dynamics of the system in terms of the system state vector $\boldsymbol{x}(n)$, and the latter describes the structure of the available measurement. In state-space notation, these equations are

$$\boldsymbol{x}(n+1) = \boldsymbol{\Phi}(n)\boldsymbol{x}(n) + \boldsymbol{v}(n) \tag{A.1}$$

$$y(n) = \boldsymbol{c}(n)\boldsymbol{x}(n) + q(n) \tag{A.2}$$

where $\boldsymbol{\Phi}(n)$ is a known p-by-p state transition matrix, $\boldsymbol{c}(n)$ denotes a known p-dimensional measurement vector, and the measurement noise sample $q(n)$ and the process noise vector $\boldsymbol{v}(n)$ are assumed to be zero-mean, white noise processes with variance and covariance matrix given, respectively, by

$$E[q(k)q(\ell)] = \begin{cases} \sigma_q^2(k), & k = \ell; \\ 0, & k \neq \ell, \end{cases} \tag{A.3}$$

and

$$E[\boldsymbol{v}(k)\boldsymbol{v}(\ell)] = \begin{cases} \boldsymbol{Q}(k), & k = \ell; \\ \boldsymbol{0}, & k \neq \ell. \end{cases} \tag{A.4}$$

The problem of interest here is to find the MMSE estimate of the system state vector $\boldsymbol{x}(i)$, given the data measured up to time n, $\psi^n = \{y(1), y(2), \cdots, y(n)\}$. This information-processing operation is called filtering if $i = n$, prediction if $i > n$, and smoothing if $i < n$. In our work, we are only concerned with filtering and fixed-lag smoothing, in which there is always a fixed delay J between the measurement and the availability of the smoothed state estimate. Given the measurement set ψ^n, it can be shown [9] that the estimate $\hat{\boldsymbol{x}}(i)$ that minimizes the mean-squared error $E\{[\boldsymbol{x}(i) - \hat{\boldsymbol{x}}(i)]^2 \mid \psi^n\}$ is the conditional mean $\hat{\boldsymbol{x}}(i) = E[\boldsymbol{x}(i) \mid \psi^n]$. Hence, we define the filtered state vector estimate $\hat{\boldsymbol{x}}_f(n \mid n) = E[\boldsymbol{x}(n) \mid \psi^n]$ and the lag-J smoothed state vector estimate $\hat{\boldsymbol{x}}_s(n - J \mid n) = E[\boldsymbol{x}(n - J) \mid \psi^n]$. The filtering operation implies that the state vector estimate $\hat{\boldsymbol{x}}_f(n \mid n)$ must be available at time n, given the measurements up to time n. One widely used filtering algorithm is the Kalman filter which offers a recursive MMSE filtered estimate. On the other hand, smoothing differs from filtering in that the state vector estimate at time n, $\hat{\boldsymbol{x}}_s(n \mid n + J)$, need not be available until time $n + J$. Note that the measurements required to compute the filtered estimate $\hat{\boldsymbol{x}}_f(n \mid n)$ are a proper subset of those required to compute the smoothed estimate $\hat{\boldsymbol{x}}_s(n \mid n + J)$, which implies that a fixed-lag smoother offers a state estimate with an error variance no larger than that of the Kalman filter. The specific fixed-lag smoother used here is the stable finite-dimensional algorithm described by Chirarattananon and Anderson [10]. Throughout this paper, we only consider the single-lag smoother with $J = 1$ to minimize the total coding delay $L + J$ and to hold complexity to a minimum.

In order to formulate a fixed-lag smoother, we first recall the recursive difference equations of the Kalman filter. Define the filtered state error vector $\boldsymbol{d}(n)$ as the difference between the state vector $\boldsymbol{x}(n)$ and its filtered estimate vector $\hat{\boldsymbol{x}}_f(n \mid n)$, $\boldsymbol{d}(n) = \boldsymbol{x}(n) - \hat{\boldsymbol{x}}_f(n \mid n)$. Its error covariance matrix is $\boldsymbol{P}(n) = E[\boldsymbol{d}(n)\boldsymbol{d}^T(n)]$. Define the predicted state error vector $\bar{\boldsymbol{d}}(n, n-1)$ as the difference between the state vector $\boldsymbol{x}(n)$

and its one-step predicted estimate vector $\boldsymbol{\Phi}(n-1)\hat{\boldsymbol{x}}_f(n-1 \mid n-1)$, and denote its error covariance matrix as $\bar{\boldsymbol{P}}(n) = E[\bar{\boldsymbol{d}}(n,n-1)\bar{\boldsymbol{d}}^T(n,n-1)]$. From Kalman filtering theory for a white measurement noise assumption [11], the filtered state vector estimate is

$$\hat{\boldsymbol{x}}_f(n \mid n) = \boldsymbol{\Phi}(n-1)\hat{\boldsymbol{x}}_f(n-1 \mid n-1) + \boldsymbol{k}(n)\left[y(n) - \boldsymbol{c}(n)\boldsymbol{\Phi}(n-1)\hat{\boldsymbol{x}}_f(n-1 \mid n-1)\right] \tag{A.5}$$

where the Kalman gain $\boldsymbol{k}(n)$ and the two error covariance matrices $\boldsymbol{P}(\cdot)$ and $\overline{\boldsymbol{P}}(\cdot)$ are updated recursively as

$$\overline{\boldsymbol{P}}(n-1) = \boldsymbol{\Phi}(n-1)\boldsymbol{P}(n-1)\boldsymbol{\Phi}^T(n-1) + \boldsymbol{Q}(n-1), \tag{A.6}$$

$$\boldsymbol{k}(n) = \overline{\boldsymbol{P}}(n-1)\boldsymbol{c}^T(n)\left[\boldsymbol{c}(n)\overline{\boldsymbol{P}}(n-1)\boldsymbol{c}^T(n) + \sigma_q^2(n)\right]^{-1}, \tag{A.7}$$

$$\boldsymbol{P}(n) = [\boldsymbol{I} - \boldsymbol{k}(n)\boldsymbol{c}(n)]\overline{\boldsymbol{P}}(n-1). \tag{A.8}$$

To define the single-lag smoother ($J = 1$), we introduce the quantity

$$\boldsymbol{D}(n-1) = \overline{\boldsymbol{P}}(n-1)\left[\boldsymbol{P}(n-1)\boldsymbol{\Phi}^T(n-1)\right]^{-1} \tag{A.9}$$

and then the smoothed state vector estimate is given by [10]

$$\hat{\boldsymbol{x}}_s(n-1 \mid n) = \boldsymbol{D}^{-1}(n-1)\hat{\boldsymbol{x}}_f(n \mid n) + \boldsymbol{D}^{-1}(n-1) \cdot [\boldsymbol{D}(n-1) - \boldsymbol{\Phi}(n-1)]\hat{\boldsymbol{x}}_f(n-1 \mid n-1). \tag{A.10}$$

Finally, the specific smoothed estimate sample at time n is obtained by

$$\hat{x}_s(n-1) = [1 \quad 0 \quad 0 \quad \cdots \quad 0]\hat{\boldsymbol{x}}_s(n-1 \mid n). \tag{A.11}$$

Since the recursive algorithm offers a tracking capability to follow the time variations of the measurements, the choice of initial states is not crucial. In our work, the initial states are set to zero.

Appendix B

The objective performance measures used in this work are the signal-to-noise ratio (SNR) defined as

$$\mathrm{SNR} = 10\log_{10}\frac{\langle s^2(k)\rangle}{\langle [s(k) - \hat{s}(k)]^2\rangle} \tag{B.1}$$

where $\langle\cdot\rangle$ denotes time averaging over the entire utterance, and the segmental SNR (SNRSEG) given by

$$\mathrm{SNRSEG} = \frac{1}{K}\sum_{j=1}^{K}\mathrm{SNRB}_j \tag{B.2}$$

where SNRB_j is the SNR in (B.1) over the jth block of speech data.

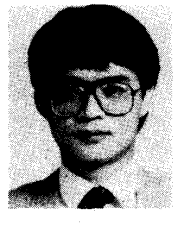
Appendix C

The five sentences used in this work are:

1) "The pipe began to rust while new." (female speaker)
2) "Add the sum to the product of these three." (female speaker)
3) "Oak is strong and also gives shade." (male speaker)
4) "Thieves who rob friends deserve jail." (male speaker)
5) "Cats and dogs each hate the other." (male speaker).

References

[1] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1971.
[2] G. W. Aughenbaugh, J. D. Irwin, and J. B. O'Neal, Jr., "Delayed differential pulse code modulation," in *Conf. Rec., Princeton Conf. Inform. Sci.,* Mar. 1968, pp. 125–130.
[3] C. C. Cutler, "Delayed encoding: Stabilizer for adaptive coders," *IEEE Trans. Commun.*, vol. COM-19, pp. 898–907, Dec. 1971.
[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] M. L. Sethia and J. B. Anderson, "Interpolative DPCM," *IEEE Trans. Commun.*, vol. COM-32, pp. 729–736, June 1984.
[6] J. D. Gibson, "Tree coding versus differential encoding of speech: A perspective," presented at the 1981 Int. Symp. Inform. Theory, Santa Monica, CA, Feb. 9–12, 1981.
[7] J. D. Gibson and T. R. Fischer, "Alphabet constrained data compression," *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 443–457, May 1982.
[8] M. L. Sethia, "Digitization of analog sources," Ph.D. dissertation, Dep. ECSE, Rensselaer Polytech. Inst., Troy, NY, Jan. 1984.
[9] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice-Hall, 1979.
[10] S. Chirarattananon and B. D. O. Anderson, "The fixed-lag smoother as a stable finite-dimensional linear system," *Automatica,* vol. 7, pp. 657–669, 1971.
[11] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1986.
[12] N. S. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.
[13] R. C. Reininger and J. D. Gibson, "Backward adaptive lattice and transversal predictors in ADPCM," *IEEE Trans. Commun.*, vol. COM-33, pp. 74–82, Jan. 1985.
[14] R. A. McDonald, "Signal-to-noise and idle channel performance of differential pulse code modulation systems—Particular applications to voice signals," *Bell Syst. Tech. J.*, vol. 45, pp. 1123–1151, Sept. 1966.
[15] C. W. Law, "Code design for tree speech coders," Commun. Res. Lab., McMaster Univ., Hamilton, Ont., Canada, Rep. CRL-35, Mar. 1976.
[16] B. J. Bunin, "Rate-distortion functions for Gaussian Markov processes," *Bell Syst. Tech. J.*, pp. 3059–3074, Nov. 1969.



Wen-Whei Chang (S'86–M'89) was born in Chungli, Taiwan, on December 4, 1958. He received the B.S. degree in communication engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1980 and the M.Eng. and Ph.D. degrees in electrical engineering from Texas A&M University, College Station, in 1985 and 1989, respectively.

Since August 1989, he has been an Associate Professor in the Department of Communication Engineering at National Chiao-Tung University, Hsinchu, Taiwan. His current research interests are in data compression and speech synthesis.

Dr. Chang is a member of Tau Beta Pi.

Jerry D. Gibson (S'73–M'73–SM'83) was born in Fort Worth, TX, on May 12, 1946. He received the B.S. degree in electrical engineering from the University of Texas at Austin in 1969 and the M.S. and Ph.D. degrees from Southern Methodist University, Dallas, TX, in 1971 and 1973, respectively.

He has held positions at General Dynamics, Fort Worth, TX, from 1969 to 1972, the University of Notre Dame from 1973 to 1974, the Defense Communications Agency in the summer of 1974, and the University of Nebraska, Lincoln, from 1974 to 1976. In 1976 he joined Texas A&M University, College Station, where he currently holds the J. W. Runyon, Jr. Professorship in the Department of Electrical Engineering. He is coauthor of the book *Introduction to Nonparametric Detection with Applications* (New York: Academic, 1975). He recently published the textbook *Principles of Digital and Analog Communications* (New York: Macmillan, 1989) and a chapter (with K. Sayood) entitled "Lattice Quantization" in *Advances in Electronics and Electron Physics* (New York: Academic, 1988). His research interests include speech processing, data compression, digital communications, and image processing.

Dr. Gibson received the 1990 Frederick Emmons Terman Award from the Electrical Engineering Division of the American Society for Engineering Education. He is a member of Eta Kappa Nu and Sigma Xi. He was Associate Editor for Speech Processing for the IEEE TRANSACTIONS ON COMMUNICATIONS from 1981 to 1985. He is currently Associate Editor for Communications for the IEEE TRANSACTIONS ON INFORMATION THEORY and a member of the IEEE Information Theory Society Board of Governors. He is also General Co-Chairman of the 1993 International Symposium on Information Theory to be held in San Antonio, TX.