Proc. IEEE Int. Conf. Commun. (Philadelphia, PA), pp. 42.1.1-42.1.6, June 1988

A VARIABLE RATE ADAPTIVE TRANSFORM CODER FOR THE DIGITAL STORAGE OF AUDIO SIGNALS

R.W. Tansony[1], and P. Kabal[2]

1. Bell Canada Switching Systems Research
   220 Simcoe St., Toronto, Ontario, Canada, M5T 1T4.
2. INRS Telecommunications
   3 Place du Commerce, Montreal, Quebec, Canada, H3E 1H6.

ABSTRACT: A transform coding algorithm designed for the mass storage of variable quality audio signals is presented. The storage of both speech and music is considered. Automatic silence deletion and signal order, energy, and bandwidth estimation are employed to provide a continuously variable bitrate adaptively matched to the characteristics of the input signal. Test results show that the algorithm offers storage savings of more than 75% over linear-PCM coders, and more than 63% over equivalent quality log-PCM coders. Results also show improved performance over the CCITT standard wideband coder. The complexity of the algorithm is such that it can be implemented in real time on existing digital signal processors.

1.0 INTRODUCTION: This paper describes an adaptive transform coding algorithm designed for the digital archiving of audio signals. The storage of both speech and music is considered. The algorithm is designed to respond efficiently to the wide variations in source material type and fidelity found in large historical archive collections.

Existing audio coding algorithms are generally optimized for the transmission of a single source (speech) at a fixed bitrate. A variable bitrate approach however is much more attractive for the mass storage of multiple source audio signals. The proposed archiving algorithm is based on the Adaptive Transform Coder (ATC). Bit assignment is derived via a Linear Predictive Coding (LPC) fit to the Discrete Cosine Transformed (DCT) signal coefficients. Total bit assignment is not constant however as in existing ATC algorithms. Assignment is signal energy and bandwidth dependent to permit constant quality coding of variable fidelity material. Automatic silence deletion is implicit in the algorithm. The coder thus adapts automatically to the type and fidelity of the input signal to maintain optimal storage efficiency.

Formal evaluation results show that the coded signal is rated indistinguishable from the original in more than 90% of all trials. On average, the algorithm provides storage savings of more than 75% over linear PCM (16 bit), and more than 63% over log-PCM coders. Results also show improved performance over the CCITT standard wideband coder.

2.0 AUDIO ARCHIVES: The archiving algorithm described in this paper has been designed to meet the needs of a large historical audio archive collection. Such collections are found in the museums of major cities around the world. Coder design was based on one such archive, containing 50% speech and 50% music -- comprised of interviews with artists and dancers, political dissertation, native songs and dances, etc.. Audio selections are held on a wide variety of storage media, including wax cylinders, shellac and vinyl disks, cassettes, and various vintages of reeled tape. Thus the source material varies in quality and fidelity over a wide range. A typical museum collection could easily contain 10000 to 20000 hours of material.

Each selection is copied onto modern 1/4 inch analog magnetic reel-to-reel tape before presentation to the public. These 1/4 inch copies must currently be re-recorded every 5 to 8 years due to the aging of the analog storage medium. This is a time consuming and expensive task, and great interest is being shown throughout the audio archiving field in the potential economies to be realized through digital optical storage. The algorithm presented in this paper has been designed to respond to both speech and music signals of widely varying quality thus providing an algorithm to be used in conjunction with optical storage media to reduce the cost of re-copying archive collection material.

3.0 THE ALGORITHM: The algorithm is based on the Adaptive Transform Coder [1,2], and thus requires that two bitstreams be multiplexed and stored:

1) The Main Information Bitstream, carrying quantized signal transform coefficients;

2) The Side Information Bitstream, carrying the signal spectral estimate (used to calculate quantizer bit allocations and step sizes). The side information bitstream also carries the signal order and signal energy estimates.

The algorithm is outlined in Fig. 1. Input signal samples, x(n), are buffered into blocks of length N=256 samples. The input signal block is then windowed by a cosine rolloff window, w(n), pre-emphasized, and normalized by dividing by the total block energy, V, to produce a block of normalized samples, x''(n), where:

$$x''(n) = \frac{x(n)w(n) - a.x(n-1)w(n-1)}{V}$$

for $0 <= n <= N-1$; (3.1)

a = 0.7 (pre-emphasis factor);
N = 256 (block length).

The block energy, V, is stored as side information (as a 16 bit quantity).

The normalized signal block is then Discrete Cosine Transformed (DCT) to provide a set, T(i), of transform coefficients:

$$T(i) = c(i) \sum_{n=0}^{N-1} x''(n) \cos\left[\frac{(2n+1)(pi)i}{2N}\right]$$

for $0 <= i <= N-1$; (3.2)

where,

$c(i) = 1$, for $i=0$;
$\sqrt{2}$, for $1 <= i <= N-1$;

pi = 3.14159.

An adaptive quantization process is employed at this stage of the algorithm to efficiently allocate bits between individual transform coefficients and the coder side information bitstream, in a way which reduces the perceived effects of quantization noise. All quantization parameters are carried by the side information bitstream in the form of a smooth spectrum estimate.

For each input signal block, autocorrelation values, r(i), are used via Levinson/Durbin recursion [3] to generate a Linear Predictive smooth estimate of the input signal spectrum, v(n), using the standard recursive equations:

$$k(i) = \frac{\sum_{j=1}^{i-1}\left[a_{i-1}(j)\, r(i-j)\right] - r(i)}{e(i-1)}$$

$$a_i(i) = -k(i);$$

$$a_i(j) = a_{i-1}(j) + k(i)\, a_{i-1}(i-j);$$

$$e(i) = \left[1 - k^2(i)\right] e(i-1); \quad (3.3)$$

(The recursion begins by setting e(0)=r(0) so that k(1) = -r(1)/r(0).)

The appropriate autocorrelation values to be used in Equation (3.3), are calculated directly as a circular autocorrelation of the input signal using:

$$r(n) = \frac{1}{2} \sum_{i=0}^{2N-1-n} x''(n)\, x''(i+n)$$

$$+ \frac{1}{4} \sum_{i=0}^{n-1} \Big[x''(n)\, x''(n-1-i) + x''(N-n-i)\, x''(N-1-i) \Big].$$

(3.4)

The Levinson/Durbin recursion results in an optimal (for minimum mean square prediction error) set of predictor coefficients, a(i), at each stage. To form the smooth spectrum estimate, the algorithm performs a 2N-point Discrete Fourier Transform on the filter coefficient set, af(i):

$$af(i) = \begin{cases} 1, & \text{for } i=0; \\ -a(i), & \text{for } 1 <= i <= Np; \\ 0, & \text{for } Np+1 <= i <= 2N-1; \end{cases}$$

(3.5)

where, Np = the number of predictors used to form the smooth spectrum estimate.

The DFT result is inverted and scaled to provide the smooth spectrum estimate, v(n):

$$v(n) = \frac{G}{DFT[af(i)]} ;$$

(3.6)

where, G is a scaling factor which forces (from Equation (3.2)):

$$\sum_{n=0}^{N-1} v(n) = \sum_{n=0}^{N-1} T^2(n).$$

The number of predictors used to generate the spectrum estimate, Np, is varied based on the current signal order estimate. Signal order is determined using the Final Prediction Error (FPE) criterion introduced by Akaike [4]:

$$FPE(i) = 1 + \left[\frac{i+1}{N} \right] \left[\frac{N}{N-1-i} \right] e^2$$

for 1 <= i <= Nmax;

where,

e, is the current prediction error (Eqn.(3.3));

N, is the transform block length;

Nmax, is an upper limit on the range of model orders considered feasible given a priori knowledge of the signals to be coded;

i, is the order of the model.

The algorithm calculates the value of FPE(i) at each stage of the Levinson/Durbin recursion (Eqn.(3.3)). The FPE(i) array is then searched for a minimum. The value of j, which provides the minimum FPE(j) is the signal order estimate. The number of predictors used is thus set to Np = j. This procedure adaptively attempts to quantify the point of diminishing return, where improvements in signal quality resulting from the addition of further predictors is not worth the extra bits required to store those predictors. The predictor count, Np, is stored once per block as side information (as a 4 bit quantity). Np is limited to a member of the set: {0,1,2,3,4,5,7,9,10,11,13,14,15,16,17,18}. The smooth spectrum estimate is stored via a set of Np log-area quantized reflection coefficients, k(i), (Equation 3.3) using the log-area function [5]:

$$f(x) = \log\left[\frac{1+x}{1-x} \right] \quad \text{for } a <= x <= b.$$

(3.7)

The function limits (a,b), and the quantizer bit allocations for each reflection coefficient were optimized using archive test data.

To determine the total number of bits to be allocated in quantizing the block transform coefficients, a search of the spectrum estimate is made for the index j, where:

$$v(i) \leq \frac{N \cdot q}{100} \quad \text{for } j \leq i \leq N-1. \tag{3.8}$$

The value j is thus the frequency index at which the signal power spectrum drops (and remains) below q% of the total energy of the block (q=5 was used during simulation runs). The block bandwidth estimate is thus:

$$f_{max} = \frac{j\ Fs}{2N} \quad Hz; \tag{3.9}$$

where, Fs is the input signal sampling frequency. The bandwidth estimate is damped to prevent audible changes in bit allocation. The block bitrate goal, G, is determined directly from the damped bandwidth estimate, Fd, using the following linear relation:

$$G = v\ Fd + z \quad kb/s. \tag{3.10}$$

(Values of v=0.0072 kb, and z=15.42 kb/s, were determined empirically using archive test data.)

The total mainstream block bit allocation required to meet the bitrate goal, is:

$$B = \frac{N\ G\ p}{Fs} \quad bits \tag{3.11}$$

where, p is a silence deletion multiplier of the form:

$$p = \frac{V}{t} \qquad p \leq 1.0 \tag{3.12}$$

and, t is a silence deletion threshold factor (a value of t=1000 was used during simulation runs). Silence deletion is thus achieved using a graduated approach which relies on the fact that distortion is less easily detected by the human ear at low signal levels, than at higher signal levels.

Given the total block bit assignment, bit allocation amongst the transform coefficients is performed under the assumption that the coefficients can be approximated by an independent identically distributed Gaussian source, using [2]:

$$b(i) = \frac{B}{N} + \frac{1}{2}\log_2\left[v(n)\right] - \frac{1}{2N}\sum_{j=0}^{N-1}\log_2 v(j) \tag{3.13}$$

Each bit allocation is then rounded to the nearest integer, and the calculations of Equation (3.13) are performed again, this time considering only those coefficients which were assigned b(i)>0 bits during the first iteration. This brings the realized block bit assignment closer to the desired goal, B. Evaluation results demonstrated that this two step iteration produced bit assignments close to the optimal allocations attained by integer-optimized methods (such as that proposed by Segall [6]), with significantly less calculation.

The transform coefficients are then uniformly quantized using the bit allocations b(i), and multiplexed with the side information (reflection coefficient set, k(i); signal mean energy, V; and signal order estimate, Np) before being written to the mass storage medium. Quantizer step sizes are adjusted once per block to match Max Gaussian - optimized values [7].

The decoding algorithm performs the following functions to reverse the operation of the encoder:

1) re-generate the smooth spectrum estimate using k(i), and Np;
2) re-calculate quantizer step sizes, and bit allocations;
3) recover the quantized transform coefficients;
4) inverse DCT the transform coefficients, and scale the result using V, to produce the output signal.

Full details on coder design are available in [9].

4.0 SIMULATION RESULTS: The variable rate archiving algorithm was tested through a 32 bit floating point Fortran simulation. Formal evaluation results on 12 audio test files representative of material found in historical archives, showed that coded signals were rated indistinguishable from the original in more than 90% of all trials. Coder bitrates varied from 31.2 kb/s for the simplest signal (a 400 Hz tone), to 96.3 kb/s for the most complex (wideband jazz group with multiple vocalists) -- (both signals were sampled at 16 kHz). Music files included narrowband (3500 Hz), and wideband (7 kHz) piano, pan pipe, and male/female vocalists, sampled at 16kHz and 8 kHz. Other files contained narrowband, and wideband male and female speech.

Tests on a narrowband female speech segment, showed that whether sampled at 8 kHz, or 16 kHz, the algorithm produced essentially the same bitrate (44.4 versus 42.7 kb/s respectively). Both of these rates, as expected, were significantly lower than the measured storage rate for the wideband version of the same segment: 66.3 kb/s. Similar results were found for male speech, and music. The coder thus demonstrated its ability to operate at a fixed, high sampling rate, yet store only the fundamental information required to re-construct the signal.

On average, the coder was found to offer storage savings of 75% over the original linear PCM (16 bit) files, and 63% over equivalent quality log-PCM coded files.

Other benchmark testing showed that the archiving algorithm was able to produce higher quality ratings at lower bitrates than the CCITT standard wideband coder [8], in 3 out of 6 test files (31.2, 43.8, and 44.4 kb/s versus 64 kb/s). In the other 3 cases, the archiving algorithm increased its bitrate by up to 27% over that of the CCITT coder; however, poor ratings for the CCITT coder in these cases show that the increase was clearly necessary to maintain high signal quality (17%, 17%, and 18% identical, versus 92%, 100%, and 100% identical).

The archiving algorithm thus responded well to the bandwidth and complexity of the input signals, providing a consistent, high quality output. Full details on test results are available in [9].

REFERENCES

1. Tribolet, J. M., and R.E. Crochiere, FREQUENCY DOMAIN CODING OF SPEECH, IEEE Transactions on ASSP, Vol. ASSP-27, No. 5, October 1979, pp. 512-530.

2. Zelinski, R., and P. Noll, Adaptive Transform Coding of Speech Signals, IEEE Transaction on ASSP, Vol. ASSP-25, No. 4, August 1977, pp. 299-309.

3. Oppenheim, A.V., ed., APPLICATIONS OF DIGITAL SIGNAL PROCESSING, Prentice-Hall, Englewood Cliffs, N.J., USA, 1978, Section 3.6.

4. Akaike, H., FITTING AUTOREGRESSIVE MODELS FOR PREDICTION, Ann. Inst. Statist. Math, Vol. 21, 1969, pp. 243-247.

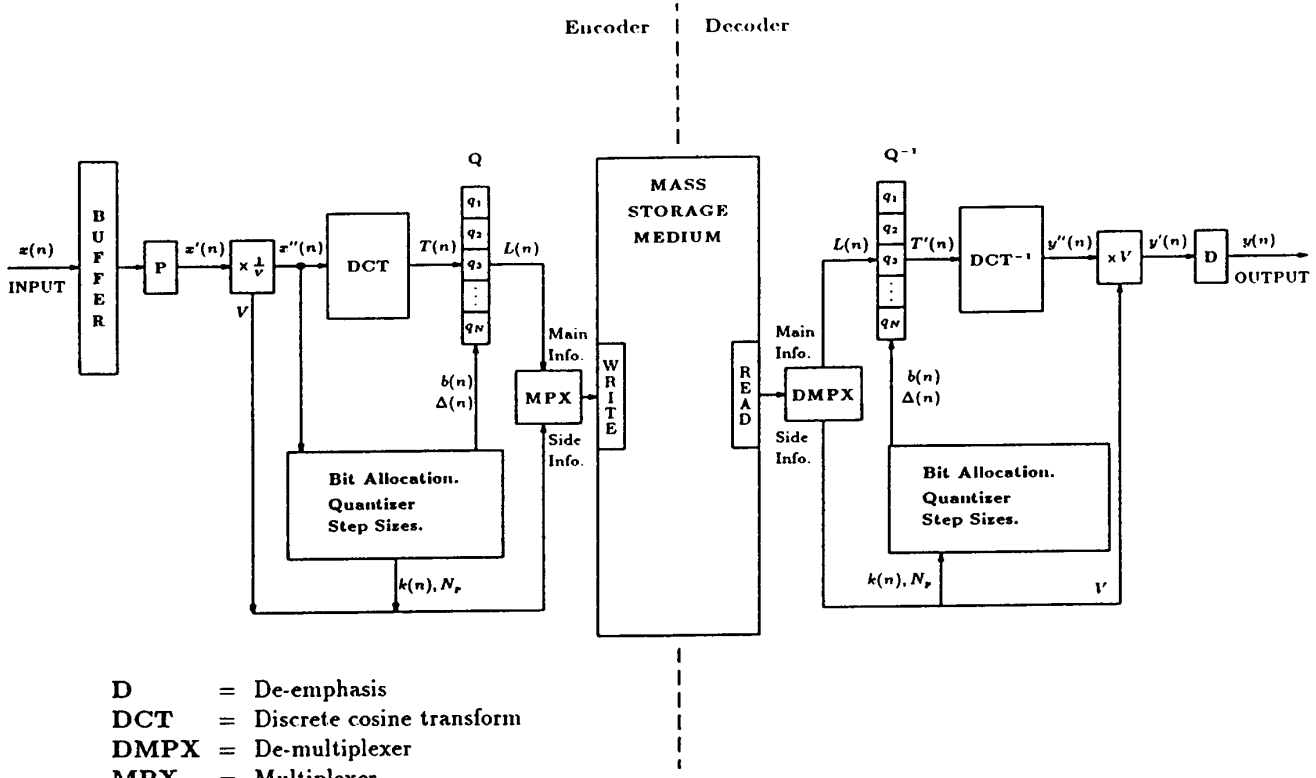
5. Gray, A.H., and J.D. Markel, QUANTIZATION AND BIT ALLOCATION IN SPEECH PROCESSING, IEEE Transactions on ASSP, Vol. ASSP-24, No. 6, December 1976. pp. 459-473.

6. Segall, A., BIT ALLOCATION AND ENCODING FOR VECTOR SOURCES, IEEE Transactions on Information Theory, Vol. IT-22, No. 2, March 1976, pp. 162-169.

7. Max, J., QUANTIZING FOR MINIMUM DISTORTION, IRE Transactions on Information Theory, Vol. IT-6, 1960, pp. 7-12.

8. Taka, M., P. Murmelstein, P. Combescure, and F. Westall, OVERVIEW OF THE 64 kb/s (7 kHz) AUDIO CODING STANDARD, Proc. IEEE International Conference on ASSP, 1986, pp. 17.1.1 - 17.1.6.

9. Tansony, R.W., A VARIABLE RATE ADAPTIVE TRANSFORM CODER FOR THE DIGITAL STORAGE OF AUDIO SIGNALS, Masters Thesis, McGill University, Montreal, Quebec, 1987.



D = De-emphasis
DCT = Discrete cosine transform
DMPX = De-multiplexer
MPX = Multiplexer
P = Pre-emphasis

Figure 1 Algorithm Overview