To be presented at ICASSP, May 2001

JOINT AND TANDEM SOURCE-CHANNEL CODING WITH DELAY CONSTRAINTS

Jongtae Lim and David L. Neuhoff

EECS Department, University of Michigan, Ann Arbor, MI 48109
{jongtlim, neuhoff}@eecs.umich.edu

ABSTRACT[1]

Two common source-channel coding strategies, joint and tandem, are compared on the basis of distortion vs. delay by analyzing specific representatives of each when transmitting analog data samples across a binary symmetric channel. Channel-optimized transform coding is the joint source-channel coding strategy; transform coding with Reed-Solomon coding is the tandem strategy. For each strategy, formulas for the mean-squared error and delay are found and used to minimize distortion subject to a delay constraint, for data modeled as Gauss-Markov. The results of such optimizations suggest there is a threshold such that when the permissible delay is above this threshold, tandem coding is better, and when below the threshold, channel-optimized transform coding is better.

1. INTRODUCTION

Two common approaches to source-channel coding for communicating analog data samples across a noisy channel are: (1) tandem source-channel coding, in which a source code designed without regard to the possibility of channel errors is followed by a channel code designed without regard to the source, and (2) joint source-channel coding, in which source and channel code are jointly designed to combat channel errors. One common example of the joint is channel-optimized quantization, where no explicit channel code is used, but the quantizer is designed to be robust to channel errors.

Shannon showed that no system could have better performance than the best tandem source-channel coding system. However, over the years there has been considerable interest in joint source-channel coding with the motivation that it can attain better performance with less complexity and/or delay than tandem source-channel coding. On the other hand, little quantitative evidence for this claim has appeared in the literature. Recently, we looked for such evidence by quantitatively comparing representative systems of each type on the basis of distortion vs. complexity [1,2]. In the present work, we expand our investigation to find whether joint source-channel coding achieves better performance with less delay than tandem source-channel coding.

To avoid idiosyncrasies, we examine systems that are as "plain vanilla" as possible. Specifically, as the joint source-channel code, we consider the channel-optimized transform code studied by Vaishampayan and Farvardin [3]. As the tandem source-channel code, we consider a conventional transform code (optimized for a noiseless channel) followed by a Reed-Solomon channel code, as studied previously by the authors in [1,2]. We do not use entropy coding with either system, because its performance tends to be highly method specific, making it unlikely that one could choose a representative plain vanilla version. We also choose mean-squared error (MSE) as the fidelity measure which, despite its faults, is by far the most commonly used. We evaluate the performance of these systems on a Gauss-Markov (first-order AR) source over binary symmetric channels (BSC). The delay of these systems is derived and expressed in units of data samples by considering buffering delay caused by block processing.

Just as the results of previous studies [1,2] suggest there is a threshold such that tandem coding is better than joint when and only when the complexity is larger than the threshold, the results of the present study suggest there is a threshold such that tandem coding is better than joint when and only when the permissible delay is above this threshold.

2. THE SOURCE-CHANNEL CODING SYSTEMS

In this section, we describe the representative systems mentioned previously. The source is a stationary Gauss-Markov random process, denoted $\{U_i\}$, with zero mean, unit variance, and correlation coefficient $\rho=E[U_iU_{i+1}]$. The channel is a binary symmetric channel (BSC) (stationary and memoryless) whose error probability is denoted p. The number of channel uses available to transmit one source sample is denoted R and called the channel rate.

Joint: channel-optimized transform coding

For an integer L to be specified, an L-dimensional Karhunen-Loeve transform maps the L source data samples $\underline{U} = (U_1,U_2,\ldots,U_L)$ into an L-dimensional vector of coefficients $\underline{V}$. Each coefficient V_i, which is itself Gaussian with variance σ_i^2, is then scalar quantized with a quantizer Q_i having $N_{s,i} = 2^{R_{s,i}}$ levels denoted $y_{i,1}<\ldots<y_{i,N_{s,i}}$. The distortion of Q_i is $D_{s,i} = E(V_i-Q_i(V_i))^2$. Each quantization level $y_{i,j}$ is assigned an $R_{s,i}$-bit binary sequence $\underline{w}_{i,j}$ that is produced when V_i is quantized to $y_{i,j}$. Let $\underline{B}_i$ denote the binary

[1]Work supported by ARO MURI grant DAAH04-96-1-0337.

sequence produced when quantizing V_i. When R channel uses are available per source symbol, the rate allocations $R_{s,i}$, the quantization levels $y_{i,j}$, and the binary sequences $\underline{w}_{i,j}$ are chosen to minimize the overall mean-squared error distortion $D = \frac{1}{L}\sum_{i=1}^{L} D_{s,i}$ subject to the overall rate constraint $\frac{1}{L}\sum_{i=1}^{L} R_{s,i} \le R$. This is done with the method of [3] for the given source (parameterized by ρ) and the binary symmetric channel (parameterized by p).

Tandem: Transform Plus Reed-Solomon coding

The tandem strategy uses a transform code as above, except that the rate allocations and quantization levels are optimized for the source by conventional means without regard to the channel. The binary sequence $\underline{w}_{i,j}$ is simply the binary representation of the integer j. We define $R_s = \frac{1}{L}\sum_{i=1}^{L} R_{s,i}$ to be the source code rate. The channel code is an (n,k,m) Reed-Solomon (R-S) code with $n = 2^m - 1$, $k \le n$, and channel code rate denoted $R_c = k/n$. With such a code, k data symbols are encoded into n channel symbols, where each data symbol consists of m bits from the transform code and each channel symbol consists of m bits to be input to the BSC. The decoder can correct any pattern of $t = \lfloor \frac{n-k}{2} \rfloor$ or fewer channel symbol errors, where a channel symbol error is said to occur whenever the channel introduces one or more bit errors within an m-bit channel symbol.

The R-S encoder is chosen to be systematic in the sense that each codeword has the k data symbols that it encodes as its first k channel symbols. When the channel rate R is specified, it is required that the source and channel codes be chosen so that $R_s/R_c = R$. We do not assume that the number of bits, LR_s, produced by the source code is equal to, a divisor of, or a multiple of the number of bits, km, encoded by one R-S codeword. Rather, the bits produced by the transform code are concatenated into a single stream that is parsed into blocks of km for encoding by the R-S encoder. It follows that the bits $\underline{B}_i$ describing coefficient V_i might be embedded in two or more symbols from one or two codewords[2].

The R-S decoder operates as follows. When the BSC output, considered as an n-symbol sequence, is within Hamming distance t of some R-S codeword, the decoder produces the first k symbols of that codeword. On the other hand, when the Hamming distance between the BSC output and each codeword exceeds t, the received sequence and decoder are said to FAIL, and the decoder simply produces the first k channel output symbols. The resulting decoded symbol stream is parsed into blocks of LR_s bits and presented to the transform decoder. We have found that using this strategy when the received sequence FAIL's works better than either simply producing the first k symbols of the closest codeword, or declaring an ERASURE and replacing the missing transform coefficients with zeroes.

[2]The values of R_s, n, k and m considered are such that $\underline{B}_i$ will never be spread over three or more codewords.

3. DISTORTION

The distortion of the channel-optimized transform coding may be computed by the method of [3].

We give a brief overview of the method for computing the tandem code distortion that was presented in [1,2]. The distortion of the tandem code is determined by the source parameter ρ, channel parameters p and R, source code parameters L and R_s, and channel code parameters n, k and m. The distortion computation is complicated by the fact that the binary sequence $\underline{B}_i$ representing coefficient V_i is, in general, embedded in a different way in the channel coded bit stream than the bits $\underline{B}_{i+L}$ representing coefficient V_{i+L}, even though V_{i+L} is quantized in the same way as V_i. For example, $\underline{B}_1$ will be the first $R_{s,1}$ bits of the first channel codeword, but depending on the values R_s, L, k and m, $\underline{B}_{L+1}$ might appear later in the same codeword or might be divided between the first and second codeword, to mention just two of the possibilities. Moreover, the bits of $\underline{B}_{L+1}$ might not begin a symbol, whereas those of $\underline{B}_1$ do.

In effect, we need to average the MSE of coefficients of the same type as V_i over the different ways that the bits representing such coefficients can be embedded in the channel coded bit stream. To this end, let g denote the greatest common divider of LR_s and km, let $N_s = km/g$ and $N_c = LR_s/g$. It is easy to see that $N_s LR_s = N_c km$ is the least common multiple of LR_s and km. That is, N_s is the smallest integer such that the bits produced by N_s applications of the transform code can be packed into an integer number of R-S codewords. Accordingly, we consider distortion associated with encoding $N_s L$ data samples. The corresponding transform coefficients, $V_1,\ldots,V_{N_s L}$, are quantized as described earlier, with the understanding that V_i and V_{i+L} are identically distributed and quantized by identical quantizers.

The MSE may now be written

$$D = \frac{1}{N_s L}\sum_{i=1}^{N_s L} D_i \;, \qquad (1)$$

where $D_i = E[(V_i - \hat{V}_i)^2]$, and where V_i, the ith transform coefficient, is Gaussian with variance $\sigma_i^2 = \sigma_{i \bmod L}^2$, and $\hat{V}_i$ is its decoded reproduction. To compute the D_i's we assume that when the decoder does not FAIL, it always produces the correct decision. As can be seen from [4], this is a good assumption when the error correcting capability t is at least 4. It follows that

$$D_i = D_{s,i}(R_{s,i})\, P(NF_i) + E[(V_i - \hat{V}_i)^2 \mid F_i]\, P(F_i) \;, \qquad (2)$$

where F_i is the event that the received sequence containing the bits $\underline{B}_i$ representing V_i FAILS, or if $\underline{B}_i$ is spread over two received sequences, that one or both of these FAILS, and where NF_i is the NO FAIL event, i.e. the complement of F_i. By the usual argument of combined source-channel coding (cf. [5], pp. 179-181),

$$E[(V_i - \hat{V}_i)^2 | F_i] = D_{s,i} + D_{c,i} \;, \qquad (3)$$

where $D_{s,i} = D_{s,i}(R_{s,i})$ is the quantizer distortion described earlier and $D_{c,i}$ is the channel distortion defined by

$$D_{c,i} = \sum_{j,j'=1}^{N_{S,i}} (y_{i,j}-y_{i,j'})^2 \; P_i(j) \; P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_i) \; , \qquad (4)$$

where $P_i(j)$ is the probability that the quantizer for V_i produces the jth level $y_{i,j}$, and $P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_i)$ is the probability that the channel decoder outputs the binary sequence for $y_{i,j'}$ given that the channel input was the binary sequence for $y_{i,j}$ and that the sequence(s) containing $\underline{B}_i$ FAILED. To find an expression for $P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_i)$, one must take into account the position of $\underline{B}_i$ in the encoded bit stream relative to symbol and codeword boundaries. This is done by separately considering the cases that $\underline{B}_i$ is entirely contained within one R-S codeword and that $\underline{B}_i$ is spread over two codewords. We omit the details. However, the end result is an expression for $P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_i)$ that when substituted into (4) and combined with (1)-(3) yields a computable expression for D.

4. DELAY

The delay introduced by a system when transmitting symbol U_i is the number of data samples U_{i+1}, U_{i+2}, ... that subsequently arrive at the encoder until the time that the reproduction $\hat{U}_i$ is produced by the decoder. The delay of a source-channel code is the maximum delay for any source symbol. Though delay can be caused by encoding/decoding hardware, we consider only buffering delay because it is always present and it is the only type of delay that cannot be reduced by adding resources.

The delay for a channel-optimized transform code is simply the dimension L of its transform, which is the time required to buffer the L samples of one block.

For the analysis of delay in a tandem system, we consider the encoding of N_sL data samples as described in Section 3. When N_sLR_s bits are packed into N_c codewords, we take g bits as the "packing unit", where g is the greatest common divisor of LR_s and km. Thus, LR_s transform output bits consist of N_c units and one codeword consists of N_s units. We need to pack N_sN_c units coming from N_s transform blocks into N_c codewords. The appropriate expression for delay depends on the values of LR_s and km.

(1) If km is a multiple of LR_s, i.e., N_sLR_s=km, then N_s transform outputs are nicely packed into one codeword. Thus, the delay is the time required for buffering N_s transform outputs for channel encoding, i.e. $N_sL=(km/LR_s)L$. (2) If km is not a multiple of LR_s, but is greater than LR_s, thne one channel codeword, consisting of N_s units, takes N_s units from more than one transform block. For a given R-S codeword, let M_T denote the number of transform blocks from which it receives units. M_T is the number of transform block outputs that need to be buffered, for packing units into this codeword. One can show that the maximum value of M_T is $1+\lceil (km-g)/LR_s \rceil$. Thus, the delay in this case is $1+\lceil (km-g)/LR_s \rceil L$. (3) For the case that LR_s is a multiple of km, the transform outputs are nicely packed into N_c codewords, and thus the delay is the time required for buffering source samples for transform encoding and is given by L. (4) If LR_s is not a multiple of km and LR_s is greater than or equal to km, then the N_c units coming from one transformation are divided and packed into more than one codeword. The codeword that takes the last unit of the transformation cannot be sent right after the transformation since it needs to include other units from the next transformation. When the codeword containing the last unit of the transformation is decoded, all bits resulting from the prior transformation are available and the source decoding for the prior transformation can be performed. Thus, there is need to buffer two transform blocks and the delay is 2L.

After simple manipulations, the delay expressions for the four cases can be combined into:

$$\tau = \left(1+\left\lceil \frac{km-g}{LR_s} \right\rceil\right) L \; , \qquad (5)$$

where g is the greatest common divisor of km and LR_s.

5. OPTIMIZING DISTORTION WITH A DELAY CONSTRAINT

Channel-optimized transform code

For specific choices of the source parameter ρ, channel parameters R and p, and delay constraint T, a computer program evaluates the distortion of the transform code for the largest transform dimension L in the set {1,2,4,8,...., 128} such that the delay is no larger than T.

Tandem source-channel code

For specific choices of the source parameter ρ and channel parameters R and p, and a delay constraint T, a computer program finds the best choices of source code parameters L and R_s, and channel code parameters n, k and m subject to the constraints that $R_s \le R$, $nR_s/R \le k$, $n=2^m-1$, LR_s is an integer and $\tau(L,R_s,n,k,m) \le T$. To do so, the expressions of Sections 3 and 4 are used to compute the MSE and delay, with parameter values varying in three nested loops. The values of L come from the set {1,2,4,8,....,128}; the values of R_s are such that LR_s equals $1,2,......,\lfloor RL \rfloor$; the values of m come from the set {1,2,...,12}; and finally $k=\lceil nR_s/R \rceil$. Note that for a given choice of L, R_s, k and m, the program must choose N_s and N_c as described in Section 3.

6. RESULTS AND CONCLUSIONS

Representative results of the optimizations described above are provided by Figure 1, which plots the SNR in dB of channel-optimized and tandem source-channel coding for the case that the source has correlation coefficient ρ=0.9, the BSC has error probability $p=2\times10^{-2}$, and the channel is used R=5 times per data sample. The bottom line shows the performance of transform coding optimized without regard to the channel, but used directly on the channel without channel coding. It is easy to see that there is no benefit to increasing the delay, i.e. the transform dimension. However, as delay increases, the SNR of channel-optimized transform coding improves. Specifically, there is about a 5 dB gain over conventional transform coding for T =4, and the gain increases until it saturates at about 11 dB for T≅30. At this point, there is no benefit to further increases in dimension.

Not surprisingly, the performance of optimized tandem source-channel codes also increases with delay. For small values of delay, its SNR is not as large as that of channel-optimized transform coding, while for larger values, it is larger. In particular, tandem coding becomes better at around 50 samples of delay. At this point, although the SNR of channel-optimized transform coding has saturated, allowing additional delay permits the tandem code to continue to improve. Thus, there appears to be a threshold such that channel-optimized transform coding gives better performance than tandem coding when and only when the delay is constrained to be less than this threshold. Also shown in the figure is the SNR of a tandem system consisting of conventional transform code with L=128 and an ideal channel code (the transform code has rate R times the capacity of the BSC and its encoded bits are assumed to be unaffected by channel errors), and also the best possible performance of any tandem code (the Shannon distortion-rate function of the source evaluated at R times the capacity of the BSC).

Tables 1 and 2 show the delay threshold and the gain of the tandem system over the joint system at 5 times the threshold. The latter is intended as an indicator of how much can be gained with the additional delay of tandem coding. Notice that the delay thresholds decrease and the gains increase as the channel becomes more reliable and more channel uses are available. This means that contrary to what one might initially think, channel coding is more immediately useful, i.e. useful with moderate delay, when the channel is more reliable.

In summary, the distortion vs. delay performance of representative tandem and joint source-channel codes has been computed and compared. The results suggest that there is a threshold such that when the permissible delay is above this threshold, tandem coding is better, and when below the threshold, channel-optimized transform coding is better. In other words, the results suggest using joint source-channel coding when low delay is required, and using tandem coding when better performance is needed and substantially more delay is allowed.

REFERENCES

[1] J. Lim and D.L. Neuhoff, "Source-channel coding strategies: tandem coding vs. channel-optimized quantization," IEEE ISIT00, p. 237, June 2000.

[2] J. Lim, D.L. Neuhoff and T.C. Nolan, "Allocating complexity between source and channel codes," Proc. Workshop Data Compression Processing Techniques, pp. 669-679, Redstone Arsenal., AL, Dec. 1998.

[3] V.A. Vaishampayan and N. Farvardin, "Optimal block cosine transform image coding for noisy channels," *IEEE Trans. Comm.*, vol. 38, pp. 327-336, Mar. 1990.

[4] R.J. McEliece and L. Swanson, "On the decoder error probability for Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 32, pp. 701-703, Sep. 1986.

[5] N.S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

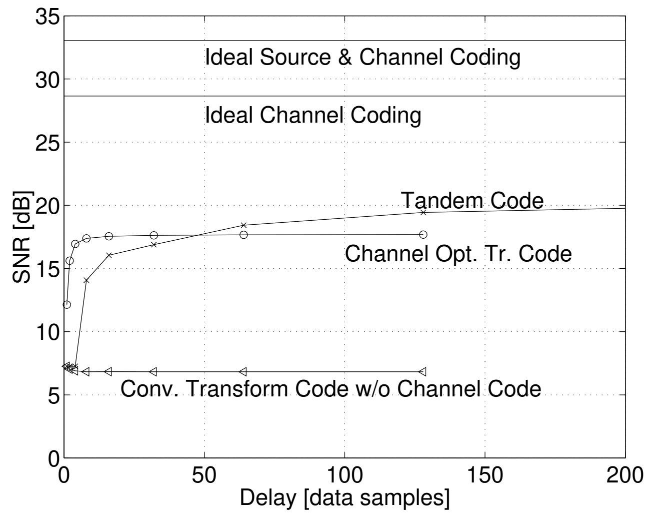

Figure 1: Performance of channel-optimized transform coding and tandem coding for R=5, $p=2\times10^{-2}$ and $\rho = 0.9$

Table 1. Delay thresholds and gains of tandem over joint coding at 5 times threshold for $\rho = 0.9$.

		R		
		3	4	5
p	0.02	400/0.6dB	80/0.8dB	47/1.9dB
	0.01	125/0.7dB	55/2.1dB	25/2.5dB
	0.001	115/0.7dB	60/1.8dB	25/2.4dB

Table 2. Delay thresholds and gains of tandem over joint coding at 5 times threshold for R=5.

		ρ		
		0.70	0.90	0.95
p	0.02	40/1.2dB	47/1.9dB	30/2.9dB
	0.01	20/2.3dB	25/2.5dB	15/2.4dB
	0.001	15/1.8dB	25/2.4dB	30/3.2dB