JOINT SOURCE-CHANNEL DECODING FOR VARIABLE-LENGTH ENCODED DATA BY EXACT AND APPROXIMATE MAP SEQUENCE ESTIMATION

MoonSeo Park and David J. Miller

Department of Electrical Engineering
The Pennsylvania State University
Rm. 227-C EE West Bldg., University Park, PA 16802

ABSTRACT

Joint source-channel decoding based on residual source redundancy is an effective paradigm for error-resilient data compression. While previous work only considered fixed rate systems, the extension of these techniques for variable-length encoded data was recently independently proposed by the authors [6], [7] and by Demir and Sayood [1]. In this paper, we describe and compare the performance of a computationally complex exact maximum *a posteriori* (MAP) decoder [6], [7], its efficient approximation [6], [7], an alternative approximate MAP decoder [1], and an improved version of this decoder suggested here. Moreover, we evaluate several source and channel coding configurations. Our results show that the approximate MAP technique from [6], [7] outperforms other approximate methods and provides substantial error protection to variable-length encoded data.

1. INTRODUCTION

In the classical approach to communications system design, the source and channel coder problems are considered independently. This is motivated by Shannon's source-channel separation theorem. However, Shannon's theorem effectively assumes that the source coder is able to remove all redundancy, and that the channel coder is able to correct all errors. In practical systems, there is often residual redundancy in the source encoder's output [10]. Thus, in practice, one can often improve performance by considering the source and channel coder designs *jointly*. In this work[1], we focus on joint source-channel (JSC) methods for decoding, based on residual source redundancy [10], [8]. Rather than attempting to remove redundancy by entropy coding or by improved prediction, JSC methods retain redundancy so that it can be used as a form of implicit channel protection by the decoder, which acts as a statistical estimator of the transmitted information sequence. This approach has been demonstrated to provide some error correction capability and has been found to achieve substantial performance benefits for very noisy channels [10],[8],[2]. Both maximum *a posteriori* (MAP) [10],[8],[5] and minimum mean-squared error (MMSE) [8],[2] methods have been suggested.

Until very recently, all JSC methods based on residual redundancy assumed fixed length codewords even though variable length coding (VLC) is widely used in image and video coding. However, the nature of variable rate systems greatly complicates the estimation problem at the decoder. Whereas under certain statistical assumptions optimal MAP sequence estimation in the fixed rate case can be efficiently achieved by dynamic programming on a trellis, this is not possible in the variable rate case. Recently, several different variable rate approaches were proposed. In [6], [7] a computationally complex exact MAP decoding method and an efficient approximation were both suggested. In [1], a different approximate MAP method was independently proposed. In [3], a MAP decoding method was proposed for the case where the number of transmitted symbols is unknown. This technique was applied to transmission of compressed video in [4]. Here, as in [6], [7], [1] we focus on the more typical situation where the number of transmitted symbols is known or provided[2], thus fixing the length of the decoded sequence. This constraint makes the estimation problem especially difficult to (tractably) solve. We describe the methods in [6], [7], [1] and also suggest an improvement to the approach in [1]. We also indicate why the approximate method from [6], [7] should outperform the other approximate techniques. Simulation results confirm this performance advantage. In addition to comparing the new decoding methods, we also investigate fundamental design choices within a DPCM coding environment. Our results indicate ranges of channel conditions where i) variable rate coding is preferred over fixed rate coding; ii) where, given JSC decoding, explicit channel coding is not required; and iii) where JSC decoding in conjunction with particular types of explicit channel coding is most effective.

2. EXACT AND APPROXIMATE MAP DECODING

Consider a (packet-sized) sequence of information symbols $\underline{i} = (i_1, i_2, \ldots, i_T)$, $i_t \in \mathcal{I}$, with $\mathcal{I}$ the symbol alphabet, of size N. Each index i_t is Huffman-coded, resulting in an encoded bit stream $\underline{b} = (b_1, b_2, \ldots, b_B)$. The JSC decoder, which is assumed to know the length B, receives the corrupted bit stream $\underline{r} = (r_1, r_2, \ldots, r_B)$. Its objective is to

This work was supported in part by the NSF under Career Award IRI-9624870.

The authors would like to thank Dr. Robert Van Dyck from The Pennsylvania State University for a suggestion which helped to improve the paper.

[1]Related material can be found in http://www.personal.psu.edu/mxp190/research/pub/publication.html.

[2]The transmitted length of the sequence will either be known, or needs to be conveyed as side information.

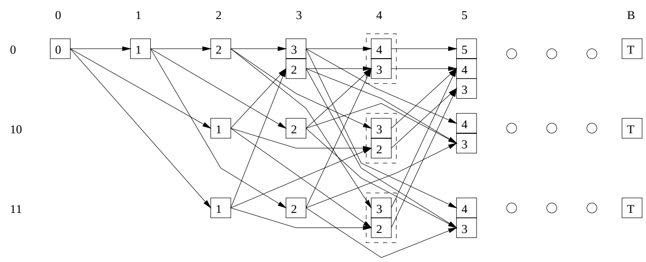

Figure 1: Directed graph representation for VLC decoding/MAP sequence estimation in [6], [7]. The horizontal axis corresponds to bit time and the vertical axis to Huffman-coded symbol choices. The boxes represent decoding states, with the enclosed numbers the length of the symbol sequence required to reach the state.

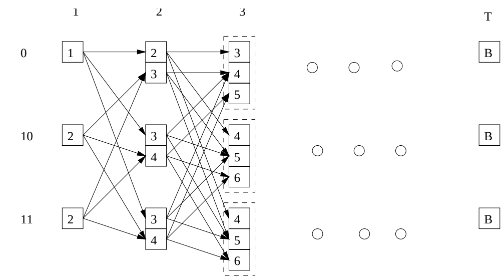

Figure 2: Directed graph representation for VLC decoding/MAP sequence estimation. An approximate method based on this structure was proposed in [1]. The horizontal axis corresponds to symbol time and the vertical axis to Huffman-coded symbol choices. The boxes represent decoding states, with the enclosed numbers the length of the bit sequence required to reach the state.

estimate the sequence $\underline{i}$ given the observed $\underline{r}$. As in previous work [10],[8],[2], residual redundancy is modeled by assuming the random process $\underline{I}$ to be first-order Markovian, based on the conditional probabilities $P[I_t = i_t | I_{t-1} = i_{t-1}]$ and the initial probabilities $P[I_1 = i_1]$. Moreover, the channel is assumed binary symmetric, with bit error rate (BER) ϵ. Given this model, the decoder objective addressed here is to estimate the sequence $\hat{\underline{i}}$ of maximum *a posteriori* probability, i.e., to realize the sequence MAP rule: $\hat{\underline{i}} = \arg\max_{\underline{i}} P[\underline{I} = \underline{i} | \underline{r}]$. For the fixed rate case, sequence MAP decoding can be posed as a search for the optimal path through an N-state trellis, efficiently implementable via dynamic programming [8]. In the variable rate case, given the transmitted length constraints, the optimal sequence MAP decoder can also be achieved by dynamic programming as proposed in [6], [7], although the problem cannot be represented on a trellis, but rather must be posed on a more complicated directed graph, due to the lack of synchronization between the VLC symbols and the received bits.

Two alternative directed graph representation are shown in figures 1 and 2 for $N = 3$ and a particular set of Huffman codewords. The structure in Figure 1 was proposed in [6], [7] and used to develop both exact and approximate MAP method. An approximate method based on the structure in Figure 2 was proposed in [1]. In both figures, the small boxes represent possible decoder states, with a decoded sequence $\hat{\underline{i}}$ specified by a connected path through the directed graph. The statistical modeling assumptions guarantee that, for each graph representation, one can compute $\log(\mathrm{P}[\underline{r}|\underline{I} = \hat{\underline{i}}]\mathrm{P}[\underline{I} = \hat{\underline{i}}])$ (which differs from $\log \mathrm{P}[\underline{I} = \hat{\underline{i}}|\underline{r}]$ by the constant $\log \mathrm{P}[\underline{r}]$) by summing metric contributions associated with each branch along the decoded path. Each branch contributes a term of the form $\log \mathrm{P}[I_t = \hat{i}_t | I_{t-1} = \hat{i}_{t-1}] + \log \mathrm{P}[\underline{v_t} | I_t = \hat{i}_t]$, with the second probability based on the BER ϵ and the Hamming distance between the Huffman codeword for $\hat{i}_t$ and its corrupted version $\underline{v_t}$ (a subsequence of bits from $\underline{r}$). A dynamic programming algorithm for realizing the exact MAP rule was suggested in [6], [7] for the structure in Figure 1. This approach could be applied to the graph in Figure 2 as well. However, in both representations, the number of states grows with the sequence length and can become very large. This complexity necessitated the development of approximate schemes, proposed independently in [6], [7] and [1], based, respectively, on the structures in Figure 1 and Figure 2.

Although there is no reason to favor either graph structure for exact MAP decoding (as dynamic programming can be applied in either case), the structures differ markedly when one considers approximate MAP decoding. For the graph in Figure 1, consider the set of states within any dashed box. In each such set, all states exhaust the same number (n) of bits and terminate a symbol subsequence at bit n using the same decoded symbol. Although the figure only indicates the dashed sets at $n = 4$, such sets are defined at all bit lengths. In [6], [7], an approximate MAP decoding method was proposed, tailored to this graph. As in standard dynamic programming, this approximate method involves a "forward" operation for propagating and accumulating metrics and saving pointers to previous states, and a "traceback" operation to find the best state sequence. However, the suggested method approximates the exact "forward" operation by saving, at each bit length, only the *best* state (in the sense of the accumulated log probability metric) from each dashed set. Denote the jth state in the kth dashed set at bit length n by $s_n(k, j)$. Then, as an example of state reduction, for the first dashed set at $n = 4$, the method chooses between $s_4(1, 1)$ and $s_4(1, 2)$. Here, $s_4(1, 1)$ corresponds to the state with symbol length 4 and $s_4(1, 2)$ to the state with symbol length 3. The saved state is the one which terminates the symbol sequence of maximum (log) *a posteriori* probability. Note that this state reduction operation at bit length n effectively prunes states (and branches) from the graph, thus restricting the candidate symbol sequences (and states) considered in state reduction at subsequent bit lengths $m > n$. Using the same example, if $s_4(1, 1)$ is chosen over $s_4(1, 2)$ at $n = 4$, then $s_4(1, 2)$ is removed from the graph, along with all branches emanating from this state. Thus, when subsequently performing state reduction for the first dashed set at $n = 5$, symbol sequences which include the branch from $s_4(1, 2)$ to

$s_5(1,2)$ will not be considered as candidates[3]. Note that although the cardinality of each dashed set grows with the bit length, the number of *retained* states for each set remains constant, at one. Thus, the computational complexity of the method remains manageable for increasing bit length. We describe this approach as "bit-constrained".

Now consider approximate decoding for the graph in Figure 2 [1]. Again the dashed sets group together states which terminate symbol sequences with the same decoded symbol. However now, rather than exhausting the same number of bits, all states in any dashed group exhaust the same number of *symbols*. The method in [1] approximates the "forward" operation to reduce complexity in a fashion similar to [6], [7], by saving, at each *symbol length*, only the best state from each dashed set. Thus, the number of saved states remains constant for increasing symbol length, with the computational complexity of the resulting method again manageable. Clearly, the approximate methods from [6], [7] and [1] implement quite similar operations. Thus, we might expect these methods to achieve roughly the same performance. However, there are potential problems with the approach in [1], as can be seen by identifying the criterion effectively optimized by the method's state reduction procedure. Note that whereas state reduction in [6], [7] compares symbol sequences of different lengths which exhaust the same number of bits, the method in [1] compares symbol sequences of the *same* length which exhaust *different* numbers of bits. We describe this approach as "symbol-constrained".

It is not obvious which state reduction rule is to be preferred in general. However, a clear difficulty with the state reduction method in [1] is that it actually only implements an *approximation* to the symbol-constrained MAP rule. In particular, note that for both decoder structures, the accumulated metric consists of only two types of terms: the log-likelihood of a symbol sequence starting at symbol time 1 and ending at some symbol time t, $\log P[\underline{I_t} = \underline{i_t}]$, and the conditional log-likelihood of a received bit sequence $\log P[\underline{r_n}|\underline{I_t} = \underline{i_t}]$. The neglected term in the log *a posteriori* probability $(-\log P[\underline{r_n}])$ need not be included in the bit-constrained case [6], [7] since all competing terminating state hypotheses $\{s_n(k,j)\}$ involve the *same* received bit sequence. However, while this term is also omitted in the symbol-constrained case, this introduces suboptimality, as different terminating state hypotheses from the same set correspond to *different bit sequences*. Thus, each candidate state hypotheses $s_t(k,j)$ requires a different term $(-\log P[\underline{r_{tkj}}])$, where $\underline{r_{tkj}}$ is the associated received bit sequence for the j-th state in the k-th dashed set in Figure 2, which is omitted in [1]. Moreover, one cannot easily correct this problem, since exact calculation of the term $\log P[\underline{r_{tkj}}]$ requires at least as much computation as implementation of the exact MAP rule itself ! As a practical solution, we assume sequences to be equally likely, i.e. for a sequence of length n, $\log P[\underline{r_n}] \simeq -n \log 2$. We then accordingly modify the symbol-constrained approach [1] to include this term. Next, we will consider the effect of this modification, and more generally, evaluate the performance of all of the JSC decoders discussed here.

3. RESULTS

Similar to [1], we evaluated the various decoding methods for DPCM coding of images. In our experiments, both variable and fixed rate systems were evaluated. Moreover, for variable rates we considered systems both with and without explicit channel coding. For channel coding we considered both i) binary convolutional encoding (BCE) (applied to Huffman coded quantization indices) and ii) non-binary convolutional encoding (NCE) applied directly to the indices (followed by Huffman coding). For decoding, in case i), we suggest an (inner) decoding of the convolutional code followed by an (outer) JSC decoding. In case ii), we use JSC decoding followed by NCE decoding. In [1], a system comprised of NCE-based channel encoding and JSC decoding was compared with a more conventional system based on BCE and (standard Viterbi) channel decoding. There, the NCE-based system was found to achieve superior performance. However, in the BCE-based scheme, no JSC decoding was used. Here, for the BCE-based scheme, we use both inner (Viterbi-based) decoding and outer (JSC) decoding.

For BCE, we chose a rate 2/3 convolutional encoder with constraint length 2 as in [9]. This was applied to the Huffman coded indices associated with a 6 level quantizer. In this case, rather than using ϵ, the JSC decoder used the effective BER of the concatenated BCE-channel-Viterbi decoder system. For NCE, we matched the code to the source alphabet so as to preserve residual redundancy in the sequence, as in earlier work [1]. We chose an NCE with 6 input symbols and 12 output symbols. Denote the quantization index at symbol time t by $x_t \in \{0,1,2,3,4,5\}$. Then, the NCE output is given by the equation:

$$i_t = \begin{cases} x_t & \text{if } x_{t-1} < 3 \\ x_t + 6 & \text{otherwise} \end{cases}$$

Our particular choices for BCE and NCE were made both to achieve good channel robustness and a common rate (3 bits/sample) so that techniques could be fairly compared.

For variable rates, in conjunction with BCE and Viterbi-based (inner) decoding, we implemented the bit-constrained approximate MAP (outer) decoder. This is denoted BC-AMAP + BCE. For variable rates, in conjunction with NCE, we implemented the exact MAP decoder [6], [7] (denoted MAP + NCE), the bit-constrained approximation [6], [7] (BC-AMAP + NCE)[4], the symbol-constrained approximate method [1] (SC-AMAP + NCE) and its modification suggested here (MSC-AMAP + NCE). We also implemented a 14-level variable rate quantizer with decoding solely based on BC-AMAP, i.e. without channel coding (denoted BC-AMAP). For fixed rates, we used an 8 level quantizer both with conventional DPCM decoding (denoted

[3]Moreover, if a removed state at bit length $n-1$ was the only source of branch connections for some state at bit length n, then this state must also be removed from the graph. For example, if $s_4(1,1)$ is removed, then so is $s_5(1,1)$.

[4]Even though it is not presented here, we also implemented a variation on the bit-constrained approximate method, wherein we retain only one state among those which exhaust the same number of symbols, instead of among those which terminate with the same symbol. In this case, the number of states grows, but manageable complexity can still be maintained. The performance for this scheme was very similar to that of BC-AMAP + NCE.

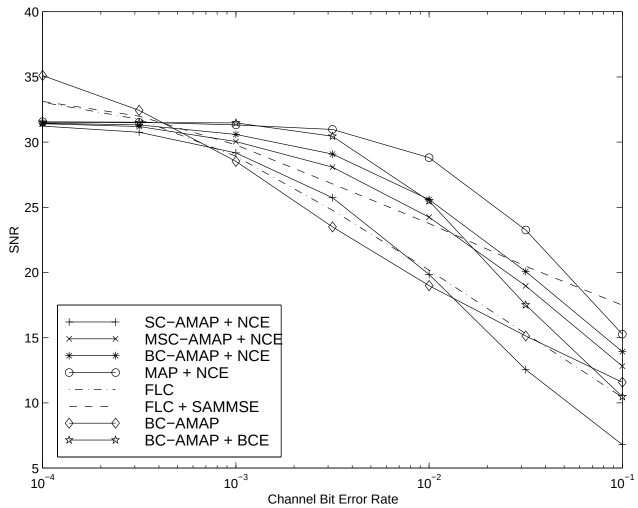

Figure 3: SNR versus channel BER performance of several techniques based on averaging mean-squared error from 10 different channel realizations.

FLC) and with sequence-based, approximate MMSE decoding [2] (denoted FLC + SAMMSE). For variable rates, we chose a packet length of $T = 64$. The overhead associated with specifying the bit length B for each packet was estimated and included in the overall bit rate. All the systems achieved a rate of approximately 3 bits/sample in coding the 256×256 Lenna image.

Figure 3 shows the SNR of several decoding techniques. We are interested in several types of comparisons: 1) comparison of the variable rate MAP methods; 2) comparison of the NCE-based schemes with the BCE-based method; 3) comparison of JSC decoding both with and without explicit channel coding; and 4) variable vs. fixed rate coding. Comparing all the MAP methods that were used with NCE, we find that while exact MAP decoding achieves the best results, the more practical BC-AMAP + NCE approach, with computational complexity roughly forty times smaller, also achieves good performance. The largest gap is roughly 3 dB at $\epsilon = 0.01$. Both these methods substantially outperform SC-AMAP + NCE from [1]. BC-AMAP + NCE achieves 5.7 dB gain at $\epsilon = 0.01$ and 7.1 dB gain at $\epsilon = 0.1$. The modification of SC-AMAP + NCE significantly improves its performance. MSC-AMAP + NCE is 4.4 dB better than SC-AMAP + NCE at $\epsilon = 0.01$ and 6.0 dB better at $\epsilon = 0.1$.

In comparing all the curves excepting exact MAP, we can identify different BER regimes where particular configurations are preferred. For the low BER range, BC-AMAP achieves the best results. Thus, here, JSC decoding for variable rates provides adequate protection without wasting bits on explicit channel coding. Moreover, as one might expect, in this regime variable rate coding is preferred over fixed rate coding. For the low-to-medium BER range ($\epsilon \in [0.0005, 0.01]$), the best performance is achieved by the scheme with inner binary convolutional decoding and outer JSC decoding. Note that, based on these results and the poor results for the binary convolutional coding scheme in [1], we must conclude that inner binary convolutional decoding in conjunction with outer JSC decoding is a more effective scheme than the approach in [1], which did not use JSC decoding. Note also that in this BER regime, we again find that a variable rate scheme outperforms the fixed rate ones. For medium-to-high BER ($\epsilon \in [0.01, 0.05]$), the best performance is achieved by combining NCE with JSC decoding. Finally, for high BER ($\epsilon > 0.05$), we have the expected result that fixed length coding (with MMSE-based JSC decoding [2]) outperforms the variable rate systems.

In conclusion, we have described several methods for extending JSC decoding to VLC data. In comparing these techniques, we have found the approximate MAP method in [6], [7] to achieve the best results, in comparison with alternative approximate decoders. Future work may extend the new methods for MMSE decoding and for other compression environments, such as subband coding.

4. REFERENCES

[1] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," *Proc. of the Data Compression Conf.*, Snowbird, U.T., March 30 - April 1, 1998, pp. 139-148.

[2] D. J. Miller and M. Park, "A sequence-based, approximate MMSE decoder for source coding over noisy channels using discrete hidden Markov models," *IEEE Trans. on Comm.*, Vol. 46, No. 2, pp. 222-231, Feb. 1998.

[3] A. H. Murad and T. E. Fuja, "Joint source-channel decoding of variable-length encoded sources," *Proc. of the Info. Theory Workshop*, Killarney, Ireland, June 22-26, 1998.

[4] A. H. Murad and T. E. Fuja, "Exploiting the residual redundancy in motion estimation vectors to improve the quality of compressed video transmitted over noisy channels," *Proc. of the IEEE Intern. Conf. on Image Proc.*, Chicago, I.L., October 4 - 7, 1998.

[5] M. Park and D. J. Miller, "Low-delay, optimal MAP state estimation in HMMS with application to symbol decoding," *IEEE Sig. Proc. Letters*, Vol. 4, No. 10, pp. 289-292, Oct. 1997.

[6] M. Park and D. J. Miller, "Decoding entropy-coded symbols over noisy channels by MAP sequence estimation for asynchronous HMMs," To appear in *Proc. of the Conf. on Info. Sciences and Systems*, Princeton, N.J., March 18 - 20, 1998.

[7] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," Submitted to *IEEE Trans. on Comm.* for publication, 1998.

[8] N. Phamdo and N. Farvardin, "Optimal detection of discrete Markov sources over discrete memoryless channels - application to combined source channel coding," *IEEE Trans. on Info. Theory*, Vol. 40, No. 1, pp. 186-193, Jan. 1994.

[9] J. G. Proakis, *Digital Communications*, McGraw-Hill, Inc., New York, 1995, p. 472.

[10] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. on Comm.*, Vol. 39. No. 6, pp. 838-846, June 1991.