

Iterative List Decoding of Concatenated Source-Channel Codes

Ahmadreza Hedayat

*Multimedia Communications Laboratory, The University of Texas at Dallas, TX 75083-0688, USA
Email: hedayat@utdallas.edu*

Aria Nosratinia

*Multimedia Communications Laboratory, The University of Texas at Dallas, TX 75083-0688, USA
Email: aria@utdallas.edu*

Received 6 October 2003; Revised 17 June 2004

Whenever variable-length entropy codes are used in the presence of a noisy channel, any channel errors will propagate and cause significant harm. Despite using channel codes, some residual errors always remain, whose effect will get magnified by error propagation. Mitigating this undesirable effect is of great practical interest. One approach is to use the residual redundancy of variable-length codes for joint source-channel decoding. In this paper, we improve the performance of residual redundancy source-channel decoding via an iterative list decoder made possible by a nonbinary outer CRC code. We show that the list decoding of VLCs is beneficial for entropy codes that contain redundancy. Such codes are used in state-of-the-art video coders, for example. The proposed list decoder improves the overall performance significantly in AWGN and fully interleaved Rayleigh fading channels.

Keywords and phrases: joint source-channel coding, variable-length codes, list decoding, iterative decoding.

1. INTRODUCTION

Variable-length codes (VLCs) for entropy coding are by now a central part of most data compression techniques, which are in turn essential for many communications applications, including text, voice, images, and video. While VLCs achieve significant compression, they also introduce dependencies in the data structure through their variable length, thus leading to error propagation in the decoded sequence.

One of the techniques that has been used to combat this undesirable effect is joint source-channel decoding. It is known that even the most efficient symbol-by-symbol compression (Huffman code) does not always achieve the entropy limit, therefore redundancy often remains in compressed data. This redundancy can, in principle, be used to assist the decoder.

Taking this argument one step further, it has been proposed to leave redundancy intentionally in entropy codes, for the purposes of resilience against channel noise. For example, the video coding standard H.263+ and its descendants use a reversible variable-length code (RVLC) [1] whose compression efficiency is less than Huffman codes. However, the RVLC allows bidirectional symbol-based decoding which is useful in the presence of channel errors. This approach

has been generalized by designing entropy codes with pre-specified minimum distance [2, 3].

The error resilience of entropy codes can be used to “clean up” any residual errors from the traditional error control coding (see Figure 1). For example, in the case of RVLC, one may start decoding from the end of the sequence whenever an error is observed. This is a separable approach to decoding. However, we know today that serially concatenated codes offer significantly improved performance if the decoding operation is done *jointly*, via the soft-input soft-output (SISO) decoding algorithm. This principle has been applied to finite-alphabet source-channel codes by Bauer and Hagenauer [4, 5], and further analyzed in [6, 7].

In this paper, we propose an improvement over the method of Bauer and Hagenauer by introducing a list decoder for source-channel decoding, made possible by a nonbinary CRC outer code. We implement this list decoder via an iterative decoding procedure similar to serial concatenated codes (Figure 2).

We briefly summarize and review the issues of iterative source-channel decoding in Section 2. We introduce list decoding of the concatenated code in Section 3. We present some analytical and experimental results in Section 4 and offer concluding remarks in Section 5.

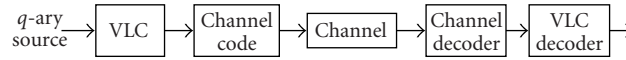


FIGURE 1: Conventional concatenated source-channel decoder.

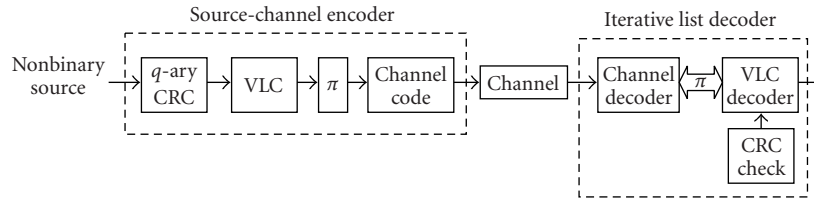


FIGURE 2: Proposed list iterative joint source-channel coding system.

2. SERIAL CONCATENATION OF VLC AND CHANNEL CODES

For the clarity of exposition, we first consider the system of Figure 2 in the absence of the CRC and list decoding component. The simplified system consists of an outer (VLC) code and an inner channel code, separated by an interleaver π . The source and channel codes are jointly (iteratively) decoded at the receiver. As mentioned previously, this method relies on residual redundancy in the VLC, in particular, sometimes redundancy is retained in the VLC on purpose, for example, in RVLCs. Thus, for the purposes of this section, we treat both codes in terms of their distance properties.

We treat the outer code, \mathcal{C}^o , as a channel code. The key difficulty of the analysis, which requires a generalization of the well-known work of [8], is that VLCs are nonlinear.

The preceding analysis closely follows that of [7]. Assume a sequence of K symbols is encoded, and the average length of the outer entropy code symbols is ℓ_{ave} . Hence, the output bit sequence of \mathcal{C}^o has a variable length with size N_{\min} to N_{\max} . Code \mathcal{C}^o is partitioned in a way such that all codewords of \mathcal{C}^o with length $N \in [N_{\min}, N_{\max}]$ form a subcode denoted by \mathcal{C}_N . In other words, to avoid dealing with variable lengths, we partition the set of all composite codewords into sets of equal length [2]. We define the free distance of \mathcal{C}^o , d_f^o as the minimum of the free distances of \mathcal{C}_N 's.

The number of inner codewords with output weight h and input weight ℓ is shown by $A_{\ell,h}^i$. Assume the outer subcode \mathcal{C}_N has $A_{\ell}^o(N)$ pairs of codewords with Hamming distance ℓ . Using the uniform interleaver notion of [8], and thanks to linearity of the inner code, the number of pairs of codewords of the overall concatenated code, with Hamming distance h , is

$$A_h(N) = \sum_{\ell=d_f^o}^N \frac{A_{\ell}^o(N)A_{\ell,h}^i(N)}{\binom{N}{\ell}}. \quad (1)$$

The pairwise error probability (PEP) of a pair of codewords with Hamming distance h is $P_h = Q(\sqrt{2hE_s/N_0})$. Using (1),

the PEP of the concatenated code is

$$\begin{aligned} P_E &\leq \sum_{N=N_{\min}}^{N_{\max}} \Pr(N) \sum_{h=d_f}^{N/R^i} A_h(N)P_h \\ &= \sum_{N=N_{\min}}^{N_{\max}} \sum_{h=d_f}^{N/R^i} \sum_{\ell \geq d_f^o} \Pr(N) \frac{A_{\ell}^o(N)A_{\ell,h}^i(N)}{\binom{N}{\ell}} Q\left(\sqrt{2h\frac{E_s}{N_0}}\right), \end{aligned} \quad (2)$$

where d_f is the free distance of the concatenated code, R^i is the rate of the inner channel code, and $\Pr(N)$ is the probability of the codewords of \mathcal{C}_N . We note that the above union bound can be used with different choices of inner and outer codes, for example, a convolutional or turbo code as inner code [4, 9, 10], or Huffman code or RVLC as outer code. A similar development is possible for symbol error rate [7], which we do not present here for the sake of brevity.

Iterative decoding of the concatenated source-channel code is performed via soft-input soft-output (SISO) modules for the inner and outer codes. For the outer code, the SISO module is performed over a bit-level trellis representation of VLC, similar to the one originally proposed by Balakirsky [11].

3. LIST DECODING OF SERIALY CONCATENATED VLC AND CHANNEL CODES

A list decoder provides an ordered list of the L most probable sequences in maximum-likelihood sense. Then, an outer error detecting code, usually a cyclic redundancy check (CRC) code, verifies the validity of the candidates and selects the error-free sequence, if exists, among the candidates. Two variations of the list Viterbi algorithm (LVA) are reported in [12].

An ordinary ML (Viterbi) decoder makes an error whenever the codeword closest to the received waveform is an erroneous codeword. For the list decoder to make an error, the correct sequence must lie outside of the L nearest neighbors

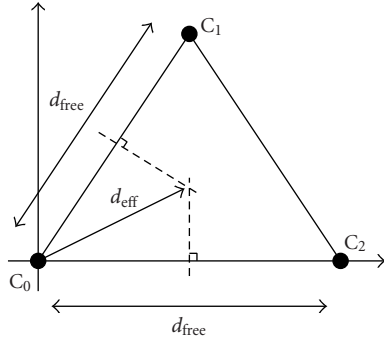


FIGURE 3: Asymptotic analysis of list Viterbi algorithm.

of the received sequence. This error is less probable than the corresponding error in the ML decoder.

In a list decoder, the distance between the received sequence and all the candidates determines the performance. Therefore, determining the exact performance is mathematically intractable. But it is possible to calculate the asymptotic coding gain, for example, see [12]. In the case of AWGN channel, a geometrical argument reveals that the asymptotic coding gain is $G = 10 \log(2L/(L+1))$ dB for a list of length L . However, the actual gain is often less due to the multiplicity of the set of L nearest neighbors, which is neglected in the analysis [12].

3.1. List decoding of variable-length codes

List decoders can also be applied for variable-length encoded sequences, given an appropriate trellis (e.g., the bit-level trellises mentioned earlier). Our list decoding is constructed with the help of a non binary CRC code, which verifies the validity of the L most probable paths in the VLC trellis. The alphabet set of the CRC code must cover all codewords of the VLC (size q). If q is a power of a prime, it is possible to construct a q -ary CRC code, otherwise the size of VLC should be extended to the nearest power of a prime. One can use the a priori knowledge that these additional symbols are never present in the data sequence, but only (possibly) present in the parity sequence.

The asymptotic error rate for a list of size $L = 2$ is based on a simple geometric construction due to Seshadri and Sundberg [12] (see Figure 3). When the three codewords are pairwise equidistant, it produces a worst-case error probability. In this case, the minimum-magnitude noise resulting in an error is shown by the vector terminating at the circumcenter of the triangle. This vector represents the effective minimum distance, denoted by d_{eff} , which is larger than $d_{\text{free}}/2$, explaining the list decoding gain, which is equal to $10 \log(2L/(L+1))$ dB, as calculated in [12].

This value of asymptotic gain, however, ignores the multiplicities of the minimum distance, and in our case minimum-distance error event has high multiplicities.¹

¹More information on the distance spectrum of VLCs is available in [2], and two examples are given in [4].

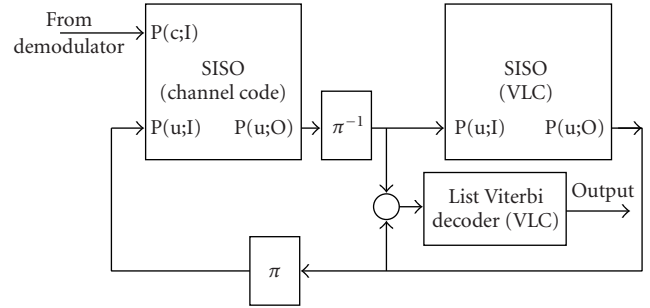


FIGURE 4: Iterative list decoding of VLC and channel code.

Therefore, we augment the asymptotic analysis of [12, 13] for $L = 2, 3$ list decoder of VLCs so that multiplicities are taken into account. We denote by N_{free} the multiplicity of the minimum-distance errors.² The number of codeword triplets at minimum-distance that include the transmitted codeword is $N_{\text{eff}} = N_{\text{free}}(N_{\text{free}} - 1)/2$. Thus, for $L = 2$ and assuming an AWGN channel, coding gain is the difference $\Delta\gamma = \gamma_1 - \gamma_2$, where γ_1 and γ_2 are the two values of E_b/N_0 such that

$$N_{\text{eff}} Q(\sqrt{2d_{\text{eff}}\gamma_2}) = N_{\text{free}} Q(\sqrt{2d_{\text{free}}\gamma_1}). \quad (3)$$

Simulations show that the coding gain thus obtained is more accurate than results that ignore multiplicities, for example, [12, 13] (see Section 5). The disadvantage is that the equation above does not admit a closed-form solution.

Similarly, worst-case analysis can be repeated for $L = 3$ list decoder to calculate d_{eff} . To obtain a more realistic approximation of the coding gain, we consider the multiplicity of the worst case of the set of three codewords, which is $N_{\text{eff}} = N_{\text{free}}(N_{\text{free}} - 1)(N_{\text{free}} - 2)/6$, given $N_{\text{free}} \geq 3$. The coding gain is calculated in a similar way as $L = 2$.

3.2. Proposed iterative list decoder

We now introduce an approximated list decoder for the concatenation of VLCs and channel codes. Our proposed iterative list decoder is demonstrated in Figure 4. After the last iteration, the final soft-output sequence produced by the SISO is decoded by the list Viterbi algorithm. The trellis used in this final decoder is similar to the one used in SISO-VLC.

The asymptotic analysis of the list decoder of turbo codes in [13] shows that the coding gain of list turbo decoder is higher than the coding gain of convolutional list decoder. Specifically, due to the low probability of multiple free-distance error events in a turbo-encoded sequence, the asymptotic coding gain is determined by the second minimum distance, yielding higher gain [13]. For the case of serially concatenated VLCs and convolutional codes, we show experimentally in Section 4 that significant improvements in coding performance can be achieved.

²The multiplicities of VLCs, in general, are not integer-valued since we must average the multiplicities of the subcodes. In our analysis, we round the multiplicities up to simplify the calculation.

TABLE 1: Variable-length codes used in Section 4.

s	$P_S(s)$	C_1	C_2 [4]	C_3
0	0.33	00	00	11
1	0.30	11	11	001
2	0.18	10	010	0100
3	0.10	010	101	0101100
4	0.09	011	0110	0001010
$E[L]$	$H = 2.14$	2.19	2.46	3.61
d_{free}	—	1	2	3

3.3. Nonbinary CRC

Wicker [14] provides a comprehensive background on Galois fields, rings of polynomials on Galois fields, and the construction of cyclic codes. We give here a quick summary of the key results as well as the procedure for designing nonbinary CRCs.

Cyclic codes are built using a generator polynomial on the underlying Galois field $\text{GF}(q)$. If the number of symbols in our application is not a power of a prime, the next higher appropriate q must be chosen, since for a field $\text{GF}(q)$, q must be either a prime or a power of a prime. Cyclic codes are built from a generator polynomial $g(X)$ on $\text{GF}(q)$. The codewords are all the multiples of $g(X)$ modulo $X^n - 1$, where $g(X)$ is a degree- r polynomial that divides $X^n - 1$.

CRC codes are shortened cyclic codes that can encode up to $n - r$ information symbols. CRC codes have excellent error detection capability. The CRC code with a generator of degree r detects all burst errors of length r or less, and the probability that the CRC will not detect a random error is q^{-r} . Due to the lack of a convenient way to calculate the error spectrum of a CRC code, ad hoc methods have been used for code design in the binary case.

Unfortunately the existing ad hoc techniques for binary CRC design are not particularly helpful for the q -ary case, but nevertheless, the general structural properties, error coverage, and burst error detection properties remain the same across different underlying Galois fields. Therefore, even though we cannot design CRC with specified minimum distance, still it is possible to arrive at codes that have very respectable error detection performance. For example, for the 5-ary code used in the next section, a possible choice for generator polynomial is the primitive polynomial $X^8 + 4X^6 + X^4 + X^3 + X^2 + 3X + 3$ which requires 8 parity symbols for data sequences up to 390617 symbols. The undetected codeword error probability for this code is only 2.56×10^{-6} .

4. EXPERIMENTAL RESULTS

Table 1 shows the 5-ary source used in our experiments and various codes designed for this source. C_1 is a Huffman code, C_2 is an RVLC for this source reported in [4], C_3 is a high-redundancy code designed by us because we observed that

TABLE 2: Convolutional codes used in Section 4 (from [8]).

CC_1 : rate $\frac{1}{2}$	$\left(1, \frac{1+D^2}{1+D+D^2}\right)$
CC_2 : rate $\frac{1}{2}$	$\left(1, \frac{1+D+D^3}{1+D}\right)$
CC_3 : rate $\frac{2}{3}$	$\begin{pmatrix} 1 & 0 & \frac{1+D^2}{1+D+D^2} \\ 0 & 1 & \frac{1+D}{1+D+D^2} \end{pmatrix}$

the free distance of the outer code is a crucial factor in performance, as seen by the asymptotic behavior of the multiplicities A_h in (1). It is noteworthy that despite the differences, the trellises of the different codes have roughly the same order of complexity, due to sparseness of the VLC trellises.

Table 2 shows the recursive convolutional codes employed as inner code in our schemes. In our experiments, a packet of K symbols is entropy-encoded, interleaved, channel encoded, and transmitted using binary phase-shift keying (BPSK) modulation over an AWGN channel or fully interleaved Rayleigh fading channel.

4.1. Iterative decoding

Figure 5a shows union bounds³ and simulation results for the concatenated code $C_2 + CC_1$. The calculation of the multiplicities for a nonlinear, variable-length code is a lengthy and time-consuming process, thus we present *truncated bounds* calculated with the first 10 terms of the multiplicities of the outer code that are available in [4]. The decoding experiment was performed with 10 iterations, with packet lengths of 20 and 200.

We consider two outer VLCs: code C_2 with free distance 2 and code C_3 with free distance 3, to build codes $C_2 + CC_1$ and $C_3 + CC_3$ with overall rates 0.445 and 0.404, respectively.⁴ The symbol error rate (SER) of the two concatenated codes is shown in Figure 5b for $K = 2000$ symbols. In a wide range of E_b/N_0 , the code $C_3 + CC_3$ outperforms $C_2 + CC_1$ and demonstrates a sharper drop in error rate. Other simulations have shown that in terms of frame error rate (FER), $C_3 + CC_3$ provides significant coding gain, about 1.4 dB at $\text{FER} = 10^{-3}$.

For $C_2 + CC_1$, we noticed that the higher number of iterations does not provide much of coding gain. We use the density evolution technique to give insight into the progress of iterative decoder. After an experimental verification that the LLR histograms are indeed Gaussian, we evaluated the approximate density evolution for $C_2 + CC_1$ and $C_3 + CC_3$ (Figure 6). The two lower curves in each plot correspond to

³Union bounds work in the high E_b/N_0 regions, and they are calculated for the optimal (ML) decoder, and iterative decoding is not optimal. This explains the deviations of simulation from union bounds.

⁴The equivalent code rate of a VLC is defined as the average length of the VLC divided by the average length of the Huffman code.

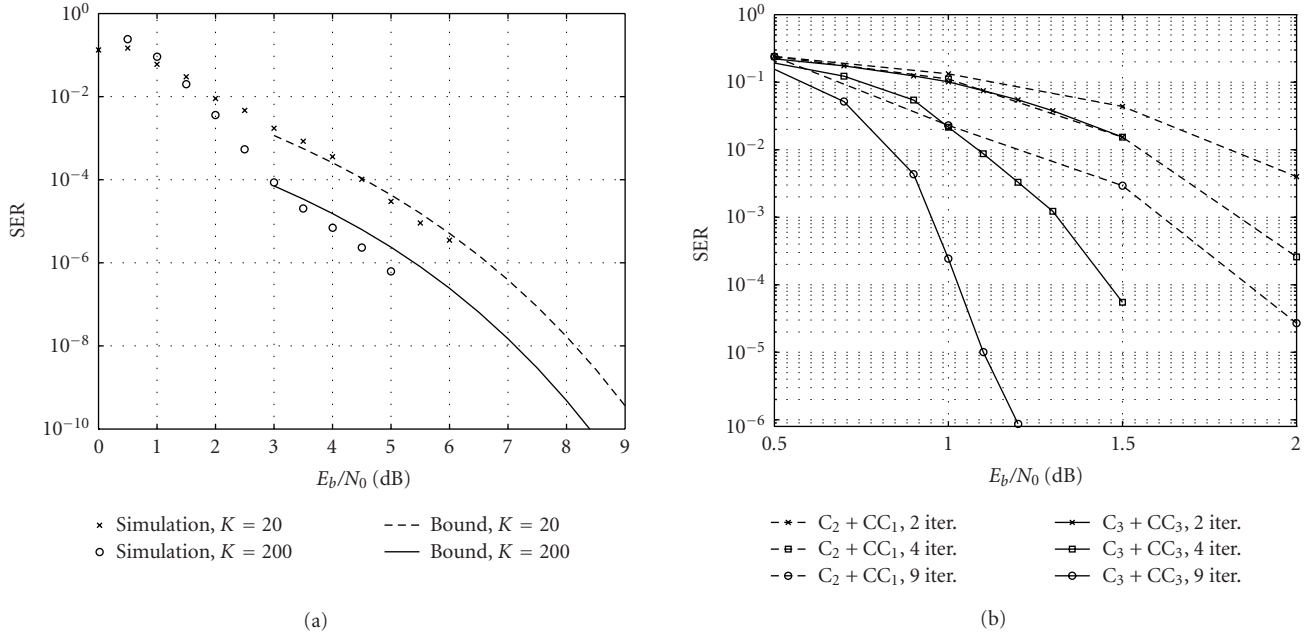


FIGURE 5: (a) Performance and union bounds of $C_2 + CC_1$, $K = 20$ and 200 symbols; (b) performance of $C_2 + CC_1$, and $C_3 + CC_3$, $K = 2000$.

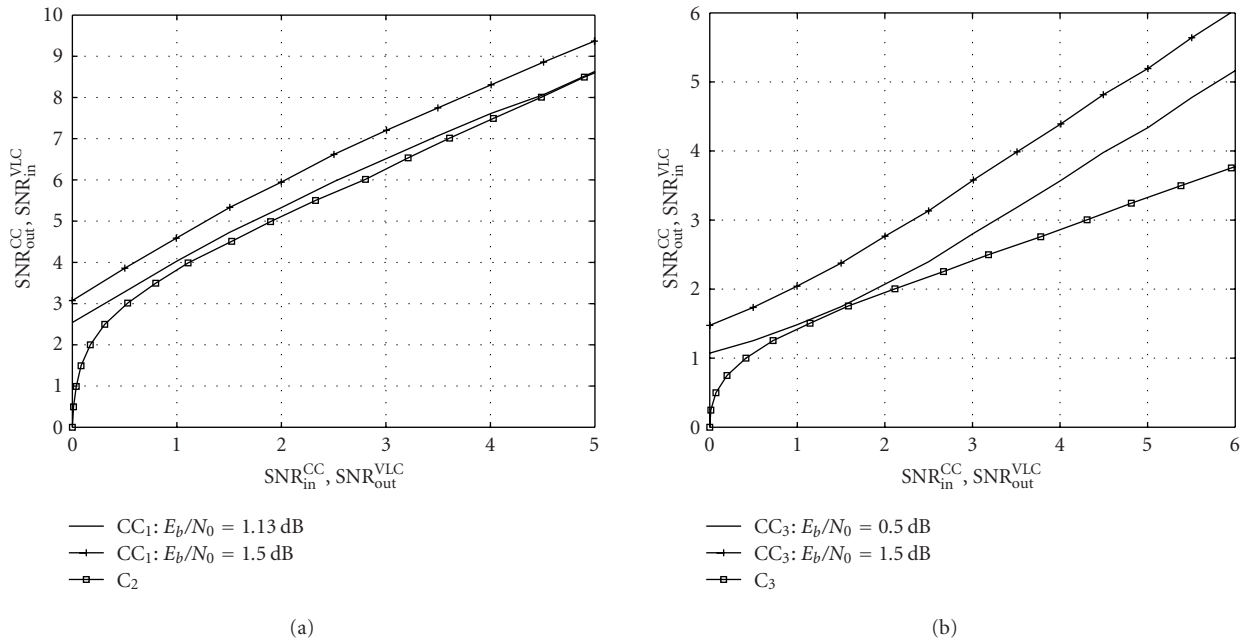


FIGURE 6: Approximate Gaussian density evolution of $C_2 + CC_1$ and $C_3 + CC_3$, $K = 2000$.

the iterative decoding threshold [15]. The code $C_3 + CC_3$ has lower threshold than $C_2 + CC_1$ (0.5 dB compared to 1.15 dB).

Borrowing the notion of iterative decoder tunnel from [15], we observe that the wider tunnel of $C_3 + CC_3$ provides a fast convergence with a few iterations: the higher the channel E_b/N_0 , the fewer the iterations needed for convergence. These observations are in agreement with Figure 5b.

4.2. Iterative list decoding

We first evaluated the accuracy of our analysis for the performance of list decoding, which takes multiplicities into account. We used code C_2 , with $K = 200$ symbols in the AWGN channel. The coding gain at FER = 10^{-4} is calculated as 1 dB for $L = 2$ and 1.4 dB for $L = 3$. These values are a better

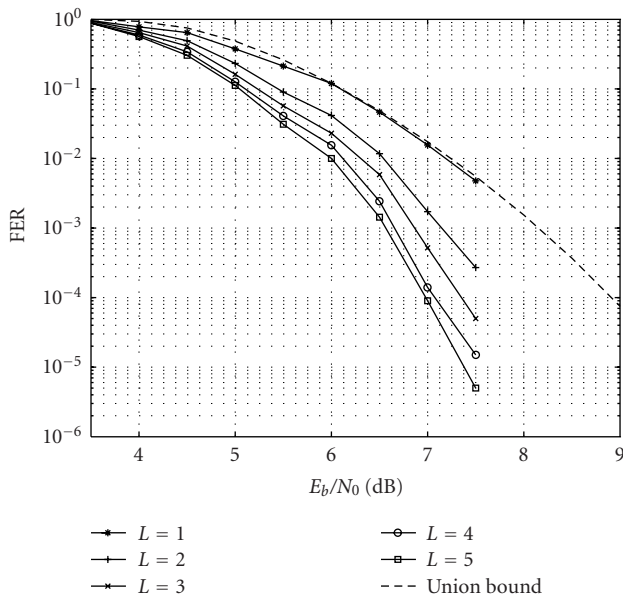


FIGURE 7: List decoding of C_2 in AWGN channel, $K=200$.

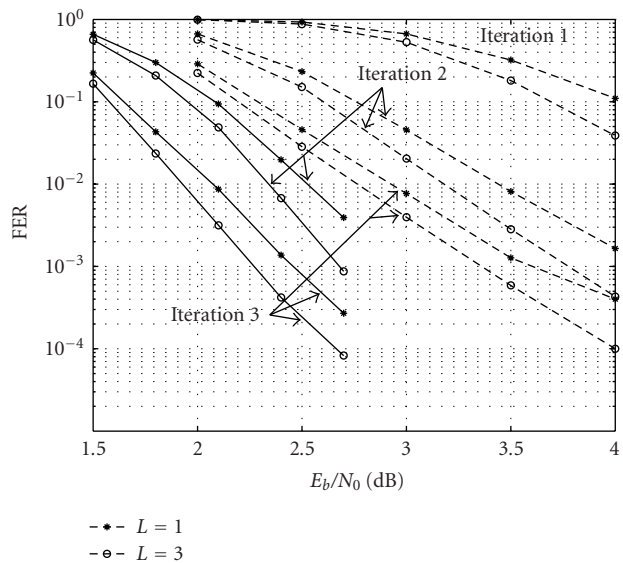


FIGURE 8: Iterative list decoding of $C_2 + CC_1$ (dashed) and $C_3 + CC_3$ (solid line) in AWGN channel, $K=500$.

match to simulations (Figure 7) than the coding gain predicted by [12].

Consider the two codes $C_2 + CC_1$ and $C_3 + CC_3$. Figure 8 presents the FER of the iterative list decoder at the first, second, and third iterations with $L = 1, 3$ in AWGN channel with $K = 500$. $C_3 + CC_3$ outperforms $C_2 + CC_1$. Figure 9 reports the FER of the concatenated codes in a fully interleaved Rayleigh channel with $K = 200$. At this frame size, the difference between the two concatenated codes is less pronounced, but still $C_3 + CC_3$ has lower error rate (except in the first iteration). List decoding has higher coding gain under fully interleaved Rayleigh channel, because of added diversity arising from increased equivalent free distance of the code [12].

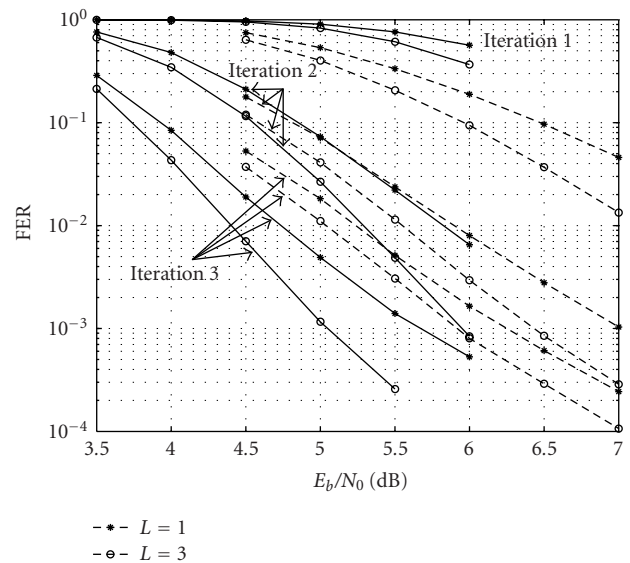


FIGURE 9: Iterative list decoding of $C_2 + CC_1$ (dashed) and $C_3 + CC_3$ (solid line) in fully interleaved Rayleigh channel, $K=200$.

The coding gain of $C_2 + CC_2$ at the fifth iteration for $L = 2$ is about 1.5 dB in Rayleigh fading, and 0.75 dB with $L = 5$ in AWGN channel. We refer the interested reader to [6] for further results on this code.

5. CONCLUSION

We propose an iterative list decoder for VLC-based source-channel codes. The iterative decoding of source-channel codes is made possible by the residual redundancy in the source code. Some source coders, such as H.263+, include additional redundancy for error resilience, making a source-channel decoder more desirable. It is shown that the amount of the redundancy in the VLC plays an important role in the performance of the code, given a total rate constraint. The list decoder is made possible by a nonbinary CRC code which also provides a stopping criterion for the iterative decoder. At a given iteration of the iterative decoder, the proposed list decoder improves the overall performance of the system. Extensive experimental results are provided in AWGN and fully interleaved Rayleigh channels.

ACKNOWLEDGMENTS

This work was supported in part by the NSF under Grant no. CCR-9985171. The work of A. Hedayat was also supported in part by Texas Telecommunications Engineering Consortium (TxTEC). This work was presented in part in Asilomar 2002 and in ICC 2003.

REFERENCES

[1] T. Okuda, E. Tanaka, and T. Kasai, "A method for correction of garbled words based on the Levenshtein metric," *IEEE Trans. Comput.*, vol. C 25, pp. 172–176, February 1976.

- [2] V. Buttigieg, *Variable-length error-correcting codes*, Ph.D. thesis, Department of Electrical Engineering, University of Manchester, Manchester, UK, 1995.
- [3] V. Buttigieg and P. G. Farrell, "Variable-length error-correcting codes," *IEE Proceedings-Communications*, vol. 147, no. 4, pp. 211–215, 2000.
- [4] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel decoding," in *Proc. Data Compression Conference (DCC '01)*, pp. 273–282, Snowbird, Utah, USA, March 2001.
- [5] R. Bauer and J. Hagenauer, "Iterative source/channel decoding using reversible variable length codes," in *Proc. Data Compression Conference (DCC '00)*, pp. 93–102, Snowbird, Utah, USA, March 2000.
- [6] A. Hedayat and A. Nosratinia, "List-decoding of variable-length codes with application in joint source-channel coding," in *Proc. 36th IEEE Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 21–25, Pacific Grove, Calif, USA, November 2002.
- [7] A. Hedayat and A. Nosratinia, "Concatenated error-correcting entropy codes and channel codes," in *Proc. IEEE International Conference on Communications (ICC '03)*, vol. 5, pp. 3090–3094, Anchorage, Alaska, USA, May 2003.
- [8] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 909–926, 1998.
- [9] K. Lakovic and J. Villasenor, "Combining variable length codes and turbo codes," in *Proc. 55th IEEE Vehicular Technology Conference (VTC '02)*, vol. 4, pp. 1719–1723, Birmingham, Ala, USA, May 2002.
- [10] X. Jaspard and L. Vandendorpe, "Three SISO modules joint source-channel turbo-decoding of variable length coded images," in *Proc. 5th International ITG conference on Source and Channel Coding (SCC '04)*, pp. 279–286, Erlangen, Germany, January 2004.
- [11] V. B. Balakirsky, "Joint source-channel coding with variable length codes," in *Proc. IEEE International Symposium on Information Theory (ISIT '02)*, p. 419, Ulm, Germany, Jun-July 1997.
- [12] N. Seshadri and C.-E. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 313–323, 1994.
- [13] K. R. Narayanan and G. L. Stüber, "List decoding of turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 6, pp. 754–762, 1998.
- [14] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.
- [15] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 891–907, 2001.

Aria Nosratinia received the B.S. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 1988, the M.S. degree in electrical engineering from the University of Windsor, Windsor, Ontario, Canada, in 1991, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 1996. From 1995 to 1996, he was with Princeton University, Princeton, New Jersey. From 1996 to 1999, he was a Visiting Professor and Faculty Fellow at Rice University, Houston, Texas. Since 1999, he has been with the faculty of the University of Texas, Dallas, where he is currently an Associate Professor of electrical engineering. His research interests are in the broad area of communication and information theory, particularly, coding and signal processing for the communication of multimedia signals. He was the recipient of the National Science Foundation Career award in 2000 and has twice received chapter awards for his outstanding service to the IEEE Signal Processing Society.



Ahmadreza Hedayat received the B.S.E.E. and M.S.E.E. degrees from the University of Tehran, Tehran, Iran, in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from the University of Texas at Dallas, Richardson, in 2004. From 1995 to 1999, he was with Pars Telephone Kar and Informatics Services Corporation, Tehran, Iran. Currently, he is a Senior Systems Engineer with Navini Networks, Richardson, Tex. His current research interests include MIMO signaling and techniques, channel coding, source-channel coding, and cross-layer schemes.

