Soft-Decision Priority-First Decoding Algorithms for Variable-Length Error-Correcting Codes

Yuh-Ming Huang, *Member, IEEE*, Yunghsiang S. Han, *Senior Member, IEEE*, and Ting-Yi Wu

***Abstract*—Joint source-channel decoding has recently received extensive attention due to the rise in the applications of multimedia wireless communication. Based on a code trellis rather than on a code tree, this work presents a maximum *a posteriori* (MAP) soft-decision priority-first decoding algorithm and its approximations for variable-length error-correcting codes. Simulation results indicate that for the code with average codeword length 6.269 bits and free distance 3, under moderate signal-to-noise ratio, one of the proposed algorithms almost reaches the lowest decoding complexity, and has nearly the same performance on symbol error probability as the MAP decoding.**

***Index Terms*—Variable-length error-correcting codes, MAP, soft-decision, joint source/channel coding.**

I. Introduction

A MEMORYLESS or Markov source is usually encoded by a variable-length code (VLC) or by a variable-length error-correcting code (VLEC). The encoded data can also be further protected by a channel code to enhance error-resistance. Recent researches in VLC-based joint source-channel decoding systems can be classified into three categories. The first is to use the source statistics for acting as one source and channel decoder at the source-decoding stage. Based on a modified form of Viterbi algorithm, Bystrom *et al.* [1] adopted symbol-level trellis structure for MAP decoding on VLECs. In order to reduce the decoding complexity, low-complexity but sub-optimal tree-based sequential decoding algorithms were proposed in [1] [2]. The decodings in the second category were performed at the channel-decoding stage if a channel code is involved. Also based on a generalized Viterbi algorithm, Murad and Fuja [3] presented an exact MAP estimation by a bit-level super-trellis which is capable of exploiting all possible sources and side information. Each state of this trellis can be regarded as a four-dimensional vector. However, this scheme has a serious drawback that, for long source sequences, the decoding complexity becomes quite expensive due to the enormous numbers of trellis states. An approximate solution was proposed in [4] using a simplified trellis with two-dimensional state vectors. Rather than constructing a single decoding unit as those in the first/second category, in the third category, the overall decoding performance was further improved by using the turbo principle between source and channel decoders; however, the decoding complexity of such approach becomes much higher. A bit-level-trellis-based iterative decoding approach [5] was adopted in [6] for decoding VLECs and it was extended by Thobaben and Kliewer [7] for decoding VLC-encoded Markov source.

In this letter, we presents a trellis-based MAP soft-decision priority-first decoding algorithm (MAPPDA) for VLECs, along with its approximations, which have a distinguishing feature that the computational complexity of the new derived branch metric is very simple when compared with the Fano metric adopted in [1] and [2]. Our proposed methods belong to the first category and they can be applied to the cases with more knowledge of the source or side information, and even can be extended to some systems in the other categories. Simulation results indicate that the approximations perform almost identically to the MAP algorithm, but with a markedly lower computational complexity while compared with other Viterbi-based algorithms.

Manuscript received April 9, 2008. The associate editor coordinating the review of this letter and approving it for publication was G. Mazzini. This work was supported by the National Science Council, Taiwan, R.O.C., under the project No. NSC 95-2221-E-260-029-MY2.

Y. M. Huang and T. Y. Wu are with the Department of Computer Science and Information Engineering, National Chi Nan University, Puli, Taiwan, R.O.C. (e-mail: ymhuang@csie.ncnu.edu.tw).

Y. S. Han is with the Graduate Institute of Communication Engineering, National Taipei University, Taipei County, Taiwan, R.O.C.

Digital Object Identifier 10.1109/LCOMM.2008.080561.

II. MAP Soft-Decision Priority-First Decoding

Let $\boldsymbol{C}$ denote a binary variable-length code (VLC) that has K codewords $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K\}$ with corresponding probabilities $\{p_1, p_2, \ldots, p_K\}$. The respective lengths of the codewords are given by $\{\ell_1, \ell_2, \ldots, \ell_K\}$. A bitstream, $\boldsymbol{X}$, is a concatenation of L codewords from $\boldsymbol{C}$. Let $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L)$, the length in bits of the k-th codeword be $||\boldsymbol{x}_k||$, and let $N = \sum_{k=1}^{L} ||\boldsymbol{x}_k||$ be the length of the bitstream in bits. In this work, the bitstream is binary-phase-shift-keying (BPSK) modulated and then sent over an memoryless and additive white Gaussian noise (AWGN) channel without channel coding protection, and the received vector in bits $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$ denotes the set of the transmitted bits corrupted by the noise. We also assume that no bits are deleted by the channel, and that the length of bitstream N is known at the receiver.

The transmitted bitstream can be represented as a path in a symbol-level trellis, where each state denotes a position in the bitstream following a valid codeword [1]. Since the number of codewords is equal to K, there are at most K branches extended from each state. The codeword, that presents a symbol, labels each branch or transition in the trellis. Since N is known, states near the final stage have only those branches corresponding to codewords of appropriate lengths to terminate in state N. Figure 1 shows a simple example of trellis of a bitstream of length N. Once a trellis is constructed, an MAP decoding can be applied to it whenever the MAP metric of each branch is specified.

Term E_N denotes the collection of all bitstreams of length N formed by sequences of codewords in $\boldsymbol{C}$. For MAP

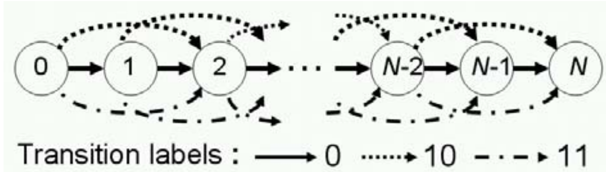

Fig. 1. Trellis diagram for a code with three codewords.

decoding, a bitstream $\hat{\boldsymbol{v}} \in E_N$ needs to be selected such that

$$\Pr(\hat{\boldsymbol{v}}|\boldsymbol{r}) \geq \Pr(\boldsymbol{v}|\boldsymbol{r}) \tag{1}$$

for all $\boldsymbol{v} \in E_N$. The hard-decision sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$ corresponding to the received vector in bits $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$ is defined as

$$y_j \triangleq \begin{cases} 1, & \text{if } \phi_j < 0; \\ 0, & \text{otherwise,} \end{cases}$$

where

$$\phi_j \triangleq \ln \frac{\Pr(r_j|0)}{\Pr(r_j|1)}. \tag{2}$$

Due to the memoryless channel assumption, the condition in (1) can be rewritten as

$$\begin{aligned} & \Pr(\boldsymbol{r}|\hat{\boldsymbol{v}}) \Pr(\hat{\boldsymbol{v}}) \geq \Pr(\boldsymbol{r}|\boldsymbol{v}) \Pr(\boldsymbol{v}) \\ \Leftrightarrow \quad & \sum_{j=1}^{N} (-1)^{\hat{v}_j} \phi_j + 2 \ln \Pr(\hat{\boldsymbol{v}}) \geq \sum_{j=1}^{N} (-1)^{v_j} \phi_j \\ & + 2 \ln \Pr(\boldsymbol{v}) \qquad (3) \\ \Leftrightarrow \quad & \sum_{j=1}^{N} (y_j \oplus \hat{v}_j)|\phi_j| - \ln \Pr(\hat{\boldsymbol{v}}) \leq \sum_{j=1}^{N} (y_j \oplus v_j)|\phi_j| \\ & - \ln \Pr(\boldsymbol{v}). \qquad (4) \end{aligned}$$

Equation (3) is the commonly adopted path metric for an MAP decoder such as the modified Viterbi decoder proposed by Bystrom*et al.* [1]; however, the proposed priority-first soft-decision decoding algorithm adopts (4). Notably, (4) can be regarded as the generalized $correlation\ discrepancy$ [8] for the VLC case.

Next, under the assumption of independent source symbol, a new branch metric for each branch can be defined in a trellis as follows. For branch labeled by $\boldsymbol{x}_s^i = \left(x_s^{(i_1)}, x_s^{(i_2)}, \ldots, x_s^{(i_{\ell_i})}\right)$ between state $s - \ell_i$ and state s in a trellis, the branch *metric* associated with it is defined as

$$M\left(\boldsymbol{x}_s^i\right) \triangleq \sum_{j=1}^{\ell_i} (y_{s-\ell_i+j} \oplus x_s^{(i_j)})|\phi_{s-\ell_i+j}| - \ln \Pr(\boldsymbol{x}_s^i). \tag{5}$$

The path metric is then a summation on the metrics of all branches on the path. That is, for a path, $\boldsymbol{X}_{s_T} = \left(\boldsymbol{x}_{s_1}^{i_1}, \boldsymbol{x}_{s_2}^{i_2}, \ldots, \boldsymbol{x}_{s_T}^{i_T}\right)$, ending at state s_T and having T branches, the metric of the path is given by

$$M(\boldsymbol{X}_{s_T}) = \sum_{k=1}^{T} M(\boldsymbol{x}_{s_k}^{i_k}). \tag{6}$$

Equation (5) (along with (6)) can be treated as the VLC version of the MAP metric given in [9].

Han [10] shows that an estimation on the path metric of the path starting from state s_T to the state N, the final state, can be added to (6) to perform an MAP decoding whenever the estimation is no more than the best path metric from state s_T to state N among all possible paths between them. Moreover, increasing the estimate speeds up the searching. Thus, such estimation is added to (6) as follows.

Without loss of generality, assume that p_1 denotes the largest probability of all codewords in $\boldsymbol{C}$, and that ℓ_s denotes the longest length of them. A new path metric can be defined as follows, while still performing exact MAP decoding:

$$M'(\boldsymbol{X}_{s_T}) = \sum_{k=1}^{T} M(\boldsymbol{x}_{s_k}^{i_k}) + \left(-\left\lfloor \frac{N - s_T}{\ell_s} \right\rfloor \ln p_1\right). \tag{7}$$

The above equation clearly indicates that the branch metric has a lower bound of $M(\boldsymbol{x}_s^i) \geq -\ln \Pr(\boldsymbol{x}_s^i) \geq -\ln p_1$. Moreover, the number of valid codewords in a path starting from state S_T to state N is at least $\left\lfloor \frac{N-s_T}{\ell_s} \right\rfloor$. Thus, $-\left\lfloor \frac{N-s_T}{\ell_s} \right\rfloor \ln p_1$ is no more than the best path metric among all paths starting from state S_T to state N.

Owing to the non-negativity of the branch metric $M(\boldsymbol{x}_{s_k}^{i_k})$ and the estimation $-\left\lfloor \frac{N-s_T}{\ell_s} \right\rfloor \ln p_1$ in (7), the path metric in (7) is non-decreasing along any path in a trellis. Accordingly, a new soft-decision priority-first-type MAP decoder (MAPPDA) can then be established. The MAPPDA can be made to operate on a trellis, instead of on a code tree, as with the traditional stack algorithm. This is achieved by introducing a *state table* to record all states in the trellis that has been expanded, i.e., whose children have been visited.

⟨The trellis-based MAPPDA for VLC ⟩

Step 1. The Stack initially has only one original state (zero state) whose metric is assigned zero.

Step 2. If the top path in the Stack ends at the terminal state (state N) in the trellis, then the algorithm stops.

Step 3. Delete the top path from the Stack, and put the ending state of this top path into the state table. Calculate the path metrics of the successors of the top path in the Stack.

Step 4. If any of the new paths ends at a state already in the state table, then discard the new path.

Step 5. If any of the new paths merges with a path already in the Stack, then remove the one with higher path metric value.

Step 6. Insert the remaining new paths into the stack and reorder the Stack according to ascending metric values. Go to Step 2.

As indicated in the previous algorithm, the *Stack* contains all paths explored by the MAPPDA, which are not the prefix of all other paths in the *Stack*. The *Stack* appears to function similarly to the stack in the traditional sequential decoding algorithm. The *state table* records the information of the ending states of the paths that had previously been the top paths of the *Stack*. The optimality of the MAPPDA can be proved by a similar argument given in [9].

In order to further lower the decoding complexity of MAPPDA, the effect of the source priori probability can be disregarded with almost no performance degradation, that is the term -lnPr($\boldsymbol{x}_s^i$) in (5) and the estimation in (7) were not included in both equations. This approximate solution is denoted

TABLE I

AVERAGE (AVE) AND MAXIMUM (MAX) NUMBER OF BRANCH METRIC COMPUTATIONS

SNR_b	1 dB		2 dB		3 dB		4 dB		5 dB		6 dB		7 dB		8 dB	
method	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
VA [1]	325754	332028	325754	332028	325754	332028	325754	332028	325754	332028	325754	332028	325754	332028	325754	332028
SA [2]	1745236	3119519	174797	1022914	65591	299155	55178	60863	52835	56099	52185	53775	52019	52269	51984	52108
MAPPDA20	261606	265241	224994	234686	171405	182678	114930	122284	76512	81757	59072	61715	53494	54224	52210	52532
AMAPPDA30	123750	129187	93783	103933	70486	77157	58154	61261	53574	55460	52306	53049	52031	52262	51984	52093

as AMAPPDA. Moreover, one heuristic policy was adopted on MAPPDA and AMAPPDA to ensure that the ending state of the next searched path was within a threshold to the ending state of the farthest visited path. The resultant algorithms are respectively denoted as MAPPDA@ and AMAPPDA@ depending on the selected threshold value @. Even though this policy slightly degrades the error rate performance of the proposed algorithms, it reduces the decoding complexity drastically. The larger threshold value we select, the less degradation in performance the algorithms will have.

III. SIMULATION RESULTS OVER THE AWGN CHANNEL

The VLEC given in Table III of [2] with average codeword length in bits and free distance equal to 6.269 and 3, respectively, for 26-symbol English alphabet was considered in our simulation. For the VLEC, the code redundancy with respect to the Huffman code was considered. Hence, the following discussions adopt the SNR per information bit, i.e., SNR_b = SNR/R, where $R = 4.15572/6.269$ and SNR denotes the signal-to-noise ratio per channel bit.

As revealed in the algorithm, the computational efforts of the MAPPDA are determined not only by the numbers of branch metrics evaluated, but also by the cost of searching and reordering the stack elements. However, a balanced-tree data structure [9] can be adopted in the stack implementation, to ensure that the latter cost becomes of comparable order to the former one. Therefore, the branch metric computation of the MAPPDA should be considered as the key determinant of algorithmic complexity. The empirical investigation of the average decoding complexity and the performance in terms of symbol error rate (SER), which is equal to the Levenshtein distance divided by the number of symbols in the transmitted message, is now considered.

Table I compares the average computational efforts of the proposed MAPPDA20 and AMAPPDA30 with those of the modified Viterbi [1] and stack [2] algorithms, which were respectively abbreviated as VA and SA. When $SNR_b \geq 6$ dB, the average number of branch metric values evaluated by SA or AMAPPDA30 is reduced to about 52000, which is near the smallest possible number of branch metric values evaluated by any optimal decoding algorithm. However, the decoding complexity of SA becomes tremendously high at lower SNR_bs.

In Fig. 2, at all SNR_b levels, both MAPPDA20 and AMAPPDA30 all have better performance than the SA and achieve nearly the same performance on SER as the optimal decoders, VA and MAPPDA. In particular, the maximum stack size required of the MAPPDA20 or AMAPPDA30 is far smaller than that of the SA.

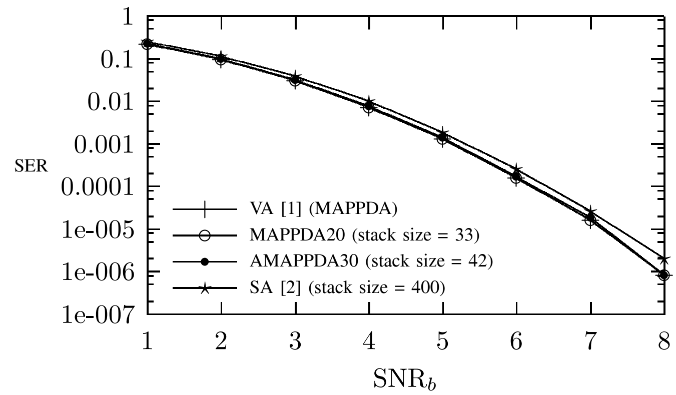

Fig. 2. Symbol error rate (SER) for 2000 randomly generated source symbols.

IV. CONCLUSIONS

The distinguishing feature of the new derived metric is that the branch metric computation is very simple when compared with the Fano metric adopted in the stack algorithm. Moreover, due to the fact that the metric value of the correct path will be accumulated far slowly than that of the uncorrect path, a fast and exact MAP decoding algorithm can be obtained.

REFERENCES

[1] M. Bystrom, S. Kaiser, and A. Kopansky, "Soft source decoding with applications," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 10, pp. 1108–1120, Oct. 2001.

[2] V. Buttigieg and R. Deguara, "Using variable-length error-correcting codes in MPEG-4 video," in *Proc. Int. Symp. on Inform. Theory*, Sept. 2005, pp. 2379–2383.

[3] A. H. Murad and T. E. Fuja, "Robust transmission of variable-length encoded sources," in *Proc. Wireless Commun. Netw. Conf.*, Sept. 1999, pp. 968–972.

[4] M. Jeanne, J. C. Carlach, and P. Siohan, "Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 10–15, Jan. 2005.

[5] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel decoding," in *Proc. Data Compression Conf.*, 2001, pp. 273–282.

[6] K. Lakovic and J. Villasenor, "On design of error-correcting reversible variable length codes," *IEEE Commun. Lett.*, vol. 6, no. 8, pp. 337–339, Jan. 2002.

[7] R. Thobaben and J. Kliewer, "Low-complexity iterative joint source-channel decoding for variable-length encoded markov sources," *IEEE Trans. Commun.*, vol. 53, pp. 2054–2064, Dec. 2005.

[8] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 2004.

[9] Y. S. Han, P.-N. Chen, and H.-B. Wu, "A maximum-likelihood soft-decision sequential decoding algorithm for binary convolutional codes," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 173–178, Feb. 2002.

[10] Y. S. Han, "A new treatment of priority-first search maximum-likelihood soft-decision decoding of linear block codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 3091–3096, Nov. 1998.