A New Soft Variable Length Decoder for Wireless Video Transmission

Tsu-Ming Liu, *Student Member, IEEE*, Sheng-Zen Wang, Bai-Jue Shieh, and Chen-Yi Lee, *Member, IEEE*

***Abstract*—In this paper, we propose an adaptive and scalable soft variable length code (VLC) decoder to greatly reduce overall design complexity. Generally, a soft VLC decoder needs to maintain many states for the correct decoding when the sequence length or table size grows. We propose an adaptive sorting scheme to reduce the memory accesses and design complexity. We reduce the table size by using symbol-merging and table-merging schemes. In addition, the proposed Black-Box model improves the accuracy of performance estimation by a novel measurement of "symbol-alias" and also achieves a better tradeoff between performance and complexity. Further, no side information is transmitted and the proposed soft VLC decoder is bandwidth-efficient. The proposed design is evaluated using the model of MPEG-4/UDP-Lite/UEP/AWGN, and hence it is standard-compliant. We averagely improve the peak signal-to-noise ratio by 0.4∼2.9 dB as compared with traditional VLC decoding and standard-support reversible VLC decoding schemes.**

***Index Terms*—MPEG-4, soft-input soft-output, variable length decoder (VLD), wireless transmission.**

I. Introduction

VARIABLE length codes (VLCs), also called Huffman codes [1] are commonly used to approach the entropy rate of a given data source. They are extensively used in recent image and video coding standards including JPEG, MPEG-1/2/4 and the new design of H.264 [2]. However, most of the VLC designs are highly sensitive to error disturbances. Table look-up decoding method may render extremely vulnerability and lose synchronization over a noisy channel. Although many conventional methods like automatic repeat request (ARQ) and forward error correction (FEC) reduce the effect of channel errors, these solutions have been found to be expensive in band-limited communications of delay sensitive video signals [3].

Particularly, ARQ-based designs are inadequate for the broadcast transmission due to the need of backward channel. Besides, they may induce significant delay that would potentially result in network congestion; while FEC designs may be bandwidth-inefficient when the channel conditions are fairly mild, and fine-tune to a particular error-rate when the channel condition differs. Therefore, it is strongly interesting to look for an alternative design to reduce the error sensitivity of variable length encoded video source.

In recent years, more and more researchers pay lots of attention about the source and channel design jointly. To improve the error robustness on VLCs, we focus on the error recovery method. It can be partitioned into three levels that are source level, channel level and joint source-channel level. Many source-level error recovery methods are suggested including reversible VLC (RVLC) [4], and self-synchronization VLC (SSVLC) [5]. These methods do not consider any channel behavior and the error rate improvement is still insufficient. Further, the usage of channel-level error recovery is very expensive for the band-limited system. A tradeoff between source and channel level is proposed that can be termed as joint source and channel (i.e., JSC) design. The idea of JSC design has been gaining increasing attention because of the significant growth of multimedia wireless communications. Besides, the channel conditions about broadcasting on DVB system [6] faced the channel behaviors without backward notifications.

To improve the error resilience, soft VLC decoders with joint source and channel design have been proposed [7]–[10]. Such algorithms generally need to maintain many states when the sequence length or table size grows. Hence, soft VLC decoders have problems of high complexity and intensive memory accesses. Reduced complexity algorithms with sub-optimal solution have been made in [11]. However, they didn't apply the algorithms to larger VLC table. In this paper, we propose an adaptive and scalable soft VLC decoding algorithm to significantly reduce the complexity. Particularly, to simplify the algorithm, we transform the metric derivation in soft-input soft-output (SISO) algorithm [12] into the symbol-constrained directed graph [8], [11] for the symbol-based VLC trellis decoding [13], [14]. Through the help of graph representation, we develop a modified sorting scheme that achieves the same decoded error rate with fewer states. To reduce the table size, we merge two symbols with the same prefix into one symbol, and we also merge the identical prefix and codeword between two tables. By the proposed merging scheme, we can greatly reduce the table size as well as complexity at the cost of slight error rate increment.

Bystrom *et al.* [9] used the minimal Hamming distance (d_H) to quantify the relation between table and performance. The performance represents the capability of error recovery. However, it is still inaccurate when the different tables reach the same d_H. We propose a novel measurement to improve the accuracy of performance estimation. Further, we reduce the penalty of over-

Manuscript received March 1, 2005; revised January 21, 2006. This work was supported by the National Science Council of Taiwan, R.O.C., under Grant NSC94-2215-E-009-046, and by the NCTU-MTK Research Program. This paper was recommended by Associate Editor H. Watanabe.

T.-M. Liu and C.-Y. Lee are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: mingle@si2lab.org; cylee@si2lab.org).

S.-Z. Wang is with MediaTek Inc., Hsinchu 300, Taiwan, R.O.C. (e-mail: SJ_Wang@mtk.com.tw).

B.-J. Shieh is with the Sunplus Technology Company, Ltd, Hsinchu 300, Taiwan, R.O.C. (e-mail: bjshieh@sunplus.com).

Color versions of one or more figures/tables in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2006.887133

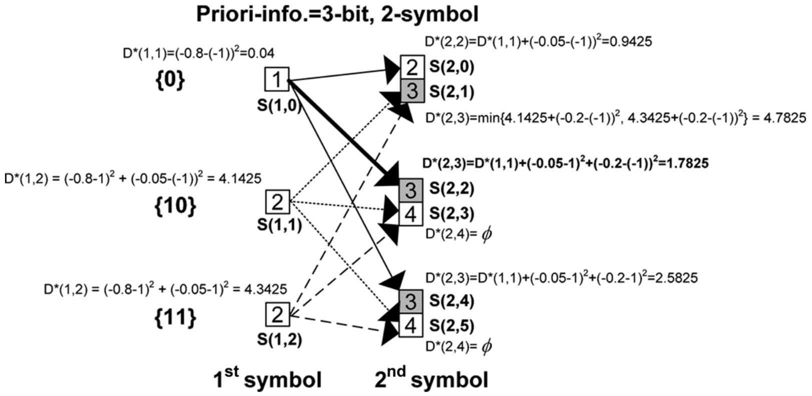

Priori-info.=3-bit, 2-symbol
D*(1,1)=(-0.8-(-1))^2=0.04
{0}
1
S(1,0)
D*(1,2) = (-0.8-1)^2 + (-0.05-(-1))^2 = 4.1425
{10}
2
S(1,1)
D*(1,2) = (-0.8-1)^2 + (-0.05-1)^2 = 4.3425
{11}
2
S(1,2)
D*(2,2)=D*(1,1)+(-0.05-(-1))^2=0.9425
2
S(2,0)
3
S(2,1)
D*(2,3)=min{4.1425+(-0.2-(-1))^2, 4.3425+(-0.2-(-1))^2} = 4.7825
D*(2,3)=D*(1,1)+(-0.05-1)^2+(-0.2-(-1))^2=1.7825
3
S(2,2)
4
S(2,3)
D*(2,4)= ϕ
D*(2,3)=D*(1,1)+(-0.05-1)^2+(-0.2-1)^2=2.5825
3
S(2,4)
4
S(2,5)
D*(2,4)= ϕ
1st symbol
2nd symbol

Fig. 1. Proposed translation between symbol-constrained directed graph and SISO algorithm.

design and observe the tendency of performance through the proposed Black-Box (B-B) model. Thus, the proposed model reaches a better tradeoff between performance and complexity.

The proposed soft VLC decoder is verified with not only a simple VLC table but also a practical MPEG-4 table. From the analysis of a simple VLC source data, our algorithm can save 37.3% of memory accesses in a channel condition of SNR$=8$ dB. Further, we obtain the optimal parameters through the B-B model. Compared to the existing soft VLC decoder [15], [16], we merge the VLC tables and codewords to reduce the decoding complexity. Finally, the proposed design shows about 0.4 dB∼2.9 dB PSNR improvement as compared with the table look-up decoding and reversible VLC decoding in additive white Gaussian noise (AWGN) channel.

The rest of this paper is organized as follows. Section II briefly introduces the SISO algorithm. Section III presents our low-complexity design of soft VLC decoder. Section IV describes the proposed B-B model. After that, we present the complexity and performance evaluation on MPEG-4 in Section V. Finally, concluding remarks are made in Section VI.

II. Approximated Decoding of SISO Algorithm

A. Graph Representation of SISO

SISO decoding technique [12] is considered as an exhaustive decoding procedure to resist the error disturbance in a noisy channel. It estimates and searches on the tree-like path in the existence of AWGN. The input sequence is a packet-based transmission. We don't exploit soft output for the iterative decoding due to the real-time demands. It uses L bits and equivalently N symbols to represent the priori information in one packet. Specifically, the SISO algorithm chooses the estimated sequence X as the one that maximizes the joint probability for the observed sequence Y [12]. Based on the similar estimation, we use the graph representation to simplify the SISO algorithm when table size or decoded symbol grows.

To help understanding our simplified algorithm, we use a symbol-constrained directed graph representation [8], [11]. Assume we have a simple VLC table with only 3 symbols {0,10,11} and a packet that includes 3 bits (and equivalently 2 symbols of *a priori* information) with content as "0 10." After BPSK modulation, the modulated sequence is $\{-1,+1,-1\}$. When the packet is transmitted over the AWGN channel, the received packet may become $\{-0.8,-0.05,-0.2\}$ (i.e., error occurs in second bit).

Simple VLC table

Code length	Code word
1	1
3	011
4	0100
4	0101

(a)

Simplified MPEG VLC table

Code length	Code word	sign	LAST
2	1s	1	1
4	011s	1	0
5	0100s	1	0
5	0101s	1	0

(b)

Fig. 2. (a) Original and (b) practical VLC table.

Fig. 1 depicts the graph representation for this example. The intermediate metric $D^*(i,j)$ denotes the cumulative square error of ith symbol and jth bit in each symbol state. $S(m,n)$ is the symbol state decoded with m-symbol and has the index of n among the identical m. The number inside each square is identical to "j" of $D^*(i,j)$. The operation of "minimum" is exercised in $S(2,1)$. Furthermore, the minimal metric is survived and the others are pruned. Therefore, we can decide the shaded squares as the final candidates. $S(2,2)$ is the minimum among them and can be used to trace back to $S(1,0)$ to obtain the correct results of $\{0,10\}$.

B. Modification of VLC Table

To apply our algorithm in MPEG-4 standard [17], we introduce the "sign" and "LAST" field from the original Huffman table. The extra fields of "sign" and "LAST" are essential for the SISO decoding procedure. In Fig. 2(a), we modify the simple VLC table as Fig. 2(b). In our proposed approach, we exploit the number of "LAST" to represent the priori information. The LAST field indicates whether or not the decoded codeword is a last symbol in this 8 × 8 block (i.e., "0": there are more nonzero coefficients in this block, "1": this is the last nonzero coefficient in this block). The number of "LAST" in one packet is defined by MPEG-4 standard and extracted from the packet header. Thus, it is available to the user without requiring any side information to be transmitted.

To deal with the "s" parameter appended in each symbol, we use a simple hard decoding with table-look-up method. The induced "sign" field in Fig. 2(b) represents the bit number of "s" in our design. The "sign" field is 1 when the "s" of each symbol

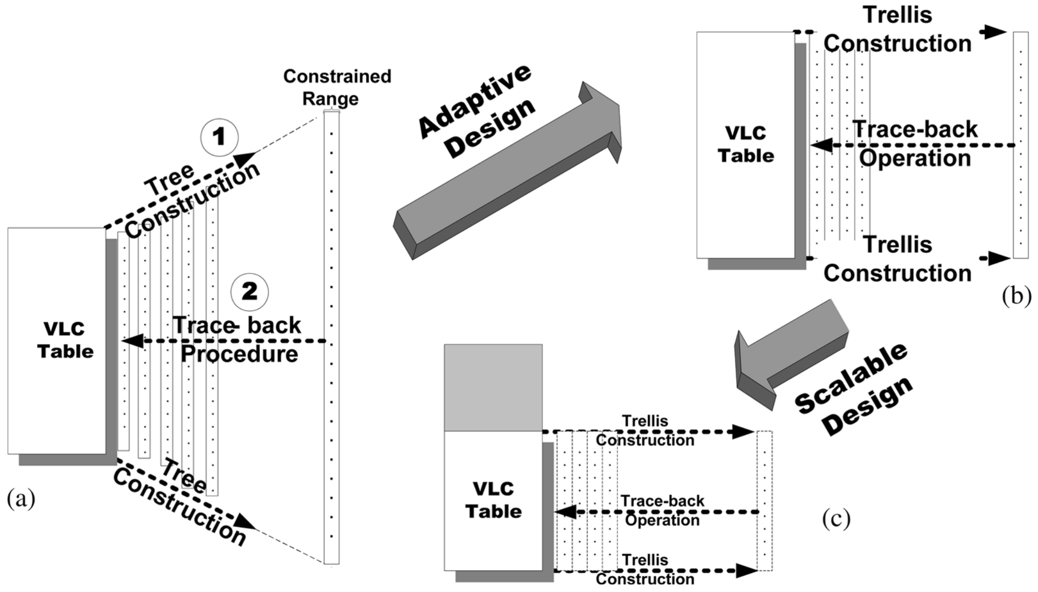


Fig. 3. Low-complexity design. (a) High-level description of the decoding procedure. (b) Graph representation after applying adaptive design. (c) Graph representation after applying scalable design.

is appended by 1-bit. More discussion about the "sign" field is provided in Section III-B.

C. Complexity Reduction

Fig. 3(a) depicts the high-level graph description of the decoding procedure in Fig. 1. The overall decoding algorithm can be partitioned into two main parts. One is the state construction and the other is the trace-back decoding procedure. Each state requires addition, shifting and selection to perform the similar function of add-compareselect (ACS) unit in the Viterbi decoder. Moreover, it requires sufficient storage elements to keep the results for the trace-back procedures. Therefore, the large numbers of states need lots of operations and storage elements, leading to the problem of high complexity in SISO decoding process.

The complexity of SISO algorithm is greatly related to the number of survival states. Further, the numbers of states also depend on table size and sequence length. In Table I, we show the complexity analysis with a VLC sequence of fixed 300 symbols and encoded through the VLC Table in Fig. 2(a) (where $\{T, S\} = \{4, 3\}$ in the equation of Table I). The numbers of storage elements occupy a large portion of overall operations. Therefore, the complexity can be denoted by the number of storage elements as well as survival states. In Fig. 3(b), we propose a state-trellis construction by adaptively selecting the survival states to reduce the access times of storage elements. In Fig. 3(c), by the symbol-merging and table-merging schemes, we can greatly reduce the table size at a cost of minor error rate increment.

III. Low-Complexity Soft VLC Decoder

A. Algorithm With Adaptive Selection

The SISO algorithm requires many states since practical tables have many entries. It becomes inadequate for VLSI implementation when the numbers of survival states grow. To reduce the number of states as well as memory accesses, we propose an *Adaptive* AMAP-2 (A-AMAP-2) to reduce memory accesses and overall complexity (see the adaptive design in Fig. 3). Lamy and Pothier [11] introduced the approximated decoding method 2 (AMAP-2) to reduce the complexity. However, this approach is not robust to the channel variation. They induced more states to retrieve the metric in the error-occurred region and increased the penalty to the error-free region.

TABLE I
Complexity Analysis of SISO Algorithm at 300 Symbols and VLC Table of Fig. 2(a)

Optimal Solution	# of survival states = 360,000 = [1] $\frac{T(S-1)}{2} \times N^2 + \left(\frac{3-S}{2}\right) T \times N$		
Process Element	Storage Element (i.e. survival states)	Shift & Selection	Addition
No. of element	360,000	1,870	3,740

[1] T: total entries of codeword, S: number of type for code-length, N: decoded symbol number.

To resist the channel variation, we propose to adaptively select the states and reduce the number of survival states. To address our improvement and differences as compared with AMAP-2 [11], we use the simple VLC table in Fig. 2(b) as an example. The corresponding graph representation is developed in Fig. 4(a). To clearly show the metric variation in each state, we just omit the arrows and the indication of "LAST." In Fig. 4(b), we show the sorting algorithm via the number of states in AMAP-2. By pruning the square of the same bit position in Fig. 4(b), we obtain Fig. 4(c) that can be used in comparison with our proposed *Adaptive* AMAP-2.

The main difference of AMAP-2 and *Adaptive* AMAP-2 is the sorting scheme in the y axis. Fig. 5(a) shows that AMAP-2 requires at least three (i.e., $M_{\mathrm{AMAP-2}}$) states for the correct decoding given the specified threshold. The correct states are la-

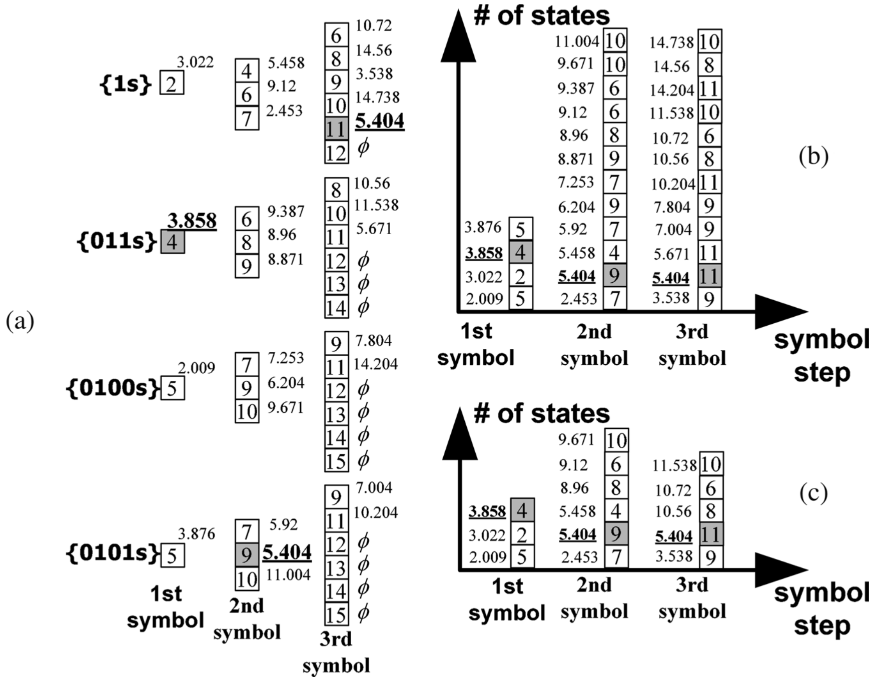
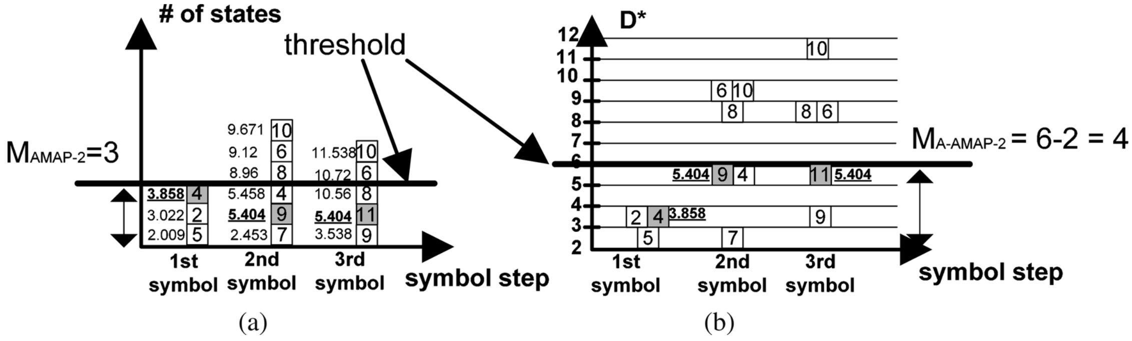


Fig. 4. (a) Graph representation of Fig. 2(b). (b) Sorting scheme before pruning process. (c) Sorting scheme after pruning process.

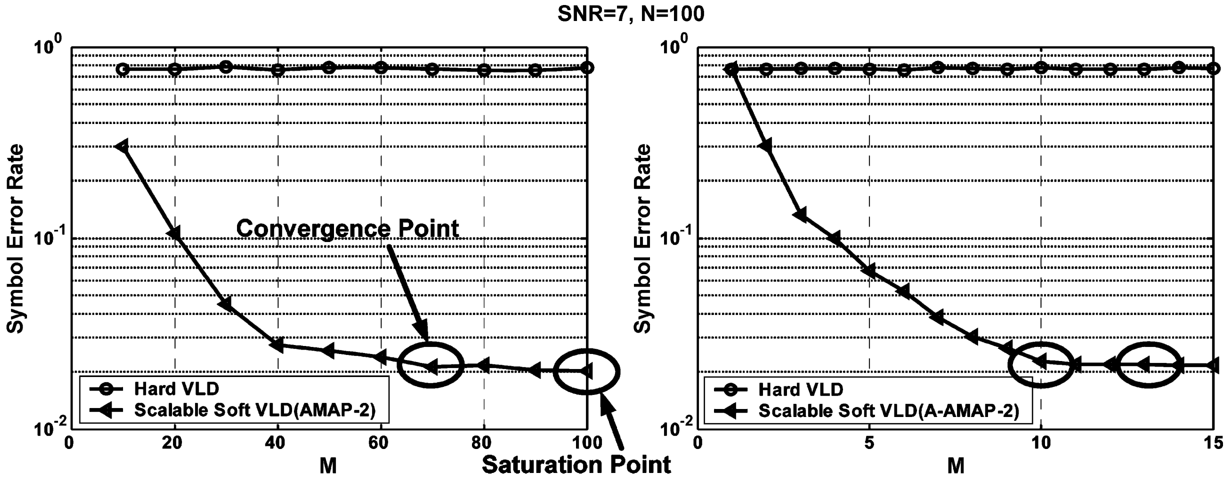


(a) (b)

Fig. 5. Comparison between (a) the AMAP-2 and (b) the proposed adaptive AMAP-2.

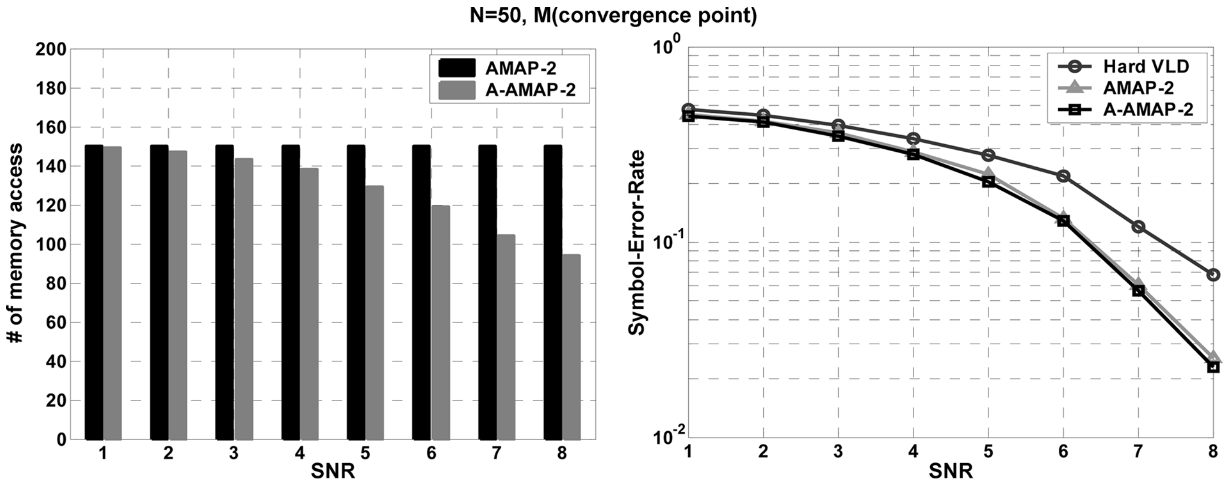
beled with the shaded region. In Fig. 5(b), by employing the D^* in the sorting algorithm instead of the number of states, the state metric range above the minimal metric for the correct decoding is 4 $(M_{\mathrm{A-AMAP-2}} = 6 - 2)$. As a result, we can find that there are 9-state and 8-state survived in AMAP-2 and *Adaptive* AMAP-2, respectively, for the correct decoding. Such improvement on the state number reduction increases when the errors occur rarely. This novel scheme adaptively selects the number of survived states in each symbol step, and that is why we call it as *Adaptive* AMAP-2 (i.e., A-AMAP-2). In Fig. 5(b), the proposed A-AMAP-2 survives more states when the errors occur frequently, such as first symbol step. Furthermore, fewer states are survived in the less error region, such as the third symbol step. It is more robust to the channel variation and provides the variable states to be survived in each symbol step.

From the analysis of performance and complexity, it's more imperative to decide the number of "M" to be survived (see $M_{\mathrm{AMAP-2}}$ or $M_{\mathrm{A-AMAP-2}}$ in Fig. 5). To exploit a meaningful metric to address the performance, we list the symbol error rate (SER) to indicate the decoding performance. The larger M achieves the better performance at the price of high complexity. The problem occurs also in the smaller M. Thus, inappropriate M will be harmful to the performance or complexity. The empirical value of M is determined from experiments. We define the saturation and convergence point to approach the optimal value. Given a simple table in Fig. 2(b), Fig. 6 depicts the measurement of "saturation M" and "convergence M." The SER will decrease with the increasing "M." Intuitively, we select the convergence point as "M" for the tradeoffs between performance and complexity. From the simulation results of Fig. 7, we assume that one survived state will cost one access of memory element. After that, we choose an optimal threshold (i.e., convergence M) as the simulated parameter, and compare the performance of our proposed A-AMAP-2 and the AMAP-2 versus channel condition. Obviously, given the same performance, our algorithm obtains less design complexity and memory accesses in high SNR. Our algorithm saves 37.3% of memory accesses at most as compared with AMAP-2 [11] in a SNR of 8 dB.



(a) (b)

Fig. 6. Performance with convergence and saturation point in (a) AMAP-2 and (b) A-AMAP-2.



(a) (b)

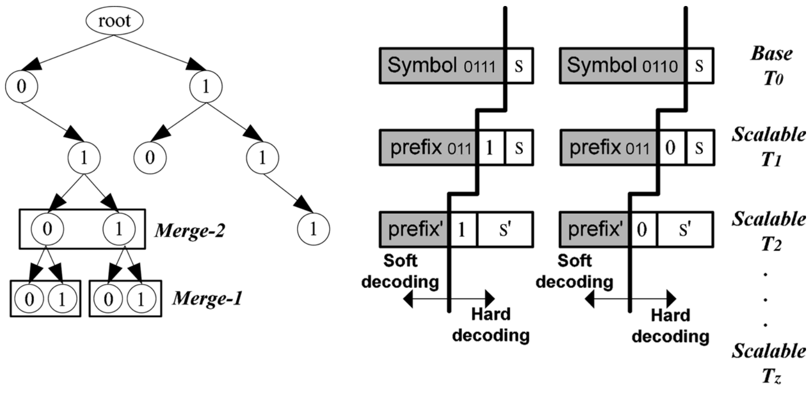
Fig. 7. Comparison in (a) memory accesses (i.e., number of survival states) and (b) performance.

B. Scalable Algorithm With Symbol Merging

To apply the SISO algorithm to MPEG-4 system, we propose a scalable scheme with symbol merging algorithm to reduce the table size. We utilize the redundancy exhibiting in different symbols to perform the merging algorithm. We consider a simple VLC table as a tree-structure in Fig. 8(a). The proposed symbol-merging scheme searches the symbols with the identical prefix and merges them into single merged-symbol. In Fig. 8(b), the original SISO decoding algorithm is a special case when z is equal to 0 (i.e., Base T_0). In other words, there is no hard (i.e., table look-up) decoding performed except "sign" bit. Such case achieves the smallest error rate with the highest complexity. However, the code-length of prefix symbol with soft decoding will decrease when the index "z" increases. Meanwhile, the number of bits with hard decoding will increase. As a result, it can be considered as a hybrid scheme that combines the hard and soft decoding methods.

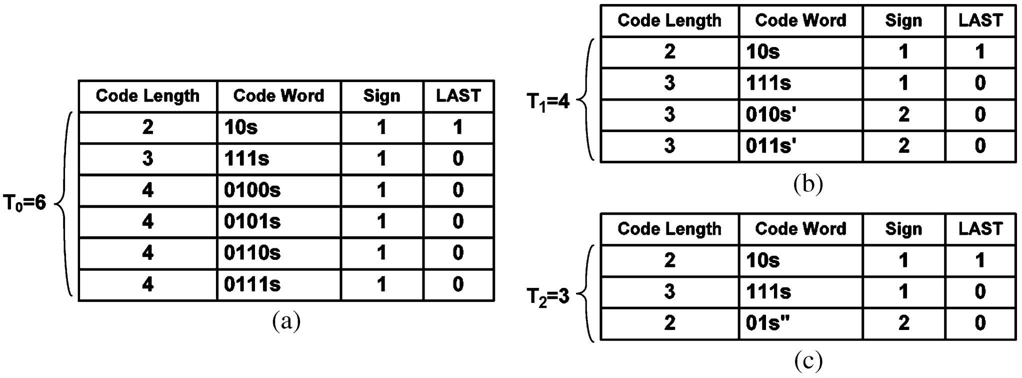
We use a simple example to illustrate the proposed scheme in Fig. 9 where "T_i" represents the number of symbols after the *Merge-i* operation. As shown, after the operation of "Merge-1", the table size is decreased by 2. Further, with the "Merge-2" operation, the total number of symbols becomes 3. The introduced "sign" field represents the number of "s" appended in the corresponding symbol. The "sign" field will increase when both symbols with the identical "SIGN" and "LAST" have been merged into one.

To quantify the number of symbols after the symbol-merging scheme, we propose the metric of "balance degree" (B.D.) in (1). The metric of B.D. is between 0 and 1. In (1), the denominator represents the maximal value as well as a special table with a complete tree-structure. It leads to "$z \times 0.5$" after the z times of summation where the ratio of $T_i + 1$ over T_i is fixed at 0.5.



(a) (b)

Fig. 8. (a) Tree-structured VLC. (b) Scalable scheme with hard and soft decoding.



$T_0=6$

Code Length	Code Word	Sign	LAST
2	10s	1	1
3	111s	1	0
4	0100s	1	0
4	0101s	1	0
4	0110s	1	0
4	0111s	1	0

(a)

$T_1=4$

Code Length	Code Word	Sign	LAST
2	10s	1	1
3	111s	1	0
3	010s'	2	0
3	011s'	2	0

(b)

$T_2=3$

Code Length	Code Word	Sign	LAST
2	10s	1	1
3	111s	1	0
2	01s"	2	0

(c)

Fig. 9. VLC table with (a) Merge-0, (b) Merge-1, and (c) Merge-2 operations.

TABLE II
REDUCTION OF TABLE SIZE BY SYMBOL-MERGING ALGORITHM

Standard	MPEG-2		MPEG-4	
Table	$\text{INTRA}_{\text{TB-15}}$	$\text{NON-INTRA}_{\text{TB-14}}$	$\text{INTRA}_{\text{TB-16}}$	$\text{NON-INTRA}_{\text{TB-17}}$
T_0	113	114	103	103
Scalable T_1	65	60	61	56
Scalable T_2	45	34	48	38
B.D.(2)	73.2%	90.7%	62%	77.8%

Therefore, the balance degree of Fig. 9 is 58% in the condition of "$z = 2$"

$$\text{B.D.}(z) = \frac{\text{real reduction}}{\text{complete reduction}} = \frac{\sum_{i=0}^{z-1}\left(1-\frac{T_{i+1}}{T_i}\right)}{\sum_{i=0}^{z-1}(1-0.5)} = \frac{\sum_{i=0}^{z-1}\left(1-\frac{T_{i+1}}{T_i}\right)}{z\times 0.5}. \quad (1)$$

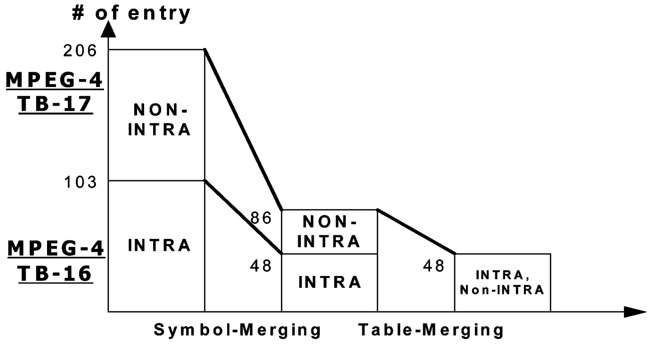


Fig. 10. Table size reduction through symbol and table merging scheme.

To prove that B.D. is a meaningful factor to our merging scheme, we measure the B.D. using the transform coefficient tables (i.e., AC TCOEF) in MPEG-2 and MPEG-4. As shown in Table II, we find that the higher of the B.D., the more reduction of the table size. The B.D. value of nonintra table is lager than that of intra one. It can be explained by the fact that there is more redundancy exploited in terms of symbol-structure. That is to say, the nonintra table is more efficient than intra table after performing symbol-merging scheme.

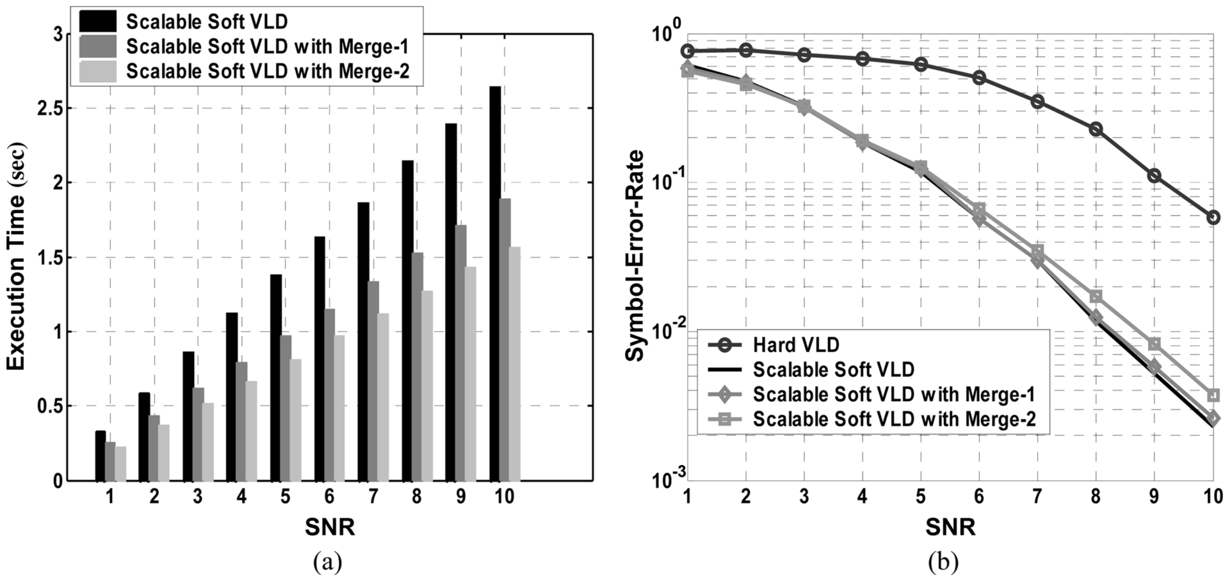


Fig. 11. Evaluation of (a) execution time and (b) performance with different symbol-merging times in Fig. 9.

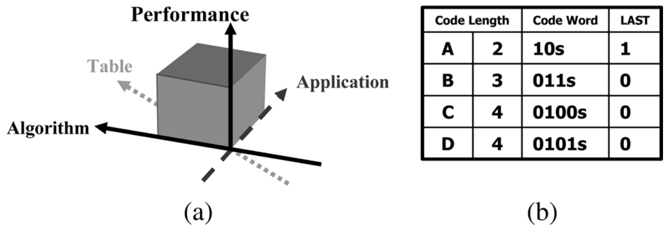
C. Scalable Algorithm With Table Merging

It is essential for switching tables on the soft VLC decoding process, since there are intra and nonintra coefficients in the ac partition of whole bit stream. To share the same soft VLC decoder on the different VLC tables, we propose a codeword merging and prefix merging method to reduce the implementation cost and memory accesses.

1) Codeword Merging: Although most VLC coding tables are generated based on the Huffman procedure, one codeword still has high probability to exist in many coding tables. If this case occurs, it is unnecessary to duplicate the codeword information in memories for every table that uses this codeword. A codeword merging is applied to set this codeword as a merged codeword and reuse the codeword information when the coding tables are required. Therefore, the information redundancy among coding tables is exploited.

2) Prefix Merging: According to the Huffman property, one codeword cannot be the prefix of another codeword in a table but this rule does not hold among different tables. Frequently, a short codeword in one table will be the prefix of a long codeword in other tables. When these codewords are found, a prefix merging is performed by storing the long codeword as a merged VLC codeword and the lengths of the VLC codewords in each table. As a result, the information redundancy among tables is further exploited.

In Fig. 10, the proposed merging method reduces the number of entries in one or different tables. In addition, the table merging method is harmless to the error rate of soft VLC decoder; on the other hand, the symbol merging algorithm is a lossy merging scheme since the performance (i.e., SER) will degrade when the numbers of merging z increase. There are tradeoffs between the complexity reduction and performance loss. In Fig. 11, the execution time reduces at the cost of little error rate increment. In this case, we choose $\mathrm{Merge} - 1$ for a better compromise since the SER will greatly increase when "i" is larger than 1 (i.e., $\mathrm{Merge} - i$ where $i > 1$).



Code Length		Code Word	LAST
A	2	10s	1
B	3	011s	0
C	4	0100s	0
D	4	0101s	0

Fig. 12. (a) B-B model. (b) Evaluation of source table.

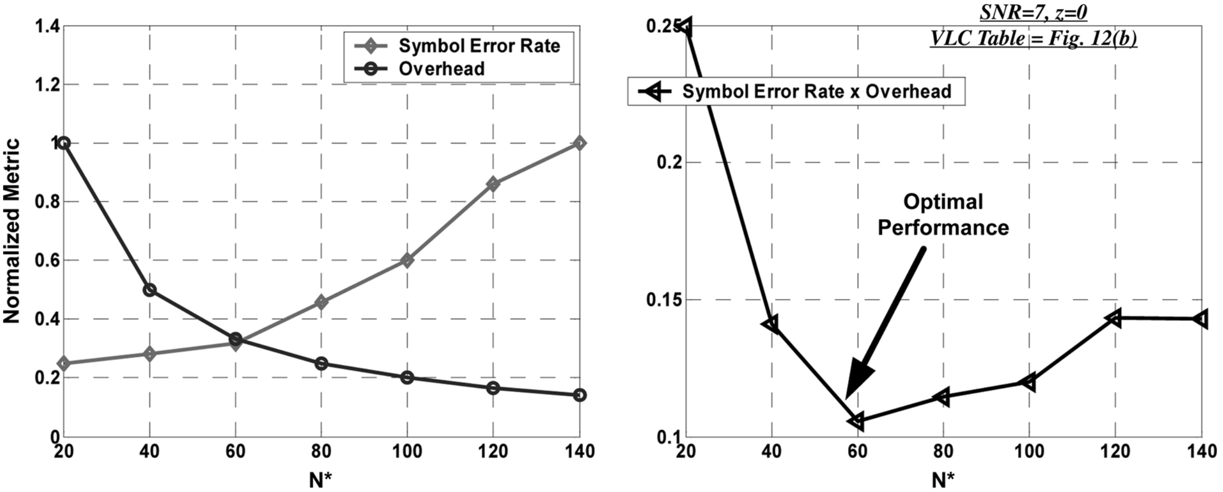
IV. Performance Modeling

A. Black-Box (B-B) Model

To optimize our proposed adaptive and scalable soft VLC decoder in Section III, it is crucial to reach a better tradeoff between performance and complexity. Because there are performance variations with different parameters, we present a B-B model to observe the tendency of performance, and then optimize the performance. Specifically, the performance of the soft VLC decoder represents the capability of error recovery. We list the SER in bit level or PSNR in decoded frame-level to indicate the decoding performance. We propose a B-B Model and introduce some parameters to describe the complexity issue. They are independent and composed of algorithm-sensitive, application-sensitive and table-sensitive. In the table-sensitive, we propose a novel measurement of "symbol-alias" to provide accurate performance estimation for different tables. Fig. 12(a) depicts the proposed B-B for the performance modeling and uses Fig. 12(b) as the source VLC table for the following illustration. Finally, we reach the optimal parameters using a performance modeling [22] for a given VLC table, application and decoding algorithm.

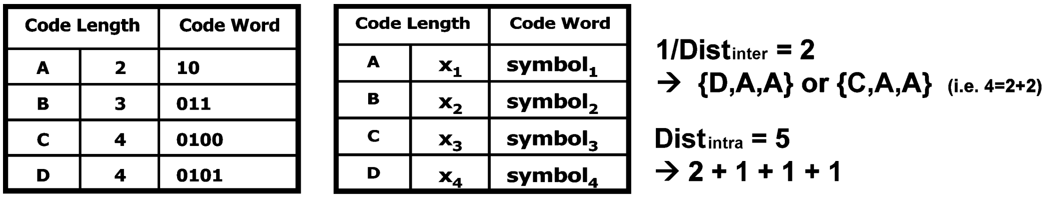
B. Algorithm-Sensitive Parameters

The algorithm-sensitive parameters are sensitive to algorithms; that is to say, different algorithms are characterized by different parameters. Using the proposed algorithm in the



(a) (b)

Fig. 13. Optimization of performance in (a) SER and overhead and (b) SER × overhead.



Code Length		Code Word
A	2	10
B	3	011
C	4	0100
D	4	0101

Code Length		Code Word
A	x_1	symbol_1
B	x_2	symbol_2
C	x_3	symbol_3
D	x_4	symbol_4

1/Dist

$1/\text{Dist}_{\text{inter}} = 2$
→ {D,A,A} or {C,A,A} (i.e. 4=2+2)

$\text{Dist}_{\text{intra}} = 5$
→ 2 + 1 + 1 + 1

(a) (b)

Fig. 14. Symbol alias of VLC table. (a) Simple table in Fig. 2(b). (b) Source table in (2) and (3).

previous section as an example, the parameter "M" and "z" are considered as an essential factor to approach the tradeoff between performance and complexity. In the adaptive design, Fig. 6 shows a better tradeoff between performance and complexity when M reaches a convergence point. Further, Fig. 11 describes that the optimal choice is achieved when z is equal to $1(\text{Merge} - 1)$. Thus, "$\text{Merge} - 1$" obtains the maximal reduction of complexity at the cost of minor performance loss.

C. Application-Sensitive Parameters

The application-sensitive parameters are unrelated to the algorithms and decided by the extrinsic applications. The packet size "N" is the most impressive factor to achieve the optimal performance. In this section, we regard it as the decoded number of symbols "N*" for the simplicity. Fig. 13(a) describes that the performance can be expressed by the normalized SER and overhead. We assume that the 15-bit resynchronization marker is inserted in the period of N*-symbol. From Fig. 13(b), it achieves the minimal SER and overhead when "N*" is equal to 60. There is no need to consider the algorithmic complexity because it has to be optimized from the algorithm-sensitive parameters.

D. Table-Sensitive Parameters

To estimate the performance with different tables, Bystrom *et al.* used the minimal Hamming distance (d_H) to quantify the relation between VLC table and performance [9]. It is still inaccurate when the different tables reach the same d_H. We propose a novel measurement of "symbol alias" to quantify their relation and provide more accurate performance estimation when their Hamming distances are the same.

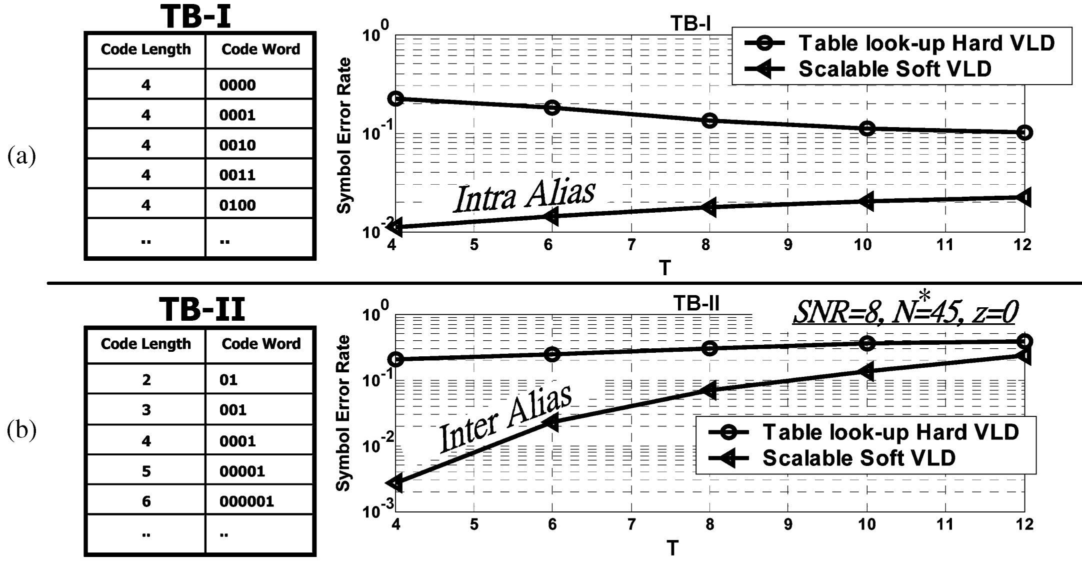
We introduce the table-sensitive parameter "T" and the symbol alias to obtain more accurate estimation. The parameter "T" (i.e., T_i in Fig. 9) denotes the number of symbols for a given VLC table. Besides, the symbol alias comprises two components. One is the distance of "Inter Alias" (i.e., $\text{Dist}_{\text{inter}}$) described by (2). The other is the distance of "Intra Alias" (i.e., $\text{Dist}_{\text{intra}}$) described by (3). Fig. 14 depicts the results of the following equations. The $1/\text{Dist}_{\text{inter}}$ is the number of elements for the specific set, which calculates whether the code-length of one symbol is the combination of the others. The $\text{Dist}_{\text{intra}}$ is the summation of minimal Hamming distance for each symbol.

$$\text{Set}_{\text{inter}} = \left\{ \{x_1, x_2, x_3, \ldots, x_T\} \middle| \sum_{i=1, x_i \neq \max\limits_i \{x_i\}}^{T} x_i = \max_i x_i \right\} \tag{2}$$

$$\begin{aligned}\text{Dist}_{\text{inter}} &= \frac{1}{\text{NumOfElement}(\text{Set}_{\text{inter}})} \text{Dist}_{\text{intra}} \\ &= \sum_{i=1}^{T} \min(D_{i1}, D_{i2}, D_{i3}, \ldots, D_{ij,j\neq i}, \cdots, D_{iT-1}, D_{iT})\end{aligned}$$

where

$$D_{ij} = d_H\{\text{symbol}_i, \text{symbol}_j\}. \tag{3}$$



TB-I

Code Length	Code Word
4	0000
4	0001
4	0010
4	0011
4	0100
..	..

TB-II

Code Length	Code Word
2	01
3	001
4	0001
5	00001
6	000001
..	..

Fig. 15. Performance evaluation of (a) "intra alias" and (b) "inter alias" in different numbers of symbol (i.e., T).

Performance(C2c) > Performance(C2d) > Performance(C2a)

T=8

Codes C2a

Code Length		Code Word
A	1	0
B	2	10
C	4	1100
D	4	1101
E	4	1110
F	5	11110
G	6	111110
H	6	111111

Codes C2c

Code Length		Code Word
A	3	010
B	3	101
C	4	0011
D	4	1100
E	5	01001
F	5	01111
G	5	10010
H	5	11100

Codes C2d

Code Length		Code Word
A	2	00
B	2	11
C	5	01010
D	5	10101
E	5	10000
F	5	01111
G	5	11000
H	5	00110

(a)

	Proposed		[9]
VLC Table	Inter Alias	Intra Alias	Min{d_H}
C2a	1/7	23	1
C2c	∞	22	2
C2d	∞	13	2

(b)

Fig. 16. (a) VLC tables. (b) Measurements for the same source.

The inter alias is more sensitive to "T" than intra alias and induces more performance loss. In Fig. 15(a), the intra alias of TB-I results from the bit alias with the symbols of identical code-length. The increase of "T" leads to the increase of SER in the soft VLC decoding. However, the SER of table look-up decoding decreases when the "T" grows. The reason is that the extra symbols prevent the decoded-symbol loss and error propagation. Further, the inter alias of TB-II results from the code-length alias with the symbols of different code-lengths. In Fig. 15(b), both SERs increase and provide more performance loss than intra alias.

To prove that the proposed symbol alias is a meaningful number, we use the VLC table in [9] to recognize the difference of performance. In Fig. 16(a), we inverse the underlined bits in C2c and C2d for the correct VLC decoding. For our proposed measurement of Fig. 16(b), the Codes C2a achieves the worst performance because of the lowest $\mathrm{Dist}_{\mathrm{inter}}$ that induces more performance loss than $\mathrm{Dist}_{\mathrm{intra}}$. Further, there is no inter alias exist in C2c and C2d. The $\mathrm{Dist}_{\mathrm{inter}}$ is fixed at infinite (∞) by default. The C2c obtains better performance than C2d because of the higher $\mathrm{Dist}_{\mathrm{intra}}$ of C2c. The relation of performance with different tables is identical to [9]. In Fig. 16(b), the higher distance of coding table will lead to the less occurrence of symbol-alias (i.e., higher performance). The performance of C2c and C2d cannot be recognized in [9] when each symbol probability is unknown. We provide more accurate estimation than [9] and reduce the design time for the performance evaluation among different tables.

V. Simulation Results

A. Environment Setup

The overall simulation chain is depicted in Fig. 17. We verify the proposed low-complexity soft VLC decoder over the AWGN channel using BPSK modulation. The input sequence is MPEG-4 encoded with resynchronization marker and data partition. In data partition mode, we have assumed that the texture part, composed of a sequence of VLC codewords, be corrupted by AWGN. The other parts are of error free. This assumption can be achieved by exploiting the UEP or RCPC Codes [18].

The functionality of RCPC in Fig. 17 can be explained by Fig. 18(a). Particularly, the first three data partitions (i.e., header, dc coefficient and dc marker for I-frames; header, motion data and motion marker for P-frames) play an important role since the loss of header parts will introduce the loss of synchronization and corrupt the whole frames. Therefore, we pay lots of efforts to protect these parts (i.e., let channel coding rate $\mathrm{R}_1 \simeq 0$). For instance, we exploit the convolutional encoder and Viterbi decoder with $\mathrm{constraint\ length} = 7$ and $\mathrm{coding\ rate} = 1/2$ as specified in IEEE 802.11a. Furthermore, because about 72% of video packet is less important [i.e., Fig. 18(b)], there is no reason to protect them as the small coding rate (i.e., let $\mathrm{R}_2 \simeq 1$). Therefore, we assume that there is no channel coding performed

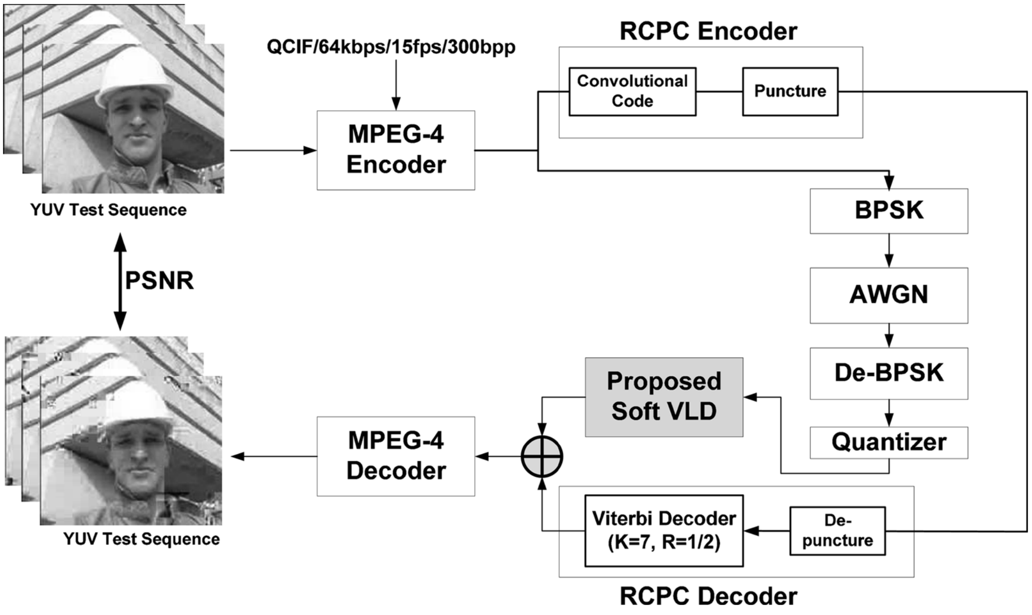
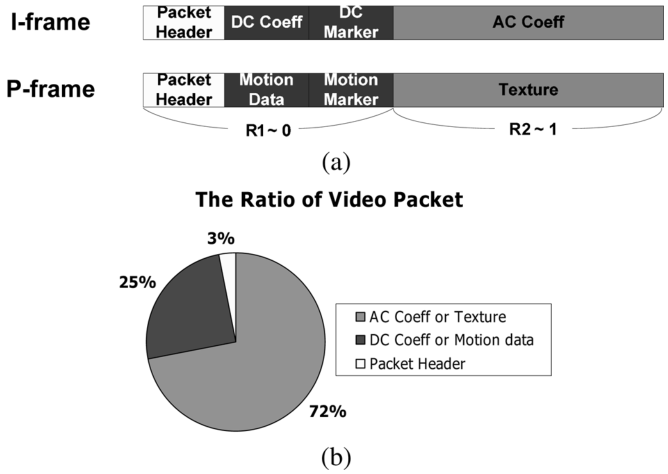


Fig. 17. Proposed overall simulation environment for soft VLC decoder.



(a)

(b)

Fig. 18. (a) Content and (b) ratio of one video packet in MPEG-4.

TABLE III
OVERLOOKING BIT ERRORS IN APPLICATION LAYER

Contents	OSI Layers
Application – MPEG	Application Layer
Transport – UDP or UDP-Lite	Transport Layer
Network – IP	Network Layer
UEP – RCPC Codes	Link Layer
Soft Information	Physical Layer

on this part, and recovered by the proposed soft VLC decoder instead.

For a tradeoff between complexity and performance, we exploit 4-bit (i.e., 16-level) quantizer to construct the overall simulation. We show five layers of open system interconnection (OSI) model [20] in Table III. In wireless network or mobile transmission, we assume that UDP-Lite [19] of transport layer and UEP of link layer are provided. Further, we exploit the soft bit stream after the 16-level quantizer and overlook the soft bit error of physical layer into the application layer. Based on the above statements, we are going to evaluate the proposed design on the MPEG-4/UDP-Lite/UEP/AWGN in the following.

B. Performance Evaluation on MPEG-4/UDP-Lite/UEP/AWGN

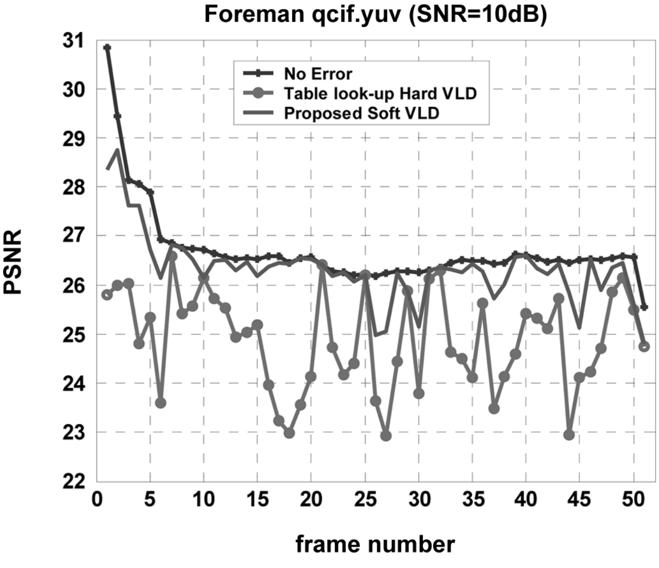
To optimize the performance and the complexity, we include the B-B model in our simulation. We use foreman (QCIF) as our test sequence and encode the sequence at 64 kbps, 15 fps for the wireless or mobile applications. In Fig. 19(a), the proposed soft VLD shows that more than 1 dB PSNR can be gained compared to the table look-up decoding at $\mathrm{BER} = 10^{-3}$ $(\mathrm{SNR} = 10\,\mathrm{dB})$. Moreover, the parameters determined by the B-B model are listed in Fig. 19(b). The "T" is determined from Table II with a given MPEG-4 table and the others are determined through the B-B model. We jointly optimize "M" and "z" and choose them as 200 and 2 for the complexity reduction. Then, we obtain "N" as 300 bits/packet for performance optimization. The performance improvement of the proposed scheme will become more prominent when the upper bound of "No error" increases.

Fig. 20 shows PSNR versus different channel conditions. The proposed soft VLC decoder is standard compliant and provides more than 1 dB PSNR gains as compared with the straightforward table look-up decoding (i.e., TLU VLD) when $\mathrm{BER} = 10^{-3}$. In addition, we also show that the improvement between the proposed soft VLC decoder and the standard-support RVLC decoder. In the programming cost point of view, the proposed soft VLC decoder requires 0.8 kb of memory space (i.e., 48 entries × 17 bit, see Fig. 10) which is less than 2.6 kb in RVLC (i.e., 168 entries × 16 bit) or traditional VLC decoder (i.e., 103 entries × 13 bit × 2). Finally, the proposed design is compared with the existing source-recovery and channel-recovery

TABLE IV
TRADEOFF BETWEEN ERROR CORRECTION AND CHANNEL BANDWIDTH

Coding Method	Performance (PSNR)	Channel Bandwidth or Coding Overhead
Table Look-Up VLD$_{[RM^2+DP^3]}$	Anchor	Anchor
Reversible VLD [4]	~0.5dB	2.2%
Self-Sync. VLD [5]	~0.5dB	1.6%
Proposed	~1.3dB	0%
Viterbi Decoder	~ Shannon Bound	100% for Viterbi
Turbo Decoder		100% ~ 600% for Turbo

[2] RM: Resynchronization Markers.
[3] DP: Data Partition



(a)

B-B Model	
Algorithm	{M, z} = {200, 2}
Application	{N} = 300bits/packet
Table	{T} = 48 entries

(b)

Fig. 19. (a) Y-component PSNR comparison within 50 frames. (b) Related parameters through B-B Model.

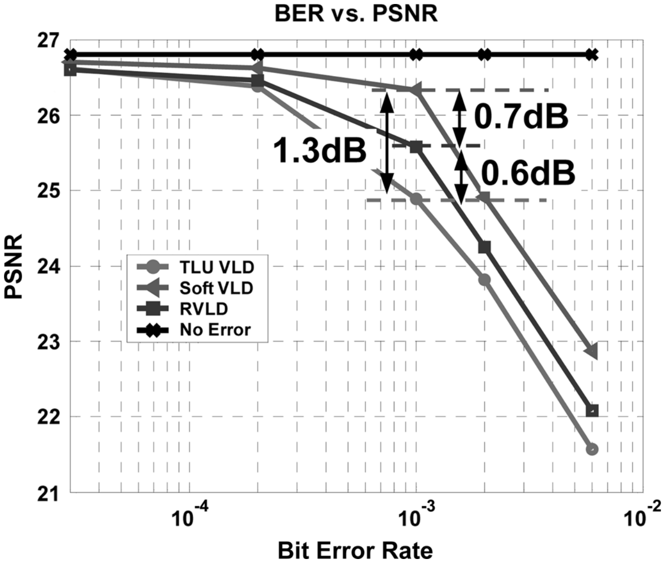


Fig. 20. Comparison between the proposed soft VLD and RVLD.

methods. Thanks to the advantage of joint designs, the proposed soft VLC decoder can achieve a compromise between the coding performance (i.e., capability of error recovery) and the channel bandwidth (see Table IV). In the source coding side, MPEG-4 supports the RVLC table to improve the error resilience, but it induces the coding overhead of 2.2% than the traditional VLC table [21]. In addition, Hong *et al.* [5] proposed a self-synchronization VLC decoding algorithm to reduce the coding overhead at the same performance compared with RVLD. On the other hand, the FEC codes provide high capability of error recovery, but it has to pay a great penalty of coding overhead (e.g., $\text{code rate} = 1/2, \text{coding overhead} = 100\%$). As shown in Table IV, the proposed soft VLC decoder is standard compliant and there is no any side information to be required. It is highly advantageous to the band-limited video transmission.

The above simulation is based on all I-frame assumption. We address the results in I-P-P structure of encoded bit stream below. It means that not only intra but also nonintra VLC coding tables should be exploited on the proposed soft VLC decoding procedure. In Table V, we show the performance improvement at the different source characteristics (i.e., Foreman, Suzie, and Silent), a PSNR improvement of 0.4 dB$\sim$2.9 dB can be achieved at the condition of 64 kbps, 15 fps, 300 bits/packet, and 12 frames/intra interval.

Compared to the existing algorithms, the proposed soft VLC decoder greatly reduces the number of survival states. In the table configuration of soft VLC decoder (see Table VI), the proposed design can easily switch table through table-merging scheme instead of duplicated structure [16]. In [16], it may require additional information to partition the intra and nonintra blocks into different channels. In [15], the survival states in each decoded symbol will grow exponentially because of the sequential decoding. To alleviate the problems on complexity, we propose a novel adaptive and scalable scheme to reduce the table size and the number of survival states. Specifically, we realize a low-complexity design with approximated SISO algorithm in trellis structure. The decoding complexity will not greatly increase when the decoded symbol step N grows. In Table VI, we need about 48$\sim$300 survival states for the correct decoding since the adaptive scheme and merging algorithm have been applied.

TABLE V
PSNR IMPROVEMENT WITHIN DIFFERENT VIDEO CHARACTERISTICS

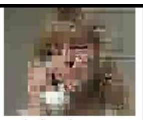
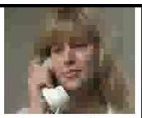
Video pattern	 original	 proposed	 original	 proposed	 original	 proposed
QCIF resolution, 64kbps, 15fps, 300bits/packet, 12frames/intra interval, 10dB/channel condition						
Total frame #	Foreman, 400 frames		Suzie, 150 frames		Silent, 300 frames	
PSNR Gain	0.417dB		2.896dB		0.7752dB	

TABLE VI
SUMMARY OF THE PROPOSED SOFT VLC DECODER

Soft VLC Decoder		**Proposed**	
Implementation		MPEG-4 + SISO	
Table Configuration		Reduced-Single (48 entries, also see Fig. 10)	
PSNR	R.M.	Enable	
	D.P.	Enable	
	E.C.[4]	Disable	
Source Characteristics		Foreman, QCIF, 64kbps, 15fps, 12frames/INTRA, 300bits/packet	
Testing Environment		AWGN+BPSK	
PSNR Improvement		0.4dB	
Condition		Decoding Complexity	
(@ decoded symbol step N)[5]		48~300	# of survival states
(@ 100 symbols in 1 packet)[6]		~13,920	
see Source Characteristics		~8.3 x 10^6	# of operations per second

[4] EC: Error Concealment.
[5] Each decoded symbol step has been described in Fig. 4 ~ 5.
[6] The total # of states in one packet for the decoding procedure of VLC.

To show more practical evaluation on complexity, we list not only the number of survival states but also the number of operations per second. Under the decoding specification of 64 kbps, 15 fps, and 300 bits/packet, the proposed design requires about 8.3×10^6 of operation per second.

VI. CONCLUSION

In this paper, an adaptive and scalable soft VLC decoder has been presented to significantly reduce overall decoding complexity. Our approach reduces complexity by simplifying the algorithm and reducing the table size. Specifically, we simplify the algorithm by adaptively selecting the survival states to reduce the number of states and memory accesses. Further, we reduce the table size by using a symbol-merging and table-merging scheme. In order to obtain better performance and complexity, we propose a novel measurement of “symbol-alias” to improve the accuracy of performance estimation based on B-B model. Experimental results show that our proposed adaptive scheme saves 37.3% of memory access as compared to the state-of-the-art algorithms [11] in a channel condition of $\mathrm{SNR} = 8$ dB. For the application of MPEG-4 video transmission, we provide a low-complexity design and only need constant survival state (i.e., about 48∼300 states) for each decoded symbol. Furthermore, our proposed soft VLC decoder has more than 0.4∼2.9 dB PSNR gain compared with the table look-up decoding method and the standard-support RVLD. To cope with the high error rate in the band-limited system, the proposed method provides bandwidth-efficient and error-recovery simultaneously.

REFERENCES

[1] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” *Proc. IRE*, vol. 40, pp. 1098–1101, Sep. 1952.

[2] *Draft ITU-T Rec. Final Draft Int. Standard of Joint Video Specification*, ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, May 2003.

[3] S. Shirani, F. Kossentini, and R. Ward, “An adaptive markov random field based error concealment method for video communication in an error prone environment,” in *Proc. ICASSP*, Mar. 1999, vol. 6, pp. 3117–3120.

[4] Y. Takishima, M. Wada, and H. Murakami, “Reversible variable length codes,” *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 158–162, Feb./Mar./Apr. 1995.

[5] G. Y. Hong, B. Fong, and A. C. M. Fong, “Error localization for robust video transmission,” *IEEE Trans. Consum. Elecron.*, vol. 48, no. 3, pp. 463–469, Aug. 2002.

[6] *Digital Broadcasting Systems for Television, Sound and Data Services; Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television*, ETS 300 744, European Telecommunications Standards Inst. (ETSI), 1997.

[7] 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.

[8] K. Sayood, H. H. Out, and N. Demir, “Joint source/channel coding for variable length codes,” *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 787–794, May 2000.

[9] M. Bystrom, S. Kaiser, and A. Kopansky, “Soft source decoding with applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 10, pp. 1108–1120, Oct. 2001.

[10] M. Park and D. J. Miller, “Joint source-channel decoding for variable length encoded data by exact and approximated MAP sequence estimation,” *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.

[11] C. Lamy and O. Pothier, “Reduced complexity maximum *a posteriori* decoding of variable-length codes,” in *Proc. GLOBECOM ’01*, Nov. 2001, vol. 2, pp. 25–29.

[12] J. Wen and J. Villasenor, “Soft-input soft-output decoding of variable length codes,” *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 689–692, May 2002.

[13] V. B. Balakirsky, “Joint source-channel coding with variable length codes,” in *Proc. IEEE Int. Symp. Inf. Technol.*, Germany, Jul. 1997, p. 419.

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

[15] L. Perros-Meilhac and C. Lamy, “Huffman tree based metric derivation for a low-complexity sequential soft VLC decoding,” in *Proc. Int. Conf. Commun.*, Apr. 2002, vol. 2, pp. 783–787.

[16] Q. Chen and K. P. Subbalakshmi, “Joint source-channel decoding for MPEG-4 video transmission over wireless channels,” *IEEE J. Sel. Areas Commun.*, vol. 21, pp. 1780–1789, Dec. 2003.

[17] *Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14496-2, Dec. 1999.

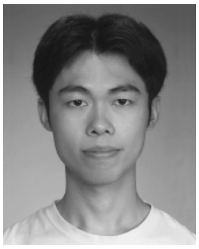
[18] J. Hagenauer, “Rate-compatible puncture convolutional codes (RCPC codes) and their applications,” *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[19] Larzon *et al.*, “The Lightweight User Datagram Protocol (UDP-Lite),” RFC 3828, Jul. 2004.

[20] *Information Processing Systems—OSI Reference Model—The Basic Model 1994*, ISO/IEC 7498-1.

[21] C. W. Tsai, T. J. Huang, K. L. Fang, and J. L. Wu, “A hybrid and flexible H.263-based error resilient and testing system,” in *Proc. IEEE TENCON*, Aug. 2001, vol. 1, pp. 19–22.

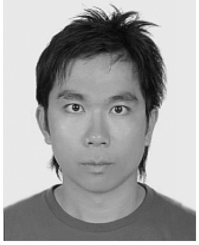
[22] T. M. Liu and C. Y. Lee, “A low-complexity soft VLC decoder using performance modeling,” in *Proc. IEEE ICIP’04*, Oct. 2004, pp. 3233–3236.



Tsu-Ming Liu (S’04) was born in I-Lan, Taiwan, R.O.C., in 1980. He received the B.S. and M.S. degrees in electronics engineering from National Chiao-Tung University, Taiwan, R.O.C., in 2002 and 2004, respectively. In 2004, he joined the Institute of Electronics Engineering of National Chiao-Tung University, where he is currently working toward the Ph.D. degree.

During 2004, he was an intern in Sunplus Technology Co. Hsinchu, Taiwan, R.O.C. From 2004 to 2006, he was a Lecturer in the Tze Chiang Foundation of Science and Technology (TCFST). His major research interests include binary shape coding, joint source and channel design, H.264/AVC video decoding, and associated VLSI architectures. He has published over 15 papers in the field of mobile multimedia.

Mr. Liu received a Best Impact Award from IEEE Taipei Section (2006/8). He is a member of Phi Tau Phi.



Sheng-Zen Wang received the B.S. and M.S. degrees in electronics engineering from National Chiao-Tung University, Taiwan, R.O.C., in 2003 and 2005, respectively.

In 2006, he joined MediaTek Inc., Hsinchu, Taiwan, R.O.C., and develops digital TV backend related system. His major research interests include memory controller design, H.264/AVC video decoding, and associated VLSI architecture.

Bai-Jue Shieh was born in Taipei City, Taiwan, R.O.C., in 1974. He received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1996, and the M.S. and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1998 and 2001, respectively.

In 2001, he joined Sunplus Technology Company, Ltd, Hsinchu, Taiwan, R.O.C. His research interests include IC design flow, cell-based and fully custom VLSI design, video signal processing, system-on-chip design technology, cell library design, and memory circuit design.

Chen-Yi Lee (M’01) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1982, and the M.S. and Ph.D. degrees from Katholieke University Leuven (KUL), Leuven, Belgium, in 1986 and 1990, respectively, all in electrical engineering.

From 1986 to 1990, he was with IMEC/VSDM, working in the area of architecture synthesis for digital signal processing. In February 1991, he joined the faculty of the Electronics Engineering Department, National Chiao Tung University, Hsinchu, Taiwan, R.O.C., where he is currently a Professor. His research interests mainly include VLSI algorithms and architectures for high-throughput DSP applications. He is also active in various aspects of system-on-chip design technology, very low power designs, multimedia signal processing, and wireless communications. He served as the Director of Chip Implementation Center (CIC) from 2000 to 2003, an organization for IC design promotion in Taiwan. He also served as the Department Chair of Electronics Engineering, National Chiao Tung University (2003–2006).

Dr. Lee was the former IEEE Circuits and Systems Society (CAS) Taipei Chapter Chair (2000–2002), the SIP task leader of National SoC Research Program (2003–2005), and the microelectronics program coordinator of Engineering Division under National Science Council of Taiwan (2002–2005).