NESTED INTERLEAVING TRANSCODER FOR MPEG-4 SIMPLE PROFILE BITSTREAM

Jinwha Yang and Edward J. Delp

Video and Image Processing Laboratory (*VIPER*)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana 47907-1285
U.S.A.

ABSTRACT

We propose a nested interleaving scheme in order to provide three levels of synchronization in a compressed video bitstream for low data rate wireless applications. The first level of interleaving specifies the starting position of each macroblock(MB) in a video frame. The second level of interleaving provides the start position of each MB header codeword and the start position of each DCT blocks. The start position of each VLC in a DCT block is located in a known position by the third level of interleaving.

Our scheme is assumed to operate in the form of a transcoder, placed before and after the channel, of a MPEG-4 Simple Profile bitstream. Since the three level interleaving provides synchronization on a VLC scale, syntax-based bit error detections can be done in a VLC unit. The detected errors can be repaired syntactically so that the transcoder generates a MPEG-4 compliant bitstream which can then be decoded with a standard MPEG-4 decoder such as MoMuSys (FDIS V1.0).

1. INTRODUCTION

Video compression standards such as MPEG-4 [1] and H.263+ [2] specify the use of BCH (511,493) as the forward error-correcting codes (FEC) when the encoded bitstream is to be transmitted over a wireless channel in Annex H [3]. In addition to the FEC method, the standards describe other optional error-resilient tools that provide the framework of unequal error protection and reduce the error propagation inherent to any compression method with predictive coding [4, 5, 2]. Among them, *video packet resynchronization, data partitioning, reversible VLC* (RVLC) [4, 6] are known to be effective on bit reversal errors. To support the error-resilient tools, MPEG-4 and H.263+ rely on synchronization patterns (or "sync markers") with 23 leading zeros extensively. Frequent sync marker insertions add extra redundancy to the encoded bitstream. There is an interleaving algorithm known as EREC [7], which can rearrange variable length blocks into known positions so that synchronization is achievable with very small redundancy compared to the error-resilient tools using sync markers. As an application to a H.261-like bitstream, EREC was used to rearrange the MB header blocks and the DCT coefficient blocks into one data structure known as EREC slots [7]. In [8], we presented a modified interleaving scheme of EREC was introduced and a simple error handling scheme to repair detected bit errors was proposed. Both schemes were implemented as a transcoder to interleave MBs in a frame for a MPEG-4 Simple Profile bitstream.

In this paper we will further develop the interleaving scheme in a nested fashion to provide three levels of synchronization along with bit reversal error detection in the VLC codewords. We will define a trans-encoder as an operation that converts a standard bitstream to a stream in a private format for transmission. A trans-decoder is defined as an operation that reverts the received stream back to a standard bitstream. The term transcoder is a generic name to refer to both the trans-encoder and the trans-decoder. The nested interleaving scheme in this paper is not compliant with the MPEG-4 standard, hence it is implemented in the form of a transcoder.

2. NESTED INTERLEAVING

The nested interleaving operation on a compressed video stream will be described using the MPEG-4 Simple Profile bitstream. Unlike the interleaving scheme proposed in [8], which treats the entire MB bitstream as a codeword, the nested interleaving scheme treats each codeword as codewords whose code bits are to be interleaved at the second level and regards each MB as bundle of the codewords which are to be interleaved at the first level. The procedure is illustrated in Figure 1. The terms to be used in describing the

This work was supported by a grant from the Indiana 21^{st} Century Research and Technology Fund. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu.

operation are defined as follows:

- CBP_i : bitstream up to MCBPC in header section of the i^{th} MB
- H_i,T_i : header section except CBP_i or TCOEF section of the i^{th} MB
- K_{ij} : the i^{th} DCT block section of the T_j
- H_{ij},T_{ij} : the j^{th} codeword in H_i (or T_i)
- $C(H_i)$, $C(T_i)$: the count of the codewords in H_i (or T_i)
- $C(K_{ij})$: the count of the codewords in K_{ij}
- $L(CBP_i)$: the bit length of CBP_i
- $L(H_{ij})$, $L(T_{ij})$: the bit length of the j^{th} codeword in H_i (or T_i)
- $L_2(H_i)$, $L_2(T_i)$: the bit length of the i^{th} codeword in the buffer IB_1 (or IB_2) after first level interleave
- N_{MB} : the total number of MBs in a frame
- N_K : the total number of coded DCT blocks in a frame
- $C(H)_{avg}$, $C(T)_{avg}$: average count of codeword for the header (or the TCOEF) sections per a MB
- CBP_{avg} : average bit length of CBP_i per a MB
- *codeword slot* : a bit array that stores a given codeword bits
- $S_2(H)$, $S_2(T)$: total codeword slot counts in all the header (or TCOEF) section after first level interleave
- *codeword bank* : a group of codeword slot with predefined size
- BH_i, BT_i: the i^{th} codeword bank in IB_1 (or IB_2) to hold interleaved H_i (or T_i)
- IBC : the interleaving bit slot buffer for CBP
- IB_i : the interleaving buffer, where index i denote the interleave level and the contents of the buffer
- R_i^k : left over entities of the i^{th} unit after the k^{th} interleaving iteration step
- V_i^k : vacancy information of IB_i at the k^{th} interleaving iteration step.

The trans-encoder parses a frame of a MPEG-4 bitstream with the structure shown in Figure 1a. Then it stores each codeword in the MBs of the given frame individually as in Figure 1b except the frame header information, which is not interleaved by the trans-encoder. While parsing, it lists the CBP_i bits, the H_i and the T_i section boundaries and the K_{ji} boundaries in every MBs and counts the coded MB in a frame. One thing to note here is that the *mode and coded block pattern for chrominance*(MCBPC) and *coded block pattern*(CBP) [5] information in the H_i controls how other information in H_i and the following T_i should be decoded. Due to the fact, if the CBP_i, the H_{ij} and the T_{ik} are interleaved without distinction, it becomes impossible for the trans-decoder to de-interleave the MCBPC and CBPY information in the H_i. To make the de-interleaving possible, the CBP_i, the H_i and the T_i must be interleaved separately.

2.1. MCBPC interleaving

Since CBP_i must be decoded prior to other header information, it is interleaved with one level. Based on the N_{MB}, the average bit length for CBP_i is determined by

$$CBP_{avg} = \left\lceil \frac{1}{N_{MB}} \sum_{i=1}^{N_{MB}} L(CBP_i) \right\rceil . \quad (1)$$

The interleaving bit buffer IBC has N_{MB} slots with CBP_{avg} bits long. This one level interleaving procedure is identical to the second level of the following MB header interleaving procedure, except that IBC is used instead of IB_3.

2.2. MB header interleaving

MB header information except CBP_i is interleaved with two level to place the start of VLCs in the header on known positions. Based on $C(H_i)$ from the section boundary list and the N_{MB}, average codeword bank length is determined by

$$C(H)_{avg} = \left\lceil \frac{1}{N_{MB}} \sum_{i=1}^{N_{MB}} C(H_i) \right\rceil , \quad (2)$$

The interleaving buffer for the first level is known as IB_1. The IB_1 has N_{MB} multiples of BH and one BH holds $C(H)_{avg}$ codeword slots.

At the initial interleaving step with the iteration step $k = 0$, each H_i is placed into BH_i, using as many codeword slots as the bank BH_i can accommodate. If $C(H_i) = C(H)_{avg}$, the bank BH_i becomes full. If $C(H_i) < C(H)_{avg}$, BH_i can hold the entire H_i and leaves free space $V_i^0 = C(H)_{avg} - C(H_i)$ at the rear part of BH_i. If $C(H_i) > C(H)_{avg}$, the BH_i can accept only part of the codeword slots in the H_i, leaving $R_i^0 = (C(H_i) - C(H)_{avg})$ codeword slots from the H_i unassigned to the BH_i. After the initial step, the buffer IB has both full banks and partially full banks. There are still codeword slots that have not been placed into IB_1 unless $C(H_i) = C(H)_{avg}$ for all $i \in [1, N_{MB}]$.

For the next iterations, the iteration step k is increased. To fill the R_i^{k-1}slots left over from the H_i to the IB_1 at $k-1^{th}$ iteration, the next available bank BH_{i+k} is checked for vacancy $V_{i+k}^{k-1} > 0$. If not, the slot index i increases and the left over slots from the next H_i will be processed. If the $V_{i+k}^{k-1} > 0$, then entire R_i^{k-1} or up to V_{i+k}^{k-1} left over codeword slots are placed in the *reverse direction* [8] to the BH_{i+k} as shown in Figure 1c, leaving $R_i^k = 0$ or $R_i^{k-1} - V_{i+k}^{k-1}$ slots and reducing the vacancy $V_{i+k}^k = V_{i+k}^{k-1} - R_i^{k-1}$ or 0 for next placement. After this procedure is done for all

N_{MB}, if some codeword slots are still left, the interleaver repeats the above procedure until all the left over slots from the H_i are place into some vacant space in the IB_1. When the iteration step k becomes N_{MB}, the first level interleaving finishes(see Figure 1e).

The second level interleaving deals with the codeword bits. Now a separate bit interleaving buffer IB_3 for the header bits should be allocated. Average bit length for the buffer is given by

$$L(H)_{avg} = \left\lceil \frac{1}{S_2(H)} \sum_{i=1}^{N_{MB}} \sum_{j=1}^{C(H_i)} L(H_{ij}) \right\rceil, \quad (3)$$

where $S_2(H) = N_{MB} \times C(H)_{avg}$. The IB_3 consists of $S_2(H)$ bit arrays, each of which can store $L(H)_{avg}$ bits. We can obtain the interleaving procedure for the second level, by substituting { $S_2(H)$, codeword slot, codeword bits, $L_2(H_i)$, $L(H)_{avg}$ } terms respectively into the places of the { N_{MB}, codeword bank, codeword slots, $C(H_i)$, $C(H)_{avg}$ } terms in the first level procedure (see Figure 1f-g). Also, when the R_i^{k-1} left over bits are to be interleaved, they should be placed in the reverse direction.

2.3. DCT block interleaving

In order to place the DCT VLCs, the blocks and the MBs on predefined positions, codewords in DCT block are interleaved with three levels. Based on N_K, $C(K_{ji})$, N_{MB} and $C(T_i)$ from the section boundary list, the following parameters are defined in Equations 4-7.

At the first level, a block section buffer BS is formed to store M by N_{MB} block sections, where M is the average block count per MB. The block sections K_{ji} are interleaved into BS on section unit, similarly to the first level of the section 2.2. This interleaving may result in unoccupied sections in BS. Then the interleaving buffer IBK for the block codeword bank is defined to have $N_K - 1$ codeword banks with $C(K)_{avg}$ slots and the last codeword bank with $C(K)_{end}$ slots. The codeword slots in the occupied section of BS are interleaved into the IBK similar to the first level of the section 2.2 (see Figure 1d left).

$$C(K)_{avg} = \left\lceil \frac{N_{MB} \times C(T)_{avg}}{N_K} \right\rceil, \quad (4)$$

$$C(K)_{end} = (N_{MB} \times C(T)_{avg}) - ((N_K - 1) \times C(K)_{avg}), \quad (5)$$

$$C(T)_{avg} = \left\lceil \frac{1}{N_{MB}} \sum_{i=1}^{N_{MB}} C(T_i) \right\rceil, \quad (6)$$

$$L(T)_{avg} = \left\lceil \frac{1}{S_2(T)} \sum_{i=1}^{N_{MB}} \sum_{j=1}^{C(T_i)} L(T_{ij}) \right\rceil, \quad (7)$$

where $S_2(T) = N_{MB} \times C(T)_{avg}$.

At the second level, the codeword banks of IBK, indexed as IBK_{ji} ($j \in [1, M], i \in [1, N_{MB}]$), are interleaved into a separate buffer IB_2 consisting of BT_i with $C(T)_{avg}$ slots (see Figure 1d right). Since both BT_i and IBK_{ji} have fixed length slots, the interleaving takes one iteration. Following a sequential order of i, IBK_{ji} is placed into BT_i. When $j = M$, the bank IBK_{Mi} will be placed close to the end part of BT_i. If there are slots carried over from $IBK_{M(i-1)}$, the slots are placed before the IBK_{Mi} bank. If slots of IBK_{Mi} are to be placed beyond the end of BT_i, they are carried over to BT_{i+1}. When $i = N_{MB}$, the interleaving with IB_2 is complete.

At the third level, the interleaving of the VLC bits is identical to the second level of MB header interleaving. The VLCs in IB_2 are interleaved into IB_4, which consists of $S_2(T)$ bit arrays with $L(T)_{avg}$ bits capacities.

2.4. Final bitstream

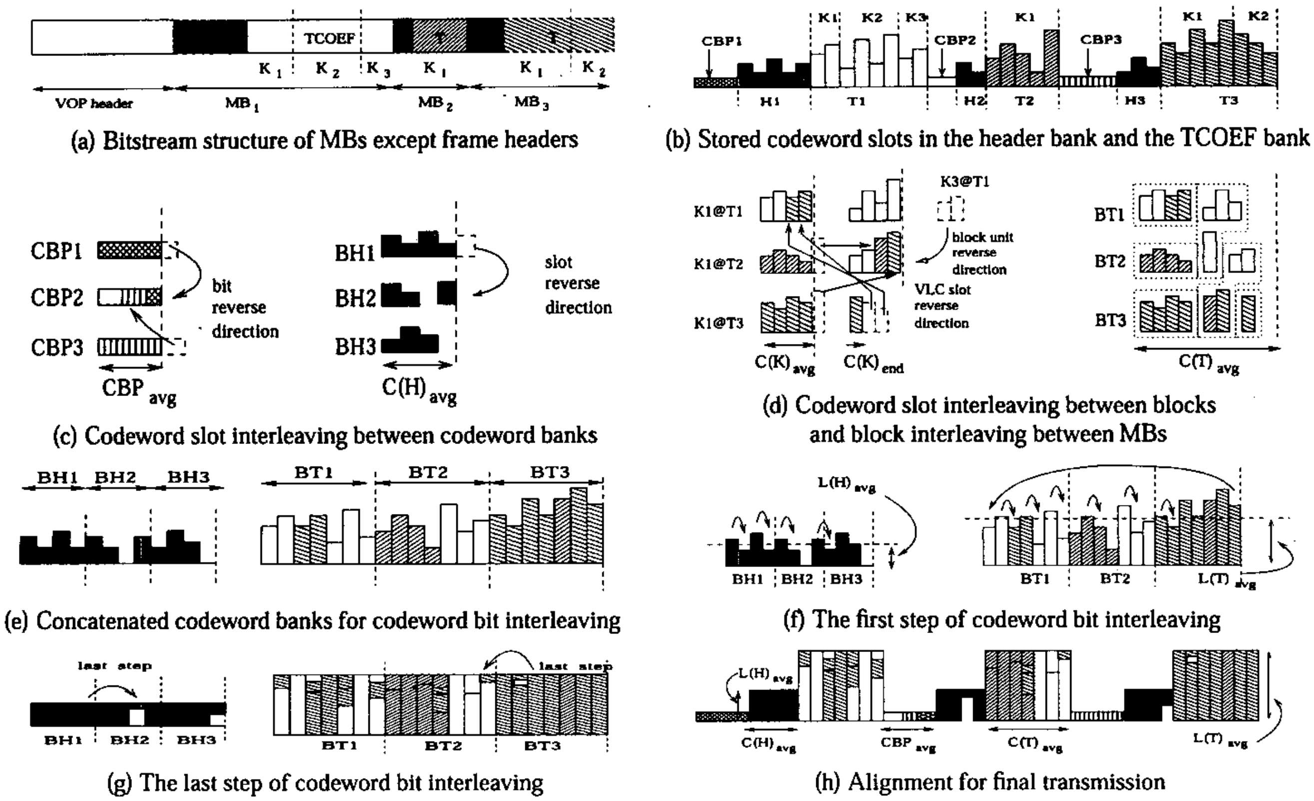
To complete the interleaving, a total $N_{MB} \times CBP_{avg} + S_2(H) \times L(H)_{avg} + S_2(T) \times L(T)_{avg}$ bits are necessary to place the $\sum L(CBP_i) + \sum\sum L(H_{ij}) + \sum\sum L(T_{ij})$ bits into the interleaving buffers IBC, IB_3 and IB_4. The incurred redundancy in bits is the difference of the above two terms.

Finally, the trans-encoder concatenates the information { N_{MB}, CBP_{avg}, $C(H)_{avg}$, $C(T)_{avg}$, $L(H)_{avg}$, $L(T)_{avg}$, N_K} to the previously parsed frame header field bitstream, forming a new frame header field for the trans-decoder. After the new frame header, the bit array buffers IBC, IB_3, IB_4 shown in Figure 1h are concatenated in bit serial order. In case the trans-encoded bitstream does not end on a byte boundary, the trans-encoder pads stuffing bits to ensure that the next frame sync marker starts on a byte boundary.

3. CONCLUSION

We have presented a three level nested interleaving transcoder scheme for the delivery of compressed video bitstream over error-prone channels. The main effect of the scheme is to provide a codeword synchronization on each VLC unit so that the propagation of error can be reduced. Also the incurred redundancy is very small compared to those incurred by the ER tools provided in MPEG-4 and H.263+. Advantages of the scheme are:

- Start positions of the MBs, the blocks and the VLCs in a frame are aligned on known positions without using a "sync word."
- The effect of error in a VLC is confined to a small number of VLCs which are interleaved together with the VLC having error.



(a) Bitstream structure of MBs except frame headers

(b) Stored codeword slots in the header bank and the TCOEF bank

(c) Codeword slot interleaving between codeword banks

(d) Codeword slot interleaving between blocks and block interleaving between MBs

(e) Concatenated codeword banks for codeword bit interleaving

(f) The first step of codeword bit interleaving

(g) The last step of codeword bit interleaving

(h) Alignment for final transmission

Fig. 1. Example of the nested interleaving scheme with $N_{MB} = 3$.

- DCT blocks can be decoded independent of the relation between the block and the MB it belongs to. This fact can be used as a correctness check for the MCBPC and CPB information.
- As a side effect of the nested interleaving, partitioning of logical groups such as the MCBPC, the MB header and the DCT block is established automatically.
- Our method from "end-to-end" is compliant with the standards.

4. REFERENCES

[1] Signal Processing: Image Communication, *Special Issue on MPEG-4*, 2000 (entire issue devoted to MPEG-4).

[2] J. Ott S. Wegner, G. Knorr and F. Kossentini, "Error resilience support in H.263+," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849–866, November 1998.

[3] ITU-T, *Video Coding for Low Bit Rate Communication: ITU-T Recommendation H.263*, January 1998.

[4] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communication Magazine*, pp. 112–119, June 1998.

[5] ISO/IEC JTC1/SC29/WG11 N3515, Beijing, *MPEG-4 Video Verification Model version 17.0*, July 2000.

[6] M. Wada Y. Takishima and H. Murakami, "Reversible variable length codes," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 158–162, 1995.

[7] D. W. Redmill and N. G. Kingsbury, "The EREC: An error-resilient technique for coding variable-length blocks of data," *IEEE Transactions on Image Processing*, vol. 5, no. 4, pp. 565–574, April 1996.

[8] J. Yang and E. J. Delp, "A *MPEG*-4 Simple Profile transcoder for low data rate wireless application," in *Proceedings of the Visual Communications and Image Processing Conference*, San Jose, California, January 21-23 2002, vol. 4671, pp. 112–123.