中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-07-005

Distributed Video Coding Based on Coding Mode-aided Motion Compensation and Robust Mdeia Hashing

Li-Wei Kang (康立威) and Chun-Shien Lu (呂俊賢)

February 27, 2007 || Technical Report No. TR-IIS-07-005
http://www.iis.sinica.edu.tw/LIB/TechReport/tr2007/tr07.html

Distributed Video Coding Based on Coding Mode-aided Motion Compensation and Robust Media Hashing[+]

Li-Wei Kang (康立威) and Chun-Shien Lu (呂俊賢)[*]
Institute of Information Science, Academia Sinica
Taipei, Taiwan 115, R.O.C

Abstract

To meet the requirement of distributed video coding (DVC) in resource-limited sensor networks, Wyner-Ziv theorem-based source coding with side information only available at the decoder states that an intraframe encoder with interframe decoder system can achieve comparable coding efficiency of a conventional interframe encoder and interframe decoder system. In this paper, firstly, a block discrete cosine transform (DCT)-based Wyner-Ziv video codec with coding mode-aided motion compensation at the decoder is proposed, denoted by "ProposedDVC1." The major characteristic is that for motion compensation at the decoder, side information generation and error correcting code (ECC) decoding are jointly performed to find the final side information. Similar to most existing Wyner-Ziv video coding systems, "ProposedDVC1" is with light encoder and heavy decoder. However, in some video communication scenarios, low complexity in both the encoder and decoder is required. In this study, another Wyner-Ziv video codec based on robust media hashing without needing to perform motion estimation at both the encoder and decoder is proposed, denoted by "ProposedDVC2." The particular contribution of ProposedDVC2 is its low complexity in both the encoder and decoder. Simulation results demonstrate the achievable coding efficiency of ProposedDVC2 is comparable with that of ProposedDVC1 while the complexity of ProposedDVC2 is much lower than that of ProposedDVC1. In addition, both ProposedDVC1 and ProposedDVC2 need no feedback channel.

Keywords: distributed video coding, Wyner-Ziv video coding, motion compensation, media hash, video communication, video sensor network.

+This work was supported in part by the National Science Council of Taiwan, R.O.C., under Grant NSC 95-2221-E-001-022. Preliminary results of this paper were published in *ICIP2006* and *MMSP2006*.

*Further author information: Send correspondence to E-Mail: lcs@iis.sinica.edu.tw, Tel: 886-2-2788-3799 X 1513.

I. INTRODUCTION

A. Background

Conventional hybrid predictive video compression standards, such as MPEG-4 and H.264/AVC [1]-[2], usually perform motion estimation/compensation among successive frames for interframe predictive coding so that the encoder is typically 5 to 10 times more complex than the decoder [3]-[4]. However, such a heavy encoder with light decoder video coding system is usually suitable for video broadcasting or video streaming applications (*e.g.*, video on demand system) where video is encoded once and decoded many times. With the advancement of emerging applications (*e.g.*, wireless video sensor networks and wireless mobile video communication), the current video coding paradigm is insufficient if some new requirements, such as the restrictions on computational capability and memory for a low power video encoder device, are considered. In fact, this calls for a new video coding paradigm with low-complexity encoder.

To meet this requirement, distributed video coding (DVC) based on the Wyner-Ziv information theorem for lossy compression [5] has recently become an emerging video coding paradigm [3]-[4], where individual frames are encoded independently (intraframe coding) but decoded conditionally (interframe decoding). The Wyner-Ziv information theorem is originally extended from the Slepian-Wolf information theorem for lossless compression [6]. In contrast to conventional hybrid predictive video coding, Wyner-Ziv video coding usually performs intraframe encoding without performing motion estimation at the encoder, whereas it performs interframe decoding with motion estimation/compensation or complex frame interpolation at the decoder. That is, part of the computational burden (*e.g.*, motion estimation) is shifted from encoder to decoder and results in a video codec with light encoder and possibly heavy decoder.

A general framework of a Wyner-Ziv video codec is shown in Fig. 1. At the encoder, an input video sequence is divided into key frames and Wyner-Ziv frames. Each key frame (K) is encoded using a conventional intraframe encoder (*e.g.*, H.263 intraframe coding [7]-[16], H.264/AVC intraframe coding [17]-[20]) while each Wyner-Ziv frame (W) is encoded using a distributed video encoder to generate Wyner-Ziv bits. On the other hand, the encoder can optionally transmit some extra information to the decoder to help side information generation. At the decoder, each key frame is decoded using the conventional intraframe decoder. For a Wyner-Ziv frame, it is decoded using the distributed video decoder with the assistance of side information. Side information can be generated using any previous decoded frames and/or the extra information transmitted from the encoder. The decoder can request more Wyner-Ziv bits from the encoder via a feedback channel optionally dependent on current decoding efficiency.

More specifically, most existing DVC schemes [3]-[4], [7]-[18] modeled Wyner-Ziv video coding as a channel coding problem. The statistical dependence between two correlated sources W and Y is modeled as a virtual correlation channel analogous to binary symmetric channel (BSC) or additive white Gaussian noise (AWGN) channel. The side information Y is viewed as a noisy version of the source W. At the encoder, the compression of W can be achieved by transmitting only parity bits derived from error correcting codes (ECC) (*e.g.*, turbo codes [7]-[10], [13]-[18]). Here, the parity bits form the Wyner-Ziv bits. The size of the transmitted Wyner-Ziv bits is usually smaller than that of the original source W. The decoder concatenates the received parity bits with the side information Y and performs error correction decoding to correct some "errors" in Y, *i.e.*, the noisy version of the source W, for the reconstruction of W. The realization of such a channel coding approach for Wyner-Ziv video coding can be divided into two categories, *i.e.*, pixel-domain Wyner-Ziv video coding and transform-domain Wyner-Ziv video coding. They are briefly described in the following two subsections.

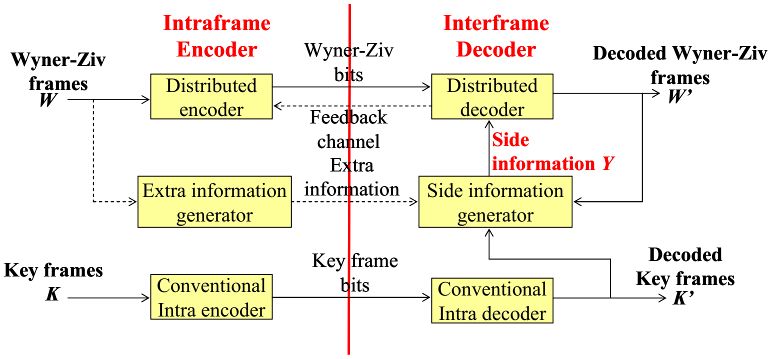


Fig. 1. General framework of a Wyner-Ziv video codec.

B. Pixel-domain Wyner-Ziv Video Coding

Aaron, Setton, and Girod [7] proposed the first pixel-domain Wyner-Ziv video codec. At the encoder, an input video sequence is divided into key frames and Wyner-Ziv frames. Each key frame (K) is encoded using a conventional intraframe encoder (*e.g.*, H.263 intraframe coding) while each Wyner-Ziv frame (W) is encoded using a Wyner-Ziv intraframe encoder to generate Wyner-Ziv bits. For each Wyner-Ziv frame, W, each pixel value is quantized using a uniform scalar quantizer with 2^M intervals to form the quantized symbol stream q. Then, q is fed into a turbo encoder to form the parity bits (Wyner-Ziv bits) stored in a buffer. The buffer transmits a subset of the parity bits to the decoder upon request.

At the decoder, each key frame is decoded using a conventional intraframe decoder. For each Wyner-Ziv frame, the decoder generates the side information (Y) by interpolation or extrapolation of previously decoded key frames and, possibly, previously decoded Wyner-Ziv frames. To exploit the side information, the decoder assumes a statistical dependency model between W and Y. The turbo decoder combines the side information Y and the received parity bits to recover the symbol stream q'. If the decoder cannot reliably decode the original symbols, it requests additional parity bits from the encoder buffer through feedback until an acceptable probability of symbol error is reached. The decoder usually needs to request $r \leqq M$ bits to decode which of the 2^M bins a pixel belongs to and, hence, compression is achieved. After decoding q', the decoder reconstructs the Wyner-Ziv frame W' as follows. If the side information Y is within the reconstructed bin, the reconstructed pixel will take a value very close to the side information. Otherwise, the function clips the reconstruction towards the boundary of the bin closest to *Y*. In addition, based on this video coding paradigm [7], several pixel-domain Wyner-Ziv video coding schemes were similarly proposed [8]-[10], [17].

C. Transform-domain Wyner-Ziv Video Coding

Puri and Ramchandran [11]-[12] proposed the first transform-domain Wyner-Ziv video codec, called "Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding (PRISM)." In PRISM, a key frame is encoded and decoded using the H.263 intraframe codec. A Wyner-Ziv frame is transformed using block DCT followed by uniform scalar quantization. For each block, the lower-frequency coefficients are compressed using a syndrome encoder. The higher-frequency coefficients are conventionally entropy-encoded. The encoder also sends a cyclic redundancy check (CRC) of the quantized coefficients. The decoder performs motion compensation to generate side information. The syndrome decoder combines the side information and the syndrome bits to recover the symbol stream. Finally, the decoder can reconstruct the Wyner-Ziv frame based on the symbol stream and the side information, followed by inverse DCT. On the other hand, Aaron, Rane, Setton, and Girod [13] proposed a transform-domain Wyner-Ziv video codec, modified from their pixel-domain codec in [7]. Similarly, based on this video coding paradigm [13], several transform-domain Wyner-Ziv video coding schemes were also proposed [14]-[16], [18].

D. Overview of the Proposed Wyner-Ziv Video Coding Schemes

In this paper, firstly, a DCT-based Wyner-Ziv video codec with coding mode-aided motion compensation at the decoder is proposed, denoted by "ProposedDVC1." The major characteristics include: (a) for each block, a large amount of candidate blocks are evaluated based on some criteria derived from Reed-Solomon (RS) decoding and best neighborhood matching to find the best candidate block as the side information; (b) ECC decoding is applied to participate in generating side information; (c) no feedback channel is required. In most existing Wyner-Ziv video codecs, the decoder generates side information firstly without considering ECC decoding, and then concatenates the parity bits and the side information to perform ECC decoding. If the decoding result is unacceptable, the decoder will request more parity bits via the feedback channel. Hence, we observe that (a) the generated side information may be not the best one for ECC decoding; (b) requesting more bits via the feedback channel indeed induces some network overheads [10]; (c) the feedback channel may be not always available. In ProposedDVC1, side information generation and ECC decoding are jointly performed without requesting any information via the feedback channel. For each side information candidate, the decoder will perform ECC decoding and check some criteria, and the best one will be selected as the final side information.

Most existing Wyner-Ziv video codecs [3]-[4], [7]-[16], [18] and ProposedDVC1 are with light encoder and heavy decoder because the encoder performs only intraframe coding and the decoder performs some complex interframe decoding operations. Such a Wyner-Ziv codec is only suitable for the scenario where the decoder can support high computational capability. However, if both encoder and decoder are required to be with low-complexity (*e.g.*, wireless video communication between a pair of mobile camera phones), an alternative solution can be described as follows [3]-[4]. A mobile camera phone captures and encodes the video using a Wyner-Ziv encoder and transmits the compressed bitstream to a base station in the network. The base station supports a transcoder consisting of a Wyner-Ziv decoder and a standard encoder (*e.g.*, MPEG or H.264/AVC encoder). The transcoder can decode the received Wyner-Ziv bitstream and re-encode it to a standard bitstream, which will be transmitted to the receiver with a low-complexity standard decoder for real-time decoding. That is, based on the transcoder supported in a network infrastructure, each device can have a light encoder (Wyner-Ziv encoder) and a light decoder (standard decoder).

To make a Wyner-Ziv video codec be directly applicable to the scenario without additional transcoder support. In this paper, another Wyner-Ziv video codec based on robust media hashing is proposed, denoted by "ProposedDVC2." The key is that the significant differences between a video frame and its reference frame are efficiently extracted and used for frame recovery based on robust image hashing without needing to perform motion estimation/compensation at both the encoder and decoder. The particular contribution of ProposedDVC2 is its low complexity in both the encoder and decoder. In addition, feedback channel is also not exploited in ProposedDVC2.

Conceptually, the two proposed Wyner-Ziv video codecs are with intraframe encoding and interframe decoding. Strictly speaking, the proposed encoders are no longer pure intraframe encoders due to some simple and efficient comparisons that will be applied at the encoders. The remainder of this paper is organized as follows. The proposed DCT-based Wyner-Ziv video codec with coding mode-aided motion compensation at the decoder (ProposedDVC1) is described in Sec. II. The proposed Wyner-Ziv video codec based on robust media hashing (ProposedDVC2) is described in Sec. III. Simulation results are presented in Section V, followed by conclusions and future works.

II. Proposed Distributed Video Coding with Coding Mode-aided Motion Compensation

The proposed block-DCT based Wyner-Ziv video codec (ProposedDVC1) is shown in Fig. 2. In ProposedDVC1, an input video sequence is divided into several GOPs (group of pictures), in which a GOP consists of a key frame followed by several Wyner-Ziv frames. For a Wyner-Ziv frame, the key is to find the best side information block, s_i, for each block, b_i.

A. Problem Formulation

In conventional video coding (*e.g.*, H.264/AVC), for each $N \times N$ block, b_i, in an inter-coded frame (*e.g.*, P frame), the encoder will perform motion estimation to find the best match block, s_i, in the reference frames as follows:

$$s_i = \arg\min_{s_i \in SW_i} d\left(b_i, s_i\right), \tag{1}$$

where SW_i denotes possible search windows in the reference frames for b_i, and

$$d(b_i,s_i)=\frac{1}{N\times N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}\left|b_i(x,y)-s_i(x,y)\right|, \tag{2}$$

where $b_i(x, y)$ and $s_i(x, y)$ are pixel values of blocks b_i and s_i, respectively.

However, in distributed video coding, the encoder cannot perform motion estimation, which should be shifted to the decoder. Hence, the problem here is formulated as follows.

Problem 1 (side information block generation for Wyner-Ziv frame reconstruction): For each $N\times N$ block b_i in a Wyner-Ziv frame, we want to find the best match side information block, s_i, to satisfy Eq. (1) at the decoder without performing motion estimation at the encoder.

The obtained side information s_i will be used to reconstruct b_i. Usually, the better the side information s_i is, the better the reconstructed b_i will be. However, s_i cannot be accurately obtained at the encoder without performing motion estimation, and the original b_i is unavailable at the decoder. Hence, it is difficult to solve Eq. (1) accurately at the decoder. In this section, a coding mode-aided motion compensation scheme at the decoder is proposed to find the best s_i for reconstruction of b_i. It is expected that the obtained side information, s_i, can be as accurate as that obtained using Eq. (1) at the encoder. The accuracy of the obtained side information (*i.e.*, motion vectors) using ProposedDVC1 will be analyzed later.

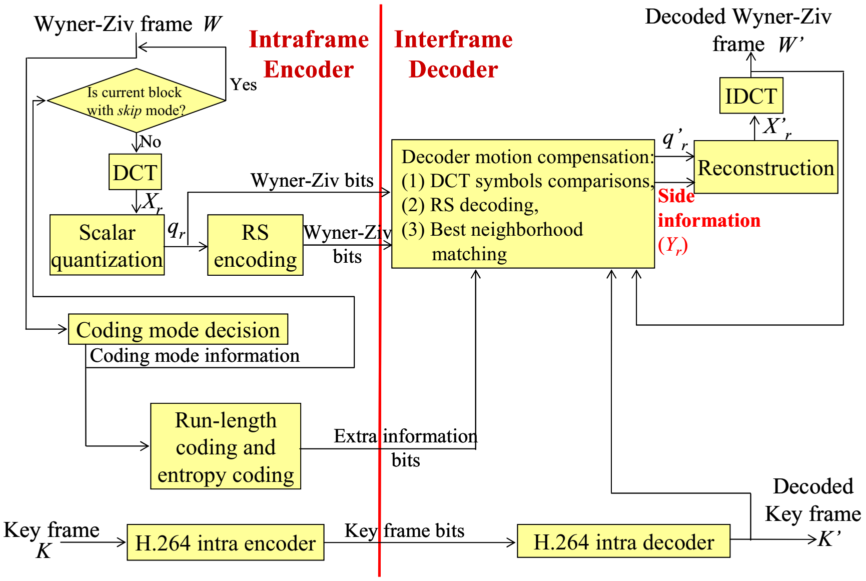


Fig. 2. The proposed Wyner-Ziv video codec (ProposedDVC1).

B. Proposed Wyner-Ziv Video Encoder in ProposedDVC1

At the encoder, each key frame (K) is encoded using the H.264/AVC intraframe encoder [2]. On the other hand, each Wyner-Ziv frame (W) is divided into several non-overlapping $N\times N$ blocks. First, the coding mode for each block in a Wyner-Ziv frame will be decided based on the estimated motion activity as follows. Here, the original previous frame will be stored in the encoder buffer. For each block, b_i, in the current frame, the difference, d_i, between b_i and the co-located block, f_i, in the previous frame is calculated using Eq. (2) as $d_i = d(b_i, f_i)$. If $d_i \leq T_1$, the coding mode of b_i is declared to be *skip* mode. If $T_1 < d_i \leq T_2$, the coding mode of b_i is declared to be *non-skip with RS coding* mode. Otherwise, the coding mode of b_i is declared to be *non-skip without RS coding* mode. Here, T_1 and T_2 are two predefined positive thresholds and $T_1 < T_2$. The extra overhead in the encoder is a buffer with the size the same as that of an uncompressed frame. Here, the coding mode information for each Wyner-Ziv frame will be encoded using the run-length coding followed by the entropy coding to form the extra information bits.

After performing block coding mode decision, each block in a Wyner-Ziv frame can now be encoded. For a block with *skip* mode, no data will be encoded. For a block with *non-skip* mode, similar to [13], a block DCT will be performed followed by a scalar quantization to obtain a symbol block containing $N \times N$ symbols. The four employed quantizers are shown in Fig. 3 with $N = 4$. For example, if the quantizer shown in Fig. 3(a) is used, the DC value will be quantized to a symbol with at most 64 levels (denoted by 6 bits).

64	32	16	4
32	16	4	4
16	4	4	0
4	4	0	0

(a)

32	16	8	4
16	8	4	4
8	4	4	0
4	4	0	0

(b)

32	8	0	0
8	0	0	0
0	0	0	0
0	0	0	0

(c)

16	8	0	0
8	0	0	0
0	0	0	0
0	0	0	0

(d)

Fig. 3. Four scalar quantizers used in ProposedDVC1.

For a block with *non-skip with RS coding* mode, the three most important symbols corresponding to the three lowest frequency DCT coefficients will be encoded directly. The remaining symbols will be encoded using (u, v) *RS* codes with z bits for each [21] to generate parity symbols, where v is the number of DCT subbands having non-zero quantization levels. Only RS parity symbols will be encoded. For example, if a 4×4 DCT block is quantized using the quantizer shown in Fig. 3(a), the three most important symbols will be encoded directly with 6 bits, 5 bits, and 5 bits, respectively. The remaining 10 non-zero symbols denoted by 4 bits ($z = 4$) for each can be encoded using (14, 10) RS code to generate 4 parity symbols. That is, there are, in total, 4×4 = 16 parity bits required. In Fig. 3(a), although some symbols with a quantization level of 4 can be denoted by 2 bits, in order to use a common RS code to encode all the remaining 10 non-zero symbols, each of them can be denoted by 4 bits by simply adding two zero bits. In this case, a block is totally encoded with 32 (= 6 + 5 + 5 + 16) bits. On the other hand, for a block with *non-skip without RS coding* mode, all the symbols will be encoded directly. Usually, this kind of block is rare. Finally, the resultant encoded symbols for all the blocks with *non-skip* mode in a Wyner-Ziv frame constitute the Wyner-Ziv bits. The key frame bits, the Wyner-Ziv bits, and the extra information bits will be transmitted to the decoder.

C. Proposed Wyner-Ziv Video Decoder in ProposedDVC1

At the decoder, each key frame will be decoded using the H.264/AVC intraframe decoder. For a

Wyner-Ziv frame, the coding mode information will be decoded first, and then all the blocks with *skip* mode will be reconstructed by assigning the co-located blocks of the previous reconstructed frame. On the other hand, for each block with *non-skip* mode, the proposed coding mode-aided motion compensation scheme is employed to find the corresponding side information. First, the search windows in the previous reconstructed frames are formed so that each block in the search windows will be a candidate side information block. Then, similar to the encoder operations, the $N\times N$ DCT followed by the scalar quantization will be applied to each candidate side information block. In addition, the reconstructed 8-connected neighboring blocks for each candidate side information block will be also extracted. The decoding of blocks with non-skip mode can be divided into two cases discussed below.

Case 1: For a block, b_i, with *non-skip with RS coding* mode, each candidate block, c_k, in the search window(s) will be evaluated. The best candidate block satisfying the following three criteria will be selected to be the side information for b_i.

(a) The difference between the three most important symbols of b_i and those of c_k should be minimized. Let $(DCT_{bi})_j$ and $(DCT_{ck})_j$, $j = 1, 2, 3$, be the representative values for the quantization levels which the three most important symbols belong to, respectively. The difference between the three most important symbols of b_i and those of c_k, $DCT\_Diff_i(b_i, c_k)$, is defined as

$$DCT\_Diff_i(b_i, c_k) = \sum_{j=1}^{3} \left| (DCT_{bi})_j - (DCT_{ck})_j \right|. \tag{3}$$

(b) The number of incorrect RS-decoded symbols (denoted by $N_{RSi}(b_i, c_k)$), returned from the RS-decoder based on the parity symbols of b_i and the symbols (except the three most important symbols) of c_k should be minimized. For a (u, v) RS code, the number of allowable maximum symbol errors is $(u - v) / 2$. If the RS code cannot correct all the error symbols, *i.e.*, the number of incorrect RS-decoded symbols exceeds the maximum allowed, $N_{RSi}(b_i, c_k)$ is set to $(u - v) / 2 + 1$.

(c) The difference between the 8-conncected neighboring blocks of b_i and those of c_k should be minimized. Let b_{ij} and c_{kj} be the 8-conncected neighboring reconstructed blocks of b_i and c_k, respectively, $j = 1, 2, \ldots, N_{recon}$, where $N_{recon} \le 8$ is the number of the 8-conncected reconstructed neighboring blocks of b_i. The difference, NB_Diff_i, between the 8-conncected neighboring blocks of b_i and those of c_k is defined as

$$NB\_Diff_i(b_i, c_k) = \sum_{j=1}^{N_{recon}} d_i(b_{ij,} c_{kj}), \tag{4}$$

where d_i function is similarly defined in Eq. (2).

Combining the above three criteria, a fitness function F_i for generating the side information for b_i is defined as

$$F_i(b_i, c_k) = w_a \times DCT\_Diff_i(b_i, c_k) + w_b \times N_{RSi}(b_i, c_k) + w_c \times NB\_Diff_i(b_i, c_k), \tag{5}$$

where w_a, w_b, and w_c are the weighting coefficients for the three terms, respectively. Finally, the side information block, s_i, in the search window(s), SW_i, for the block b_i, minimizing the fitness function F_i is selected to be the side information for b_i, *i.e.*,

$$s_i = \arg \min_{c_k \in SW_i} F_i(b_i, c_k). \tag{6}$$

An illustrated example is shown in Fig. 4, where the light-colored blocks have already been reconstructed, and the non-zero number for each symbol denotes the quantization index which the symbol belongs to. The blocks with *non-skip* mode in a Wyner-Ziv frame are reconstructed in a raster-scan order (all the blocks with *skip* mode can be reconstructed first).

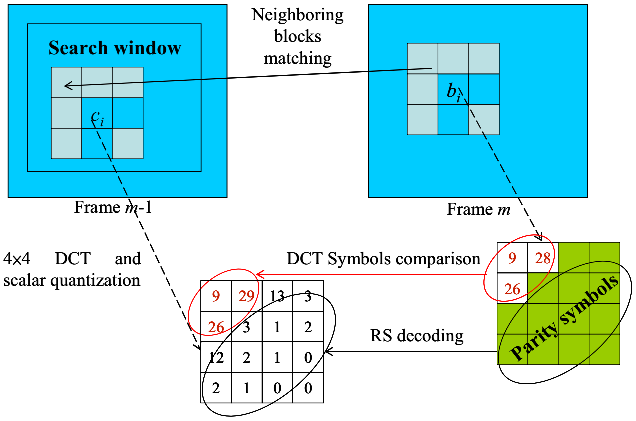


Fig. 4. An example of motion compensation at decoder for a block, b_i, with *non-skip with RS coding* mode.

Case 2: For a block, b_i, with *non-skip without RS coding* mode, two similar criteria are employed: (a) The difference between all the symbols of b_i and those of c_i should be minimized; and (b) the difference between the 8-conncected neighboring blocks of b_i and those of c_i should be minimized.

After the side information (the best candidate block) for a block with *non-skip* mode is obtained, it will help for reconstructing the block. For a block with *non-skip with RS coding* mode, the RS-decoded symbols coming from the side information will be assigned to the block. Then, similar to [13], the DCT coefficients, X_r, will be reconstructed as follows. For a symbol, q_r', in a block, if the corresponding DCT coefficient, Y_r, in the side information is within the coefficient interval denoted by q_r', the symbol, q_r', will be dequantized to Y_r; otherwise, the boundary of the quantization interval that is nearest to Y_r is used to reconstruct q_r'. For a block with *non-skip without RS coding* mode, a similar strategy is employed to reconstruct all the DCT coefficients. Finally, the inverse DCT is applied to the reconstructed DCT block to obtain the pixel block.

D. Computational Complexity of ProposedDVC1

The computational complexity of ProposedDVC1 is dominated by those of the DCT and RS encoding, and is similar to that of a conventional intraframe encoder consisting of the DCT and entropy coding. However, the computational complexity of the proposed Wyner-Ziv decoder is very heavy. The decoder performs very complex motion-compensation operations based on the criteria defined above for each block with *non-skip* mode. Hence, similar to most existing Wyner-Ziv video codecs, ProposedDVC1 is with light encoder and heavy decoder. On the other hand, at the decoder, each Wyner-Ziv frame can be decoded depending on only previous reconstructed frame(s), hence no decoding delay (no out-of-order decoding) will be induced.

E. Performance Analysis for ProposedDVC1

In conventional block-based video coding, motion vector plays the most critical role. Accurate motion vectors will usually induce fewer residual data and better coding efficiency. Hence, conventional video coding usually performs complex motion estimation at the encoder. However, in ProposedDVC1, the motion estimation operation is shifted to the decoder and, unavoidably, the obtained motion vectors cannot be as accurate as those obtained using an exhaustive search at the encoder. Here, similar to the analysis for motion vector in [22], we will analyze the accuracy of the motion vectors obtained at the decoder using ProposedDVC1 via a new metric called maximum motion vector signal-to-noise ratio (MSNR).

For a block b_i in a frame, i = 1, 2, …, n_b, where n_b is the number of the blocks in the frame, the true motion vector and the estimated motion vector for b_i is denoted by $MV_{bi} = (x_i, y_i)$ and $\overset{\wedge}{MV}_{bi}$ = ($\hat{x}_i$, $\hat{y}_i$), respectively. Then, the mean square error (MSE) motion vector, $\Delta^2 = \left(\Delta^2{}_x, \Delta^2{}_y\right)$, for a frame between the true motion vectors and the corresponding estimated ones is defined as

$$\Delta^2{}_x = \frac{1}{n_b}\sum_{i=1}^{n_b}\left(x_i - \hat{x}_i\right)^2 \text{, and} \tag{7}$$

$$\Delta^2{}_y = \frac{1}{n_b}\sum_{i=1}^{n_b}\left(y_i - \hat{y}_i\right)^2 . \tag{8}$$

Then, MSNR is defined as

$$MSNR = 10\log_{10}\frac{MV^2{}_{\max,x} + MV^2{}_{\max,y}}{\Delta^2{}_x + \Delta^2{}_y}, \tag{9}$$

where ($MV_{\max,x}$, $MV_{\max,y}$) is the maximum possible motion vector.

Here, for a video sequence, its true motion vectors are obtained by means of a block-based exhaustive search. On the other hand, the two kinds of estimated motion vectors are, respectively, obtained performing H.264/AVC full search at the encoder with quarter-pixel accuracy and the decoder motion estimation in ProposedDVC1. Due to the fact that motion estimation in ProposedDVC1 is with integer-pixel accuracy, for simplicity, the true motion vectors also use integer-pixel accuracy. For H.264/AVC interframe coding with quarter-pixel accuracy, only the motion vectors with integer-pixel accuracy are extracted to be the estimated motion vectors. Then, for a sequence, the *MSNR* values for all frames based on the two kinds of estimated motion vectors are calculated and averaged to obtain the *MSNR* value for the sequence. The *MSNR* results for the *Carphone*, *Hall monitor*, and *Salesman* sequences with different GOP sizes (*GOPSize*) are, respectively, shown in Figs. 5-7.

It can be observed from Figs. 5-7 that the MSNR performance gaps between ProposedDVC1 and H.264/AVC interframe coding are from 0.1 to 0.4 dB. For fast-motion sequences (*e.g.*, *Carphone*), higher bitrates can allow more blocks with *non-skip* mode to be used so as to produce larger MSNR value in ProposedDVC1. Therefore, the MSNR gap between H.264/AVC and ProposedDVC1 is smaller for higher bitrates. For slow/middle-motion sequences (*e.g.*, *Hall monitor* and *Salesman*), most blocks are with zero or small motion vectors, and too many blocks with *non-skip* mode are meaningless. Hence, in cases with higher bitrates, more bits cannot be efficiently used for motion estimation, and *MSNR* gaps will be larger. On the other hand, for smaller GOP sizes (*e.g.*, *GOPSize* = 2), the motion estimation for H.264/AVC interframe coding is less efficient than that with larger GOP sizes. Hence, the *MSNR* gaps can become smaller when the bitrate increases. On the contrary, for larger GOP sizes (*e.g.*, *GOPSize* = 8), the motion estimation for H.264/AVC interframe coding is efficient. Hence, the *MSNR* gaps will become larger when the bitrate increases. Based on the analysis for the motion vector accuracy, the performance for ProposedDVC1 can be evaluated. The complete evaluation of rate-distortion (RD) performance is in Section IV.

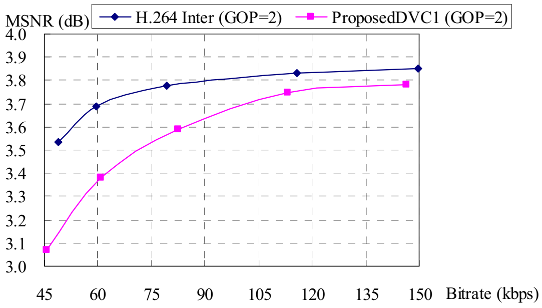


Fig. 5. MSNR performance for the *Carphone* sequence with *GOPSize* = 2.

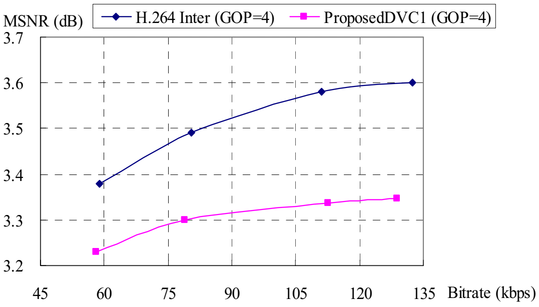


Fig. 6. MSNR performance for the *Hall monitor* sequence with *GOPSize* = 4.

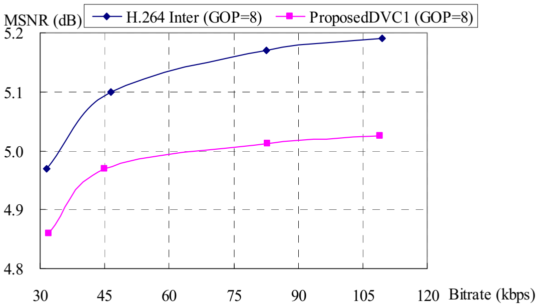


Fig. 7. MSNR performance for the *Salesman* sequence with *GOPSize* = 8.

III. Proposed Distributed Video Coding Based on Robust Media Hashing

In this section, a new Wyner-Ziv video codec based on robust media hashing is described, denoted by "ProposedDVC2." In ProposedDVC2, no motion-compensated interpolation/extrapolation is performed at both the encoder and the decoder, no ECC is applied, and no feedback channel is required. In particular, the key characteristic is that both the encoder and the decoder are with low-complexity.

A. Problem Formulation

In view of the fact that image hashing [23]-[24] is able to capture the essence of an image (or a video frame) while reducing storage requirement, ProposedDVC2 is presented by exploiting hash modification to achieve Wyner-Ziv frame recovery without needing motion estimation. This problem is associated with the proper selection of the length of an image (or frame) hash under the constraint of trade-off between visual quality and coding efficiency. The problem can be defined as follows.

Problem 2 (Media hashing for Wyner-Ziv frame reconstruction): For a Wyner-Ziv frame, W, its most significant features, extracted by comparing the hash values of W and those of its reference frame I, should be properly selected such that

$$PSNR(W, \hat{W}) \geq \text{desired PSNR value, and} \tag{10}$$

$$PSNR(W, \hat{W}) >> PSNR(I, \hat{W}), \tag{11}$$

where $PSNR$ denotes the peak signal to noise ratio (PSNR), and $\hat{W}$ is an estimate of W, obtained by modifying I using the significant features of W.

To solve this problem, our robust image hashing scheme, called structural digital signature (SDS) [23] is modified and applied to efficiently extract the significant difference between a Wyner-Ziv frame and its reference frame as the Wyner-Ziv bits at the encoder without performing motion estimation. Then, these transmitted Wyner-Ziv bits are incorporated with the generated side information (*i.e.*, the decoded/generated reference frame) to reconstruct the Wyner-Ziv frame at the decoder without performing motion estimation.

B. Structural Digital Signature

The structural digital signature method proposed in [23], which can extract the most significant components and provide a compact representation for an image efficiently, is adopted in ProposedDVC2. SDS is constructed in the discrete wavelet transform (DWT) domain due to its excellent multiscale and precise localization properties. In fact, the SDS is derived from the interscale relationships between wavelet coefficients.

For an image of size $M_1 \times M_2$, a J-scale DWT is performed. Let $w_{s,o}(x, y)$ represent a wavelet coefficient at scale s, orientation o, and position (x, y), $0 \leq s < J$, $1 \leq x \leq M_1$, and $1 \leq y \leq M_2$. It is known that a large/small scale represents a coarser/finer resolution of an image, *i.e.*, the low/high frequency part. The orientation o may be in a horizontal, vertical, or diagonal direction. The interscale relationships between wavelet coefficients can be converted into the relationships between the parent node $w_{s+1,o}(x, y)$ and its four child nodes $w_{s,o}(2x + i, 2y + j)$, $0 \leq i, j \leq 1$, with

$$|| w_{s+1,o}(x, y)| - | w_{s,o}(2x + i, 2y + j)|| \geq \delta, \tag{12}$$

where δ is a postive number.

Slightly different from [23], for each pair consisting of a parent node $w_{s+1,o}(x, y)$ and its four child nodes $w_{s,o}(2x + i, 2y + j)$, the maximum magnitude difference is calculated as

$$max\_mag\_diff_{s+1,o}(x, y) = \max_{0 \leq i, j \leq 1} \left\| w_{s+1,o}(x, y) \right| - \left| w_{s,o}(2x + i, 2y + j) \right\|. \tag{13}$$

Based on [23], the significance of a parent-child pair is completely dependent on their magnitude difference. The larger the difference is, the more significant the parent-child pair is. Here, all the "parent-4 children pairs" in a wavelet image will be arranged, *i.e.*, sorted in decreasing order based on their maximum magnitude differences. The first L parent-4 children pairs are selected for constructing the SDS of an image, where L denotes the SDS length. L should be properly determined in order to ensure that the selected parent-4 children pairs are really significant. The selection of L will be discussed later.

Once the significant parent-4 children pairs are selected, each pair will be assigned a symbol representing what kind of relationship this pair carries. According to the interscale relationship existing among wavelet coefficients, there are four possible relationship types. Assume the magnitude of a parent node p is larger than that of its child node c. When $|p| \geq |c|$, the four possible relationships of the pair are (a) $p \geq 0$, $c \geq 0$; (b) $p \geq 0$, $c < 0$; (c) $p < 0$, $c \geq 0$; and (d) $p < 0$, $c < 0$. To make the above-mentioned relationships compact, the relations (a) and (b) can be merged to form a signature symbol "+1" when $p \geq 0$ and c is ignored. On the other hand, the relations (c) and (d) can be merged into another signature symbol "-1" when $p < 0$ and c is ignored. That is, one should keep the sign of the larger node unchanged while ignoring the smaller one under the constraint that their original interscale relationship is still preserved. Similarly, the signature symbols "+2" and "-2" can be defined under the constraint $|p| < |c|$. In summary, the signature symbol $sym(p, c)$ is defined as

$$sym(p,c) = \begin{cases} +1 & if \quad (|p| \geq |c|) \quad and \quad (p \geq 0), \\ -1 & if \quad (|p| \geq |c|) \quad and \quad (p < 0), \\ +2 & if \quad (|p| < |c|) \quad and \quad (c \geq 0), \\ -2 & if \quad (|p| < |c|) \quad and \quad (c < 0). \end{cases} \tag{14}$$

Those pairs not included in the SDS (outside the first L pairs) will be labeled by the symbol, "0." Each parent-4 children pair in a wavelet image will be labeled by one of the five symbols (+1, -1, +2, -2, and 0). Here, the SDS information for a selected parent-4 children pair includes (a) its signature symbol (except "0"); (b) the spatial position of the parent node; and (c) the maximum magnitude difference.

C. Proposed Wyner-Ziv Video Codec (ProposedDVC2)

The block diagram of ProposedDVC2 is shown in Fig. 8. The proposed Wyner-Ziv encoder and decoder in ProposedDVC2 are described in the following sections. We also analyze the required parameters and the computational complexity.

C.1. Proposed Wyner-Ziv video encoder in ProposedDVC2

In ProposedDVC2, at the encoder, an input video sequence is divided into several GOPs, in which a GOP consists of a key frame followed by several Wyner-Ziv frames. For a frame I_i, i = 0, 1, 2, …, $NFrame$ – 1, where $NFrame$ denotes the number of frames in a sequence, if i mod $GOPSize$ = 0, I_i denotes a key frame; otherwise, it is a Wyner-Ziv frame.

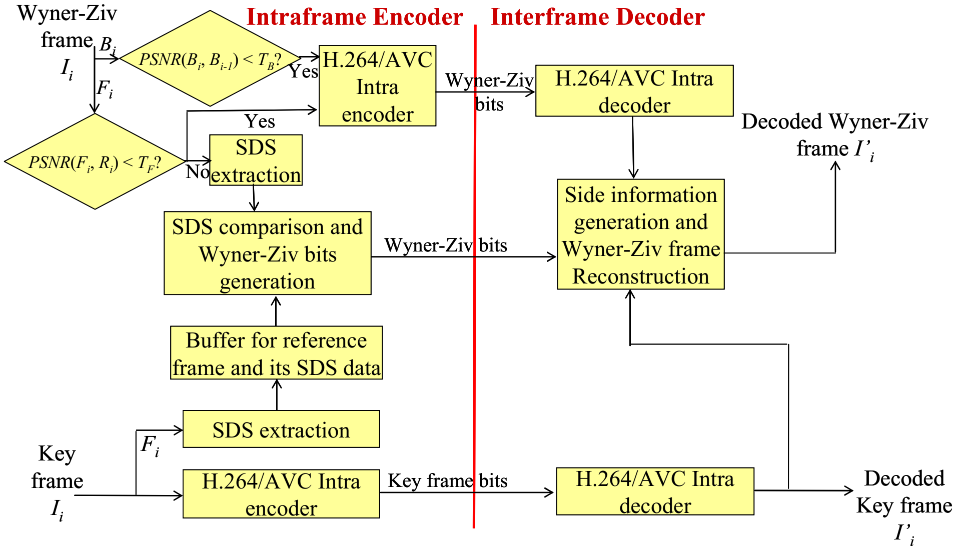


Fig. 8. The proposed Wyner-Ziv video codec based on robust media hashing (ProposedDVC2).

At the encoder, for each frame I_i, the luminance component of the central $2^n \times 2^n$ square area within the frame is extracted as F_i. For example, for a QCIF frame I_i of size 176×144, the luminance component of the central 128×128 ($n = 7$) square area is extracted as F_i. The remaining area within I_i is denoted by B_i. In addition, the chrominance components of the whole frame I_i are included in B_i. Usually, F_i and B_i correspond to the foreground and the background of a frame, respectively, as shown in Fig. 9.

For convenience, some notations are defined as follows. The wavelet image of F_i is denoted by X_i. The reconstructed F_i at the decoder is denoted by $\hat{F}_i$. The wavelet image of $\hat{F}_i$ is denoted by Y_i. The SDS of F_i is denoted by S_i. The length of S_i is denoted by L_i. The reconstructed B_i at the decoder is denoted by $\hat{B}_i$.

Each key frame I_i is encoded using the H.264/AVC intraframe encoder. An additional operation for a key frame is to extract the SDS S_i from its foreground F_i of size $2^n \times 2^n$. During the process of extracting S_i, the DWT is applied to transform F_i to the wavelet image X_i. The SDS S_i for F_i with length L_i is extracted from X_i and stored in the encoder buffer for encoding the previous/next Wyner-Ziv frame. The outputs of the H.264/AVC intraframe encoder form the key frame bits.

For each Wyner-Ziv frame I_i, its foreground F_i of size $2^n \times 2^n$ is extracted and wavelet transformed to be X_i. Then, the similarity between F_i and its foreground reference frame R_i is evaluated, where R_i can be available from the encoder buffer. Here, the reference frame R_i for F_i is determined as follows.

(a) If the immediate previous frame (I_{i-1}) of I_i is a key frame, the reference frame for F_i is set to the foreground (F_{i-1}) of I_{i-1}. That is, $R_i = F_{i-1}$.

(b) If the immediate next frame (I_{i+1}) of I_i is a key frame, the reference frame for F_i is set to the foreground (F_{i+1}) of I_{i+1}. That is, $R_i = F_{i+1}$.

(c) If I_i is between two Wyner-Ziv frames, the reference frame for F_i is set to the frame simply interpolated (averaged) by the foregrounds of the two nearest key frames.

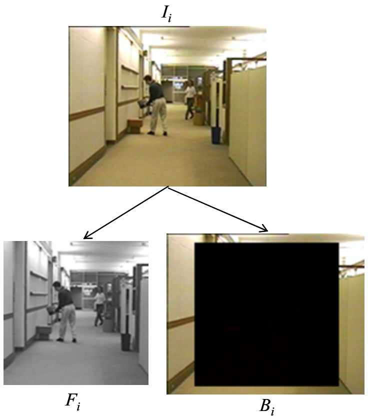


Fig. 9. The decomposition of a QCIF video frame I_i into a foreground component (F_i) and a background component (B_i).

For example, consider a video sequence, I_0, I_1, I_2, I_3, I_4, …, with *GOPSize* = 4, *i.e.*, I_0, I_4, I_8, … are key frames while the others are Wyner-Ziv frames. Based on the above definitions for reference frames, the reference frame R_1 for F_1 is F_0, the reference frame R_2 for F_2 is the frame averaged by F_0 and F_4, the reference frame R_3 for F_3 is F_4, and so on. The major principle is that the reference frame for a Wyner-Ziv frame is derived from neighboring key frames, instead of Wyner-Ziv frames. That is, key frames are always intra-encoded with higher quality and intra-decoded. They are more suitable to be reference frames for Wyner-Ziv frames.

In this study, the PSNR value, $PSNR(F_i, R_i)$, is used to evaluate the similarity between F_i and R_i. If $PSNR(F_i, R_i) < T_a$, the SDS length L_i of F_i is set to L_1. If $T_a \leq PSNR(F_i, R_i) < T_b$, L_i is set to L_2. If $PSNR(F_i, R_i) \geq T_b$, L_i is set to L_3. The relationship among L_1, L_2, and L_3, and the selection of T_a and T_b will be described later. Finally, the SDS S_i for F_i with length L_i is extracted from X_i (the wavelet image of F_i).

The remaining work for encoding F_i is to extract the most significant wavelet coefficients in X_i by comparing S_i and S_{Ri}, which is available from the encoder buffer and extracted from the reference frame R_i.

For S_i, each signature symbol $sym_i(p, c)$ (= +1, -1, +2, -2, or 0) will be compared with the corresponding symbol $sym_{Ri}(p, c)$ in S_{Ri} with the same position for the parent node. If $sym_i(p, c) \neq sym_{Ri}(p, c)$, the corresponding parent-4 children pair of $sym_i(p, c)$ in S_i is determined to be significant. If $sym_i(p, c) = sym_{Ri}(p, c) \neq 0$, then their corresponding maximum magnitude difference (Eq. (13)) will be compared. If the difference of their maximum magnitude differences is larger than a threshold D_i, then the parent-4 children pair corresponding to $sym_i(p, c)$ is determined to be significant; otherwise, it is insignificant. That is, we intend to efficiently extract the wavelet coefficients from the wavelet domain X_i of F_i, that are significantly different from the corresponding ones from X_{Ri} of R_i. For each significant parent-4 children pair, the position of the parent node and their corresponding five quantized wavelet coefficients form the Wyner-Ziv bits. Here, for a $2^n \times 2^n$ square area F_i, it takes

$log_2(2^n \times 2^n)$ bits to denote a parent-node position. Similar to [25], a wavelet coefficient, w, is quantized as

$$\hat{w} = \lfloor w/Q_s + 0.5 \rfloor, \tag{15}$$

where Q_s denotes the quantization parameter for the wavelet scale s that w belongs in, and $\lfloor \ \rfloor$ denotes the floor operation. Then all the quantized significant wavelet coefficients are entropy encoded. Finally, both the key frame bits and the Wyner-Ziv bits will be transmitted to the decoder. An illustrated example for encoding with *GOPSize* = 4 in ProposedDVC2 is shown in Fig. 10.

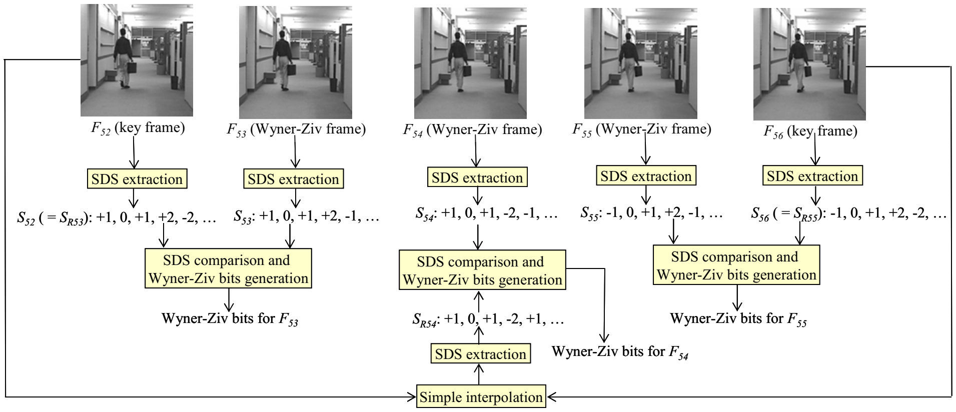

F_{52} (key frame)
F_{53} (Wyner-Ziv frame)
F_{54} (Wyner-Ziv frame)
F_{55} (Wyner-Ziv frame)
F_{56} (key frame)
SDS extraction
SDS extraction
SDS extraction
SDS extraction
SDS extraction
S_{52} (= S_{R53}): +1, 0, +1, +2, -2, …
S_{53}: +1, 0, +1, +2, -1, …
S_{54}: +1, 0, +1, -2, -1, …
S_{55}: -1, 0, +1, +2, -1, …
S_{56} (= S_{R55}): -1, 0, +1, +2, -2, …
SDS comparison and Wyner-Ziv bits generation
SDS comparison and Wyner-Ziv bits generation
SDS comparison and Wyner-Ziv bits generation
Wyner-Ziv bits for F_{53}
S_{R54}: +1, 0, +1, -2, +1, …
Wyner-Ziv bits for F_{55}
SDS extraction
Wyner-Ziv bits for F_{54}
Simple interpolation

Fig. 10. An illustrated example for encoding with *GOPSize* = 4 in ProposedDVC2.

Similar to the selection for L_i, D_i is selected as follows. If $PSNR(F_i, R_i) < T_a$, D_i is set to D_1. If $T_a \leq PSNR(F_i, R_i) < T_b$, D_i is set to D_2. If $PSNR(F_i, R_i) \geq T_b$, D_i is set to D_3. Obviously, the larger $PSNR(F_i, R_i)$ is, the more similar F_i and R_i are. When F_i and R_i are similar, significant wavelet coefficients in F_i different from the corresponding ones in R_i that should be extracted are few, implying that smaller L_i and larger D_i should be used, and vice versa. Here, L_1, L_2, L_3, D_1, D_2, and D_3 can be adjusted to generate different amounts of the Wyner-Ziv bits under the constraints: $L_1 \geq L_2 \geq L_3$ and $D_1 \leq D_2 \leq D_3$.

For each Wyner-Ziv frame I_i, its background B_i is either encoded using the H.264/AVC intraframe encoder or skipped based on the background content differences. That is, if $PSNR(B_i, B_{i-1}) < T_G$, where T_G is a predefined positive threshold, B_i is encoded using the H.264/AVC intraframe encoder. Otherwise, B_i is skipped. Usually, B_i is encoded only for fast-motion video sequences.

On the other hand, for the foreground F_i of a Wyner-Ziv frame I_i, if the video contents between F_i and R_i are sufficiently different, more significant wavelet coefficients for F_i will be extracted, which will be encoded inefficiently. Hence, if $PSNR(F_i, R_i) < T_F$, where T_F is a predefined positive threshold, F_i will be encoded using the H.264/AVC intraframe encoder. However, this kind of case usually occurs only for fast-motion video sequences.

C.2. Determination of thresholds T_a and T_b

The determination of T_a and T_b is described as follows. Obviously, if F_i and R_i are very similar, *i.e.*, $PSNR(F_i, R_i)$ is sufficiently large, a large value of L_i is meaningless. On the other hand, if F_i and

R_i are very different, a large value of L_i is useful. Now, we want to roughly evaluate the quality of the reconstructed Wyner-Ziv frames under different SDS lengths to determine T_a and T_b. The two thresholds can be viewed as the break points indicating what SDS length should be used for the current Wyner-Ziv frame. They can be approximately determined from the relationship between each pair of $PSNR(F_i, R_i)$ and $PSNR(F_i, F'_i)$ under different SDS lengths for several video sequences, where F'_i is obtained as follows. The significant quantized wavelet coefficients for F_i are extracted by comparing S_i and S_{Ri} under different SDS lengths. Only the wavelet coefficients of each parent-4 children pair for F_i with the symbol different from the corresponding symbol in R_i are filled into the wavelet image Y_{Ri} of $\hat{R}_i$ (the reconstructed R_i at the decoder). The filled image is inverse wavelet transformed to obtain F'_i. The relationship between $PSNR(F_i, R_i)$ and $PSNR(F_i, F'_i)$ under the four different SDS lengths is shown in Fig. 11 for *GOPSize* = 4. It can be observed from Fig. 11 that when $PSNR(F_i, R_i) \geq 38$ dB, the differences among various $PSNR(F_i, F'_i)$ values under the four different SDS lengths are not significant. However, when $PSNR(F_i, R_i) < 34$ dB, the differences among the $PSNR(F_i, F'_i)$ values under the four different SDS lengths are more significant. Here, T_a and T_b are set to 34 dB and 38 dB, respectively, based on empirical observations. However, T_a and T_b can be also adjusted based on desired target bitrates or current network conditions. That is, if the desired target bitrate is with low bitrate, both T_a and T_b should be smaller to induce smaller L_i, larger D_i, and fewer Wyner-Ziv bits. Otherwise, they should be larger. Similarly, if current network traffic load is heavy, the low bitrate case should be applied and vice versa.

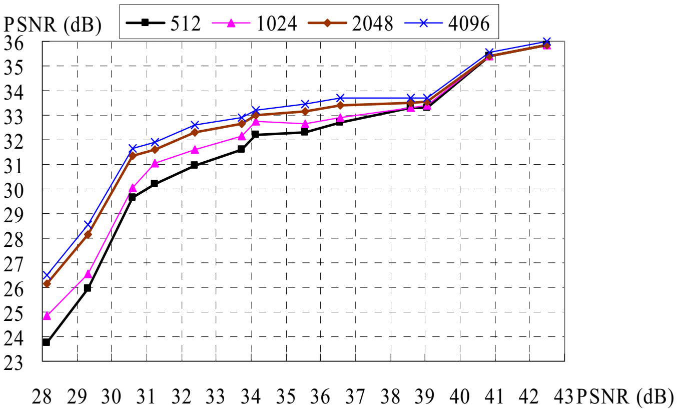


Fig. 11. The relationships between *PSNR*(*F*$_i$, *R*$_i$) (the horizontal axis) and *PSNR*(*F*$_i$, *F'*$_i$) (the vertical axis) under the four different SDS lengths.

C.3. The buffer size for storing SDS information

For the purpose of calculating $PSNR(F_i, R_i)$ and comparing S_i and S_{Ri}, both R_i and S_{Ri} should be stored in the buffer. The required buffer size can be calculated as follows. It takes $8 \times 2^n \times 2^n$ bits to store uncompressed R_i. For storing S_{Ri}, its three components should be stored as follows: (a) it takes 2 bits to store each signature symbol; (b) it takes $log_2(2^n \times 2^n)$ bits to store each parent node position in R_i of size $2^n \times 2^n$; and (c) it takes $log_2(Diff_{Ri})$ bits to store the maximum magnitude difference ($Diff_{Ri}$) for each parent-4 children pair. For S_{Ri} with length L_{Ri}, it takes $L_{Ri} \times [2 + log_2(2^n \times 2^n) + log_2(Diff_{Ri})]$

bits to store S_{Ri}. Totally, the required encoder buffer size is $\{8\times 4^n + L_{Ri}\times[2 + 2n + log_2(Diff_{Ri})]\}$ bits. For example, in this study, the maximum possible SDS length is 4096, *i.e.*, $max\{L_{Ri}\} = 4096$. The range for the wavelet coefficients is [-4096, 4096], hence, the maximum possible maximum magnitude difference is 4096, *i.e.*, $max\{Diff_{Ri}\} = 4096$. When R_i is with size 128×128, *i.e.*, $n = 7$, the required buffer size is $[8\times4^7 + 4096\times(2 + 2\times7 + log_2 4096)]$ bits = 245,760 bits = 30 Kbytes.

C.4. Proposed Wyner-Ziv video decoder in ProposedDVC2

At the decoder, each key frame is decoded using the H.264/AVC decoder. For each Wyner-Ziv frame I_i, the received Wyner-Ziv bits and the side information are used to reconstruct Y_i (the wavelet image of $\hat{F}_i$). First, the significant wavelet coefficients recorded in the Wyner-Ziv bits are entropy decoded and dequantized. Here, the side information is the wavelet image (Y_{Ri}) of the reconstructed foreground reference frame ($\hat{R}_i$), corresponding to R_i. $\hat{R}_i$ denotes the foreground component of the immediate previous reconstructed key frame or the immediate next reconstructed key frame or the frame interpolated from the two nearest reconstructed neighboring key frames. The side information Y_{Ri} is obtained by wavelet transforming $\hat{R}_i$. Then, Y_i is reconstructed by filling the decoded significant wavelet coefficients into the wavelet image Y_{Ri}. Finally, Y_i is directly used to reconstruct F_i by inverse wavelet transforming Y_i to $\hat{F}_i$. On the other hand, if F_i is intra-encoded at the encoder, it is decoded using the H.264/AVC intraframe decoder as $\hat{F}_i$. For reconstruction of the background component, if B_i is skipped at the encoder, it is reconstructed by copying the corresponding regions in the immediate previous reconstructed frame as $\hat{B}_i$. Otherwise, it is decoded using the H.264/AVC intraframe decoder as $\hat{B}_i$. Finally $\hat{F}_i$ and $\hat{B}_i$ are combined to reconstruct I_i as $\hat{I}_i$.

D. Computational Complexity of ProposedDVC2

The computational complexity of ProposedDVC2 is dominated by those of the DWT, SDS extraction, and entropy encoding. The heaviest task in the SDS extraction is the sorting operation, which can be efficiently performed by using the quick sort algorithm. Without performing motion estimation, the computational complexity of the proposed encoder should be in the similar order of that of a conventional intraframe encoder, consisting of the DCT and entropy coding. On the other hand, the computational complexity of the proposed Wyner-Ziv video decoder is dominated by those of the inverse DWT and entropy decoding, which is in the order of a conventional intraframe decoder. Hence, unlike most existing Wyner-Ziv video codecs, ProposedDVC2 is with light encoder and light decoder. However, the decoder will induce some delay due to the fact that the reference frames for some Wyner-Ziv frames should be derived from the next key frame of the next GOP. For example, the second Wyner-Ziv frame in a GOP with *GOPSize* = 4 will have 2-frame delays. The third Wyner-Ziv frame in a GOP with *GOPSize* = 4 will have 1-frame delay. The maximum possible decoding delay (*DD*) for *GOPSize* ≥ 3 is *GOPSize* – 2, *i.e.*, 0 ≤ *DD* ≤ *GOPSize* – 2 for *GOPSize* ≥ 3. When 1 ≤ *GOPSize* ≤ 2, there is no decoding delay (*DD* = 0).

IV. Simulation Results

A. Experimental Setting

Several QCIF video sequences formatted with different GOP sizes (*GOPSize* = 2, 4, and 8), frame rate (10 frames per second (fps)), and encoded with several different bitrates were used to evaluate the two proposed Wyner-Ziv video codecs (ProposedDVC1 and ProposedDVC2). In ProposedDVC1, 4×4 DCT (N = 4) was used. Here, the bitrates were adjusted by changing the quantization parameters (QPs) of the H.264/AVC encoder for key frames, changing the quantizers for Wyner-Ziv frames, and changing the number of blocks for each coding mode (*i.e.*, changing T_1 and T_2). The number of reference frames for motion compensation at the decoder was set to 1.

In ProposedDVC2, the central square area with size 128×128 (n = 7) was extracted for each frame. The SDS for each 128×128 square area was extracted by setting the size of lowest frequency subband in the wavelet domain to 16×16. Here, the bitrates were adjusted by changing the QPs of the H.264/AVC encoder for key frames and the parameters, L_1, L_2, L_3, D_1, D_2, D_3, Q_s, T_F, and T_G for Wyner-Ziv frames. A guideline for empirically adjusting L_1, L_2, L_3, D_1, D_2, and D_3 under the constraints, $L_1 \geq L_2 \geq L_3$ and $D_1 \leq D_2 \leq D_3$, was found to be $4096 \geq L_1 \geq L_2 \geq L_3 \geq 512$ and $30 \leq D_1 \leq D_2 \leq D_3 \leq 240$. For example, if the QP of the H.264/AVC intraframe encoder is set to 33, then L_1 = 2048, L_2 = 1024, L_3 = 1024, D_1 = 20, D_2 = 20, D_3 = 25, Q_0 = 20, Q_1 = 20, Q_2 = 30, Q_3 = 30, T_F = 29 dB, and T_G = 28 dB can be set to yield the bitrate = 62.39 kbps and PSNR = 34.88 dB for the *Hall Monitor* sequence. Here, all the parameters are adjusted in order to achieve the desired bitrates. One can adjust the QP of the H.264/AVC intraframe encoder to approximately achieve the desired bitrate, and then, adjust L_1, L_2, L_3, D_1, D_2, D_3, and Q_s to accurately achieve the desired bitrate. The larger the QP of the H.264/AVC intraframe encoder is, the smaller the achieved bitrate is, and vice versa. The larger L_1, L_2, and L_3 are, the higher the achieved bitrate is, and vice versa. The larger D_1, D_2, and D_3 are, the smaller the achieved bitrate is, and vice versa. The larger Q_s is, the smaller the achieved bitrate is, and vice versa. Given a fixed QP of the H.264/AVC intraframe encoder, gradually adjusting L_1, L_2, L_3, D_1, D_2, D_3, and Q_s will gradually change the achieved RD performance, and then make the RD performance become saturation. On the other hand, T_F and T_G are usually set to be smaller to make intraframe refresh for a Wyner-Ziv frame occur infrequently.

The H.264/AVC intraframe coding (*GOPSize* = 1) and H.264/AVC interframe coding [2] were employed for comparison with the two proposed codecs. Here, the setting for the H.264/AVC interframe coding is as follows: (a) the same GOP sizes (*GOPSize* = 2, 4, and 8) were employed; (b) each I frame was adjusted to the same as the corresponding key frames in the two proposed codecs; (c) the bitrates were adjusted by changing the QPs for each frame; (d) the number of the reference frames was set to 1; and (e) the RD optimization was off. The test video sequences were categorized as very slow-motion (*Claire*), slow/middle-motion (*Hall monitor*, *Mother and daughter*, and *Salesman*), and fast-motion (*Carphone*) sequences.

B. RD Performance Comparison

In this section, the RD performance comparison was conducted under (very) low bitrates for the *Carphone*, *Claire*, *Hall monitor*, *Mother and daughter*, and *Salesman* sequences under four different methods, *i.e.*, ProposedDVC1, ProposedDVC2, H.264/AVC interframe coding, and H.264/AVC intraframe coding. The obtained results are shown in Figs. 12-19. Examples of the video frames for the *Hall Monitor* sequence with *GOPSize* = 4 decoded using the H.264/AVC interframe coding, ProposedDVC1, ProposedDVC2, and the H.264/AVC intraframe coding at similar bitrates are shown in Fig. 20 for visual quality inspection, where the initial frame number is zero.

For the very slow-motion sequence (*Claire*), several observations can be found from Fig. 12. They are described as follows: (a) when *GOPSize* = 2, the RD performance of ProposedDVC1 is slightly better than those of H.264/AVC interframe coding and ProposedDVC2 among various evaluated bitrates. The PSNR performance gains of the two proposed codecs above those of the H.264/AVC intraframe coding are from 3 to 4 dB. The PSNR performance gaps between ProposedDVC2 and the H.264/AVC interframe coding are within 1 dB; (b) when *GOPSize* = 4 (also highlighted in Fig. 13), the PSNR performance of ProposedDVC2 slightly above that of ProposedDVC1 is within 1 dB. The PSNR performance gains of the two proposed codecs above those of the H.264/AVC intraframe coding are from 4 to 6 dB. The PSNR performance gaps between the two proposed codecs and the H.264/AVC interframe coding are from 1 to 3 dB; and (c) when *GOPSize* = 8, the PSNR performance gains of ProposedDVC1 are better than those of ProposedDVC2 from 0.5 to 2 dB. The PSNR performance gains of the two proposed codecs above those of the H.264/AVC intraframe coding are from 0.5 to 7 dB. The PSNR performance gaps between the two proposed codecs and the H.264/AVC interframe coding are from 5 to 7 dB.

In conventional video coding (*e.g.*, H.264/AVC), the coding performance with larger GOP size is usually better than that with smaller GOP size due to motion estimation can efficiently reduce temporal redundancy. However, it is not usually true for distributed video coding due to the fact that all frames are intra encoded. Based on the above observations for the very slow-motion sequence (*Claire*), when *GOPSize* = 2, there is only one Wyner-Ziv frame between two key frames. For ProposedDVC1, good side information can be obtained and many blocks are skipped due to the very slow motion. For ProposedDVC2, only a few Wyner-Ziv bits are generated. For H.264/AVC interframe coding, due to the fact that only one P frame is between two I frames, the motion estimation is not very efficient. Hence, the performances of the two proposed codecs are very close to that of H.264/AVC interframe coding. When *GOPSize* = 4, ProposedDVC2 can outperform

ProposedDVC1 due to ProposedDVC1 spends too many bits to denote the coding mode information (even for many blocks with skip mode) whereas ProposedDVC2 only spends a few bits for significant differences between successive frames due to the very slow motion. When *GOPSize* = 8, the RD curves of the two proposed codecs become flat quickly due to the fact that more bits from fewer key frames used cannot be efficiently employed for simple texture information in the *Claire* sequence without performing motion estimation. The H.264/AVC interframe coding can efficiently employ more bits and significantly outperform the two proposed codecs due to the larger GOP size, slow motion, and simple texture information in the *Claire* sequence. The RD performance of the H.264/AVC intraframe coding can quickly come up with those of the two proposed codecs due to the intraframe coding can efficiently employ data bits for simple texture information in the *Claire* sequence. In addition, it can be observed from Fig. 12 that the two proposed codecs with *GOPSize* = 4 roughly outperform those with *GOPSize* = 2 and *GOPSize* = 8. In summary, it is recommended that the two proposed codecs with *GOPSize* = 4 are more suitable for very slow motion sequences.

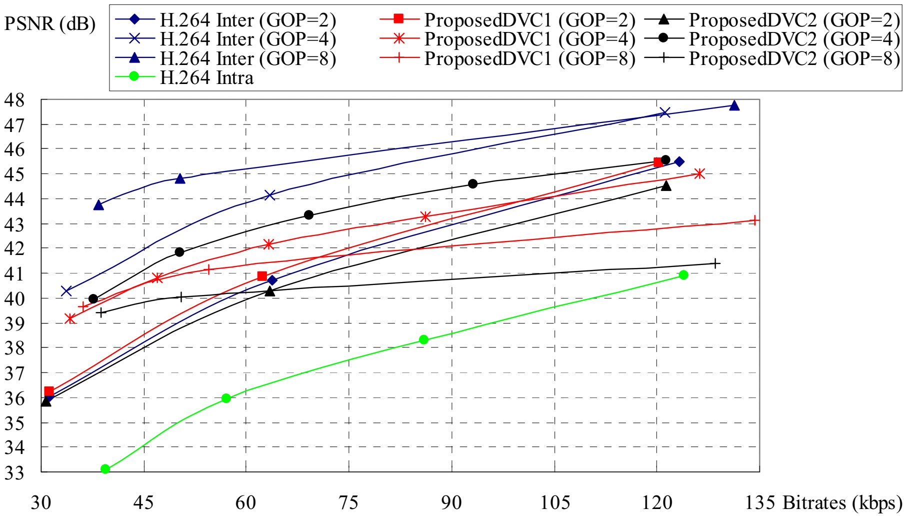


Fig. 12. RD performance for the *Claire* sequence with different GOP sizes.

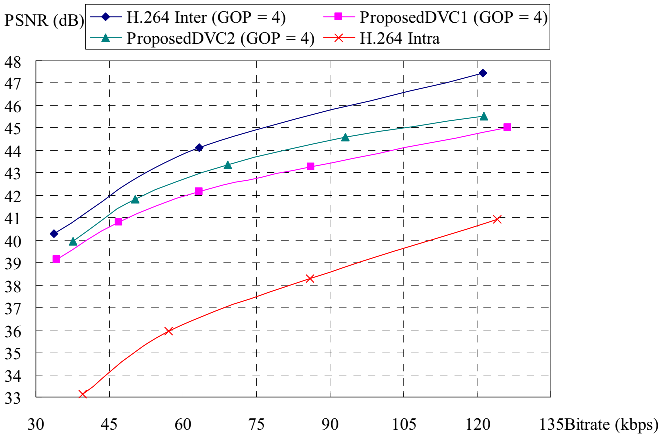


Fig. 13. RD performance for the *Claire* sequence with *GOPSize* = 4**.**

For the *Hall monitor* sequence (slow/middle-motion), several observations can be found from the obtained RD performance, as shown in Fig. 14. When *GOPSize* = 2, the RD performance of ProposedDVC1 is slightly better than those of H.264/AVC interframe coding and ProposedDVC2. The PSNR performance gains of the two proposed codecs above those of the H.264/AVC intraframe coding are from 3 to 4 dB. The PSNR performance gaps between ProposedDVC2 and the H.264/AVC interframe coding are within 1 dB. When *GOPSize* = 4, the PSNR performance gains of ProposedDVC1 are slightly better than those of ProposedDVC2 (within 1 dB). The PSNR performance gains of the two proposed codecs above those of the H.264/AVC intraframe coding are from 5 to 6 dB. The PSNR performance gaps between the two proposed codecs and the H.264/AVC interframe coding are from 1 to 2 dB. The results of the two proposed codecs for *GOPSize* = 4 are comparable with the results shown in [14]. When *GOPSize* = 8, the PSNR performance gains of ProposedDVC1 are 2~3 dB higher than those of ProposedDVC2. The PSNR performance gains of ProposedDVC1 above those of the H.264/AVC intraframe coding are from 6 to 8 dB. The PSNR performance gains of ProposedDVC2 above those of the H.264/AVC intraframe coding are from 3 to 6 dB. The PSNR performance gaps between ProposedDVC1 and the H.264/AVC interframe coding are from 1 to 2 dB. The PSNR performance gaps between ProposedDVC2 and the H.264/AVC interframe coding are from 3 to 5 dB. The results of ProposedDVC1 for *GOPSize* = 8 are comparable with the results shown in [15].

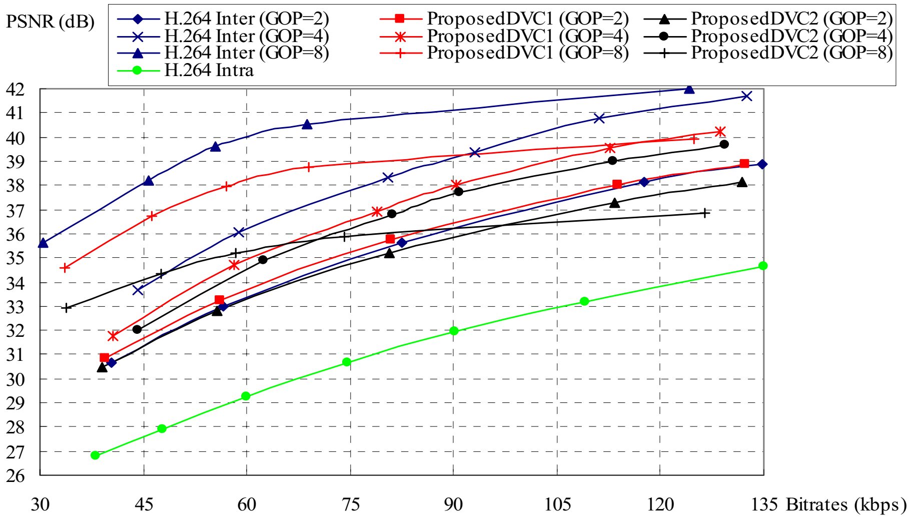


Fig. 14. RD performance for the *Hall monitor* sequence.

Based on the above observations for the *Hall monitor* sequence, when *GOPSize* = 2, for ProposedDVC1, good side information can be obtained and the background areas are almost still due to most motions occur in the foreground areas. For ProposedDVC2, only a few Wyner-Ziv bits are generated. Hence, the performance of the two proposed codecs is very close to that of H.264/AVC interframe coding. When *GOPSize* = 4, ProposedDVC1 can slightly outperform ProposedDVC2 due to the decoder's motion compensation in ProposedDVC1 is efficient for some foreground motions whereas ProposedDVC2 should spend more bits for significant differences between successive frames due to foreground motions. When *GOPSize* = 8, ProposedDVC1 can significantly outperform ProposedDVC2 due to *GOPSize* = 8 is too large for ProposedDVC2 without performing motion estimation. In addition, it can be observed from Fig. 14 that ProposedDVC1 with *GOPSize* = 4 consistently outperform that with *GOPSize* = 2 and can outperform that with *GOPSize* = 8 when the bitrate approaches 120 kbps. Similarly, ProposedDVC2 with *GOPSize* = 4 consistently outperforms that with *GOPSize* = 2 and can outperform that with *GOPSize* = 8 when the bitrate approaches 70 kbps. In summary, it is recommended, again, that the two proposed codecs with *GOPSize* = 4 are more suitable for the *Hall monitor* sequence.

For the *Salesman* sequence (slow/middle-motion), several observations can be found from the obtained RD performance, as shown in Fig. 15. The simulation results with *GOPSize* = 2 and *GOPSize* = 4 are similar to those of the *Hall monitor* sequence with *GOPSize* = 2 and *GOPSize* = 4, respectively. When *GOPSize* = 8 (also highlighted in Fig. 16), the PSNR performance gains of ProposedDVC1 are 0.5~1 dB higher than those of ProposedDVC2. The PSNR performance gains of ProposedDVC1 above those of the H.264/AVC intraframe coding are from 6 to 8 dB. The PSNR performance gains of ProposedDVC2 above those of the H.264/AVC intraframe coding are from 6 to 7 dB. The PSNR performance gaps between ProposedDVC1 and the H.264/AVC interframe coding are from 1 to 3 dB. The PSNR performance gaps between ProposedDVC2 and the H.264/AVC interframe coding are from 1 to 4 dB. The results of the two proposed codecs for *GOPSize* = 8 are comparable with the results shown in [15].

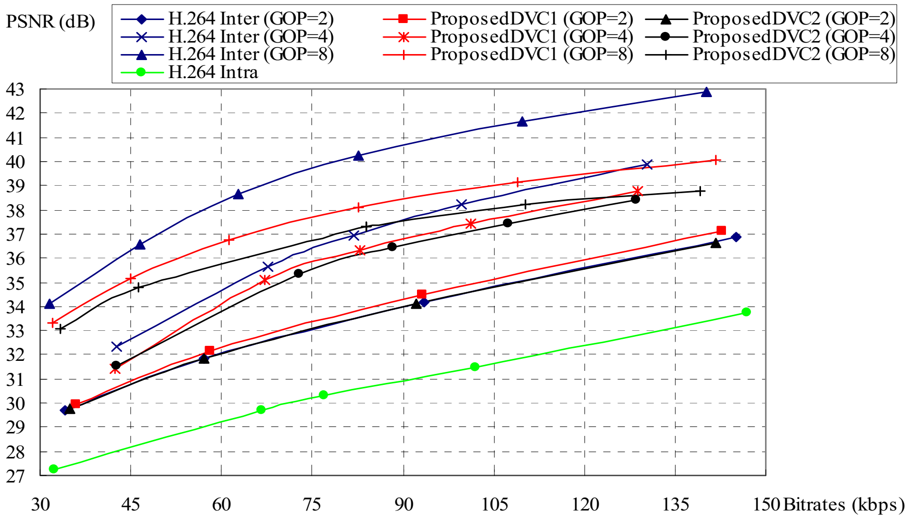


Fig. 15. RD performance for the *Salesman* sequence.

In Fig. 16, it can be observed that when *GOPSize* = 8, the performance gap between the two proposed codecs is smaller than that of the *Hall monitor* sequence. The reason is that some slight "occlusions" occur in the *Hall monitor* sequence. That is, an object appears in a frame, but does not exist in the previous frame. In this situation, ProposedDVC2 will spend more bits to denote the frame containing the appeared object. However, no such situations appear in the *Salesman* sequence. In addition, it can be observed from Fig. 15 that the two proposed codecs with *GOPSize* = 8 outperform those with *GOPSize* = 2 and *GOPSize* = 4. In summary, it is recommended again that the two proposed codecs with *GOPSize* = 8 are more suitable for the *Salesman* sequence.

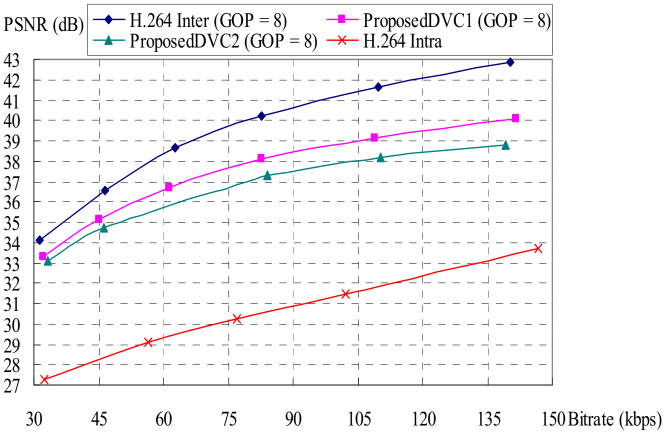


Fig. 16. RD performance for the *Salesman* sequence with *GOPSize* = 8**.**

For the *Mother and daughter* sequence (slow/middle-motion), several observations can be found from the obtained RD performance, as shown in Fig. 17. The simulation results with *GOPSize* = 2 and *GOPSize* = 4 are similar to those of the *Hall monitor* sequence with *GOPSize* = 2 and *GOPSize* = 4, respectively, except that the PSNR performance gains of the two proposed codecs above those of the H.264/AVC intraframe coding are slightly worse than those of the *Hall monitor* sequence. The reasons are that a few fast motions (*e.g.*, the movement of the mother's hand) appear occasionally and in fact, the performance gaps between the H.264/AVC interframe coding and the H.264/AVC intraframe coding are smaller. If the performance of the two proposed codecs cannot further approach that of the H.264/AVC interframe coding, the performance gains above those of the H.264/AVC intraframe coding will be smaller. For *GOPSize* = 8, it is too large for the two proposed codecs. Hence, *GOPSize* = 4 is more suitable for the *Mother and daughter* sequence. In summary, it is recommended again that the two proposed codecs with *GOPSize* = 4 are more suitable for most slow/middle-motion sequences.

For the *Carphone* sequence (fast-motion), several observations can be found from the obtained RD performance, as shown in Fig. 18. When *GOPSize* = 2 (also highlighted in Fig. 19), the RD performance of ProposedDVC1 above that of ProposedDVC2 is about 1 dB. The PSNR performance gains of the two proposed codecs above those of the H.264/AVC intraframe coding are from 1 to 2.5 dB. The PSNR performance gaps between the two proposed codecs and the H.264/AVC interframe coding are from 0.5 to 3 dB. As for *GOPSize* = 4 and *GOPSize* = 8, they are too large for the two proposed codecs. Although the main object in the *Carphone* sequence does not perform very large motions, many large global motions exist in the sequence. In summary, it is recommended that the two proposed codecs with *GOPSize* = 2 are more suitable for fast-motion sequences.

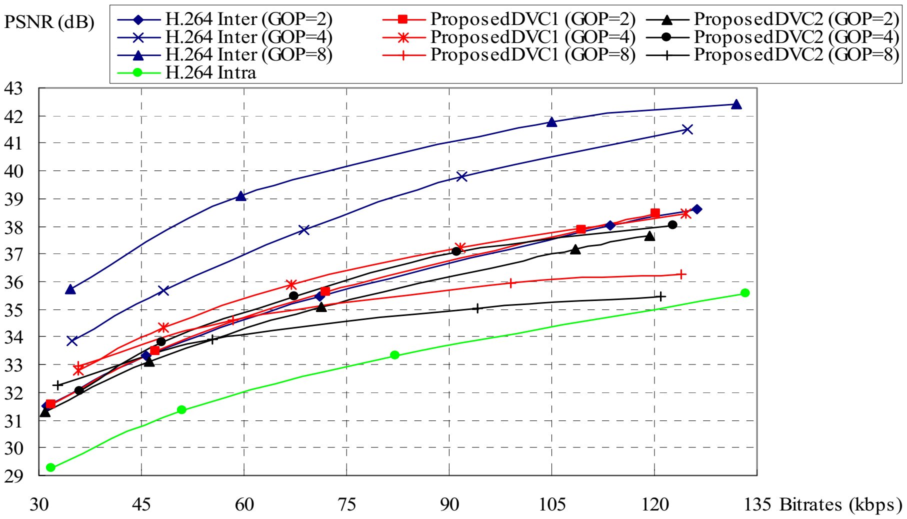


Fig. 17. RD performance for the *Mother and Daughter* sequence.

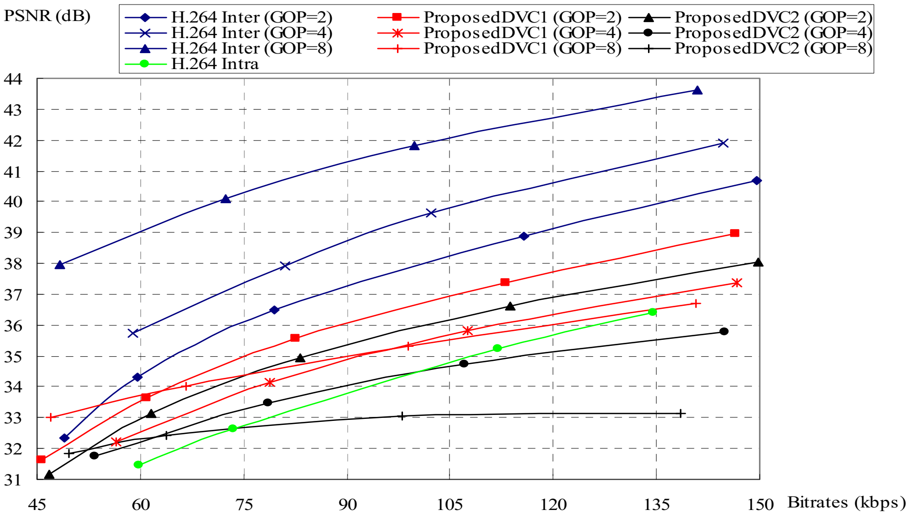


Fig. 18. RD performance for the *Carphone* sequence.

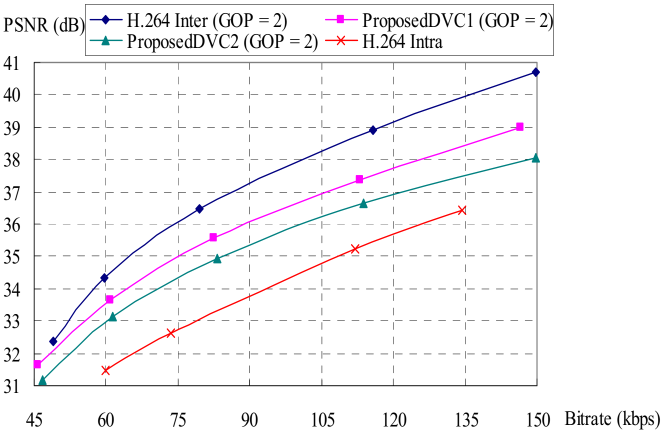
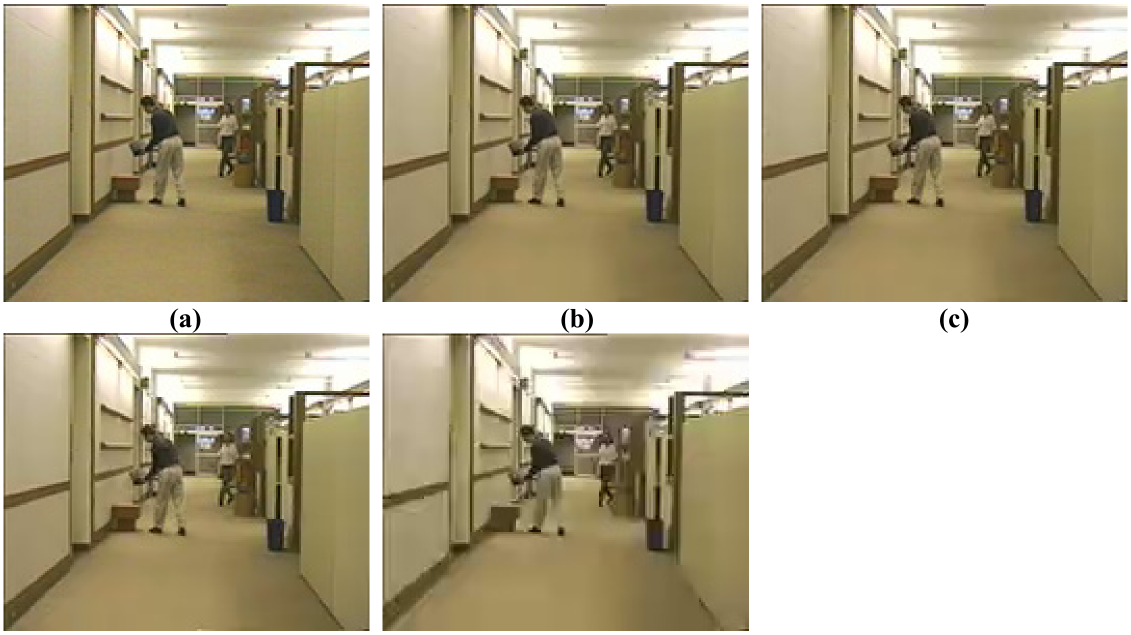


Fig. 19. RD performance for the *Carphone* sequence with *GOPSize* = 2.

It can be observed from Fig. 20, the visual qualities of the decoded frames of the two proposed codecs are better than that of the H.264/AVC intraframe coding, and comparable with those of the H.264/AVC interframe coding and the uncompressed frame for the *Hall Monitor* sequence with *GOPSize* = 4.



(a) **(b)** **(c)**

(d) **(e)**

Fig. 20. The 97th frame for the *Hall monitor* sequence with *GOPSize* = 4**: (a) the uncompressed frame; (b) H.264/AVC interframe coding at bitrate =93.05 kbps (PSNR = 39.39 dB); (c) ProposedDVC1 at bitrate = 90.36 kbps (PSNR = 38.02 dB); (d) ProposedDVC2 at bitrate = 90.85 kbps (PSNR = 37.70 dB); and (e) H.264/AVC intraframe coding at bitrate = 90.22 kbps (PSNR = 31.93 dB).**

In summary, the two proposed Wyner-Ziv video codecs with *GOPSize* = 4 are more suitable for video sequences with slow or slow/middle-motions while those with *GOPSize* = 2 are more suitable for fast-motion video sequences. The two proposed codecs with *GOPSize* larger than 2 are not suitable for fast-motion video sequences with large, global motions due to the fact that only few blocks can be skipped in ProposedDVC1 and larger amounts of Wyner-Ziv bits will be generated in ProposedDVC2. On the other hand, based on the motion vector accuracy analysis shown in Figs. 5-7 and RD performances shown in Figs. 12-19 for ProposedDVC1, it can be observed that the motion vector accuracy, *i.e.*, the side information quality, indeed dominates the whole coding performance. Although there are performance gaps between each of the two proposed codecs and the H.264/AVC interframe coding, it is worth noting that the computational complexities of the two proposed encoders are significantly lower than that of the H.264/AVC interframe encoder. This is because the H.264/AVC interframe encoder will perform complex motion estimation. As the computational complexity of the H.264/AVC interframe encoder performing motion estimation is much higher than those of the two proposed encoders, in order to make the comparisons as fair as possible, the same GOP size is used and the number of reference frames is fixed to 1 during RD performance comparisons. However, the computational complexity of the employed H.264/AVC interframe encoder is still much higher than those of the two proposed encoders. On the other hand, although the performance of ProposedDVC2 is usually (slightly) worse than that of ProposedDVC1, the computational complexity of the decoder in ProposedDVC2 is significantly lower than that in ProposedDVC1.

V. Conclusions and Future Works

In this paper, a Wyner-Ziv video codec with coding mode-aided motion compensation at the decoder (ProposedDVC1) and a Wyner-Ziv video codec based on the robust media hashing (ProposedDVC2) were proposed. ProposedDVC1 is with light encoder and heavy decoder, and has the following characteristics: (a) for each block, a large amount of candidate blocks are evaluated based on some criteria derived from the RS decoding and the best neighborhood matching to find the best candidate block as the side information; (b) ECC decoding is applied to participate in generating side information; (c) no feedback channel is required. ProposedDVC2 is with light encoder and light decoder, and has the following characteristics: (a) no motion-compensated interpolation/extrapolation is performed at both the encoder and the decoder; (b) no ECC is applied; (c) no feedback channel is required. The two proposed Wyner-Ziv video codecs have been shown to present significant gains over the H.264/AVC intraframe coding while having comparable encoding complexity. Unavoidably, there is still a performance gap from the H.264/AVC interframe coding due to the H.264/AVC interframe coding performs complex motion estimation at the encoder.

ProposedDVC1 is suitable for a scenario where the decoder can support high computational capability. For example, in a video sensor network, there may be thousands of low-complexity encoder devices (video sensors) and only one or a few high-complexity decoder devices (decoding center). ProposedDVC2 is suitable for a scenario where both the encoder and decoder are with low-complexity restrictions. For example, a pair of wireless mobile camera phones can communicate with each other directly without intermediate transcoder support. For future research, the two proposed Wyner-Ziv video codecs will be extended to multiview distributed video coding scenarios [26]-[27], in which more accurate side information may be generated based on the information from multiple views. In the multiview DVC methods [26]-[27], the basic paradigm for single view DVC similar to [7], [13] is naturally extended to multiview DVC. Video frames from different views are encoded independently and decoded jointly. That is, inter-view communication is not allowed at the encoder. However, we observe that if few data exchanges (*e.g.*, hash information exchange) among views can be allowed at the encoder, more inter-view redundancies can be removed. In addition, the two proposed codecs will be applied in practical distributed environments (*e.g.*, video sensor network [28] or multihop wireless network [29]). On the other hand, the rate control, error resilience, and security issues deserve further studying.

REFERENCES

[1] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6-17, Jan. 2005.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003.

[3] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71-83, Jan. 2005.

[4] R. Puri, A. Majumdar, P. Ishwar, and K. Ramchandran, "Distributed video coding in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 94-106, July 2006.

[5] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Information Theory*, vol. IT-22, no. 1, pp. 1-10, Jan. 1976.

[6] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Information Theory*, vol. IT-19, no. 4, pp. 471-480, July 1973.

[7] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner-Ziv coding of video," in *Proc. of IEEE Int. Conf. on Image Processing*, Barcelona, Spain, Sept. 2003, pp. 869-872.

[8] A. B. B. Adikari, W. A. C. Fernando, H. K. Arachchi, and W. A. R. J. Weerakkody, "Sequential motion estimation using luminance and chrominance information for distributed video coding of Wyner-Ziv frames," *IEE Electronics Letters*, vol. 42, no. 7, pp. 398-399, March 2006.

[9] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira, "Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding," in *Proc. of IEEE Int. Conf. on Image Processing*, Atlanta, GA, USA, Oct. 2006, pp. 253-256.

[10] C. Brites, J. Ascenso, and F. Pereira, "Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities," in *Proc. of European Signal Processing Conference*, Florence, Italy, Sept. 2006.

[11] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," in *Proc. of Allerton Conf. on Communication, Control and Computing*, Allerton, USA, Oct. 2002.

[12] R. Puri and K. Ramchandran, "PRISM: a "reversed" multimedia coding paradigm," in *Proc. of IEEE Int. Conf. on Image Processing*, Barcelona, Spain, Sept. 2003, pp. 617-620.

[13] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, USA, Jan. 2004, pp. 520-528.

[14] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *Proc. of IEEE Int. Conf. on Image Processing*, Singapore, Oct. 2004, pp. 3097-3100.

[15] A. Aaron and B. Girod, "Wyner-Ziv video coding with low encoder complexity," in *Proc. of Picture Coding Symposium,* San Francisco, CA, USA, Dec. 2004.

[16] J. Ascenso, C. Brites, and F. Pereira, "Content adaptive Wyner-Ziv video coding driven by motion activity," in *Proc. of IEEE Int. Conf. on Image Processing*, Atlanta, GA, USA, Oct. 2006, pp. 605-608.

[17] Z. Li, L. Liu, and E. J. Delp, "Wyner-Ziv video coding with universal prediction," *IEEE Trans. on*

Circuits and Systems for Video Technology, vol. 16, no. 11, pp. 1430-1436, Nov. 2006.

[18] X. Artigas and L. Torres, "Iterative generation of motion-compensated side information for distributed video coding," in *Proc. of IEEE Int. Conf. on Image Processing*, Genova, Italy, Sept. 2005, pp. 833-836.

[19] L. W. Kang and C. S. Lu, "Wyner-Ziv video coding with coding mode-aided motion compensation," in *Proc. of IEEE Int. Conf. on Image Processing*, Atlanta, GA, USA, Oct. 2006, pp. 237-240.

[20] L. W. Kang and C. S. Lu, "Low-complexity Wyner-Ziv video coding based on robust media hashing," in *Proc. of IEEE Int. Workshop on Multimedia Signal Processing*, Victoria, BC, Canada, Oct. 2006, pp. 267-272.

[21] Wicker, *Error control systems for digital communication and storage*, Prentice-Hall, 1995.

[22] Z. Li, L. Liu, and E. J. Delp, "Rate distortion analysis of motion side estimation in Wyner-Ziv video coding," accepted and to appear in *IEEE Trans. on Image Processing*.

[23] C. S. Lu and H. Y. M. Liao, "Structural digital signature for image authentication: an incidental distortion resistant scheme," *IEEE Trans. on Multimedia*, vol. 5, no. 2, pp. 161-173, June 2003.

[24] C. S. Lu, S. W. Sun, C. Y. Hsu, and P. C. Chang, "Media hash-dependent image watermarking resilient against both geometric attacks and estimation attacks based on false positive-oriented detection," *IEEE Trans. on Multimedia*, vol. 8, no. 4, pp. 668-685, Aug. 2006.

[25] S. Zafar, Y. Q. Zhang, and B. Jabbari, "Multiscale video representation using multiresolution motion compensation and wavelet decomposition," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 1, pp. 24-35, Jan. 1993.

[26] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed multi-view video coding," in *Proc. of SPIE Electronic Imaging*, vol. 6077, San Jose, CA, USA, Jan. 2006.

[27] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Fusion-based multiview distributed video coding," in *Proc. of ACM Int. Workshop on Video Surveillance and Sensor Networks*, Santa Barbara, CA, USA Oct. 2006.

[28] W. C. Feng, E. Kaiser, W. C. Feng, and M. L. Baillif, "Panoptes: scalable low-power video sensor networking technologies," *ACM Trans. on Multimedia Computing, Communications and Applications*, vol. 1, no. 2, pp. 151-167, May 2005.

[29] H. Wu and A. A. Abouzeid, "Energy efficient distributed JPEG2000 image compression in multihop wireless networks," in *Proc. of IEEE Workshop on Applications and Services in Wireless Networks*, Boston, MA, USA, 2004, pp. 152-160.