

Motion-Aware Mesh-Structured Trellis for Correlation Modelling Aided Distributed Multi-View Video Coding

Yongkai Huo, Tao Wang, Robert G. Maund, *Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

Abstract—A joint source-channel coding has attracted substantial attention with the aim of further exploiting the residual correlation residing in the encoded video signals for the sake of improving the reconstructed video quality. In our previous paper, a first-order Markov process model was utilized as an error concealment tool for exploiting the intra-frame correlation residing in the Wyner-Ziv (WZ) frame in the context of pixel-domain distributed video coding. In this contribution, we exploit the inter-view correlation with the aid of an inter-view motion search in distributed multi-view video coding (DMVC). Initially, we rely on the system architecture of WZ coding invoked for multi-view video. Then, we construct a novel mesh-structured pixel-correlation model from the inter-view motion vectors and derive its decoding rules for joint source-channel decoding. Finally, we benchmark the attainable system performance against the existing pixel-domain WZ coding based DMVC scheme, where the classic turbo codec is employed. Our simulation results show that substantial bitrate reductions are achieved by employing the proposed motion-aware mesh-structured correlation modelling technique in a DMVC scheme.

Index Terms—XXXXX.

I. INTRODUCTION

MULTI-VIEW Video Coding (MVC) [1] has recently attracted substantial attention in the context of both sophisticated 3D-TV and low-complexity wireless sensor network scenarios. A number of video coding techniques [2]–[4] have been developed for MVC, which typically rely on a high-complexity encoder and a low-complexity decoder. The constraint of the inter-view prediction based MVC is that all cameras of the MVC must exchange their monoscopic views with each other for inter-view prediction. What is even more challenging is that the communication between cameras must have a low-latency in delay-sensitive lip-synchronized interactive applications. These requirements are unrealistic in many applications, such as wireless video sensor networks

(WVSN) [5] for example, where both the energy and the computational complexity are constrained [6]. Hence it is necessary to shift the computationally complex tasks from the sensor to the base station (BS) or the server of the network, in addition to limiting the data exchange among the sensors. In theory, the Wyner-Ziv (WZ) [7] video coding philosophy, also known as distributed video coding (DVC) [8], is capable of assisting the sensors in relocating the computational burden to the BS, whilst simultaneously limiting the data exchange among themselves. Below, we will firstly review the WZ coding techniques for monoscopic video and for multi-view video, followed by the motivation of our proposed algorithm.

Two basic types of WZ coding structures [8] have been proposed for monoscopic video. In [9]–[11], the authors advocated a WZ codec, which is composed of an inner turbo-code-based [12] Slepian-Wolf (SW) codec [13] concatenated with an outer quantization-reconstruction component pair. More specifically, the odd-indexed video frames, namely the so-called key frames are intra-coded, while the even frames, namely the WZ frames are encoded by the WZ codec. At the receiver, the side information of the WZ frames will be estimated from their adjacent key frames for joint inter-frame decoding. However, a specific impediment of this structure is that the turbo decoder has to invoke a “request-and-decode” [9] process for the transmission of the WZ frames’ parity bits, which precludes its application in delay-sensitive services. Low density parity check (LDPC) codes were employed for distributed source coding in [14], which were the so-called rate-adaptive LDPC accumulate (LDPCA) codes and the sum-LDPC-accumulate (SLDPCA) codes. It was shown in [14] that the LDPCA codes are capable of approaching the capacity of a variety of communication channels more closely - including that of the virtual channel in DVC - than the family of turbo codes. Based on the WZ video coding structure of [11], in the European project DIStributed CODing for Video sERVICES (DISCOVER) [15] the transform-domain of the WZ frames was encoded by the LDPCA code of [14] at the transmitter. A so-called unsupervised motion learning technique was proposed in [16], which estimates the Motion Vectors (MVs) of the next video frame during the decoding of the current frame with reference to the previous reconstructed frame. This technique may be readily applied for both pixel-domain and transform-domain coding. The authors of [17] proposed a more realistic WZ video coding approach, which performs online estimation of the channel-induced noise (CN)

Manuscript received April 27, 2013; revised August 30, 2013 and October 23, 2013; accepted October 27, 2013. This work was supported in part by the EU’s Concerto Project of European Research Council’s Senior Fellow Grant and in part by the RC-U.K. under the India-U.K. Advanced Technology Centre. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stefano Tubaro.

The authors are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: yh3g09@ecs.soton.ac.uk; tw08r@ecs.soton.ac.uk; rm@ecs.soton.ac.uk; lh@ecs.soton.ac.uk).

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

Digital Object Identifier 10.1109/TIP.2013.2288913

model parameters at the decoder, which can be invoked for both pixel-domain and for transform domain WZ video codecs. Moreover, three levels of granularity were proposed by Brites and Pereira for pixel-domain WZ (PDWZ) video coding in [17], namely frame-, block- and pixel-level granularity, while both DCT-band and DCT-coefficient level granularity was proposed for transform-domain WZ (TDWZ) video coding. Then in [18] the same authors proposed an efficient encoder rate control (ERC) solution for feedback free transform-domain WZ (TDWZ) video coding. The approach of multiple side information (SI) components was proposed in [19] for improving the accuracy of SI using a single estimation mode. As a further advance, a more accurate parity rate estimator (PRE) is employed for more closely estimating the parity rate necessitated. A context-adaptive Markov random field reconstruction algorithm was proposed in [20], which exploits the spatio-temporal correlation by modelling the WZ frames. The so-called optical flow was proposed in [21] to improve the side information generation, which is exploited at the decoder side to compensate for weaknesses of block-based methods. A successive bit-plane-by-bit-plane refinement of the SI estimation algorithm was investigated in [22], leading to successively improved SI. The same authors proposed a motion-compensated multi-hypothesis prediction technique for medical imaging applications. Moreover, techniques conceived for multiple SI generation were proposed in [23], [24], where additional information may be used for improving the estimated SI of WZ frames. Apart from the WZ coding architecture mentioned above, another DVC architecture was proposed in [25], which allows the flexible sharing of complexity between the encoder and decoder.

Below, we review the WZ coding techniques designed for multi-view video sequences. In [26], the authors proposed a novel framework for the distributed compression of multi-view images, which was based on a tree-structured compression algorithm that guaranteed an optimal rate-distortion performance for specific video signals. Yeo and Ramchandran extended their previous PRISM framework detailed in [25] into distributed MVC in [27], [28], where the achievable error-resilience was studied in wireless scenarios. Two alternative models were proposed for exploiting the inter-view correlation, namely the view-synthesis-based correlation model and the disparity-based correlation model. The view-synthesis-based correlation model requires at least two other camera views and relies on both disparity estimation and view interpolation, while the disparity-based correlation model requires only a single additional camera view. In [29], the authors extended the WZ framework proposed in [11] into distributed MVC. At the encoder side, a wavelet-based WZ scheme was proposed for compressing each camera's view independently, where all coefficients were organized as proposed in the SPIHT scheme of [30] on a bitplane by bitplane basis. At the decoder side, a flexible prediction technique was proposed for generating the required SI, which jointly exploited both the temporal and inter-view correlations. The common benefit of the frameworks advocated in [28], [29] is that inter-camera communication is completely avoided and the computational complexity of the encoder was shifted to the decoder.

The iterative source-channel decoding (ISCD) [31] principle can be utilized for improving the system's performance by exploiting the residual correlation within the source signals. Moreover, a first-order Markov model based error concealment technique was developed in our previous work [32], where the intra-frame correlation was exploited for achieving an improved reconstructed video quality. However, the ISCD principle has not been conceived for family of the WZ multi-view video codecs. Since the inter-view correlation is not removed by the WZ multi-view video encoder, it is beneficial to exploit the residual correlation at the receiver for the sake of reducing the required bitrate. In this treatise, we develop a novel mesh-structured source model (MSSM) based decoder for exploiting the correlation among the inter-view pixels, which will be combined with a turbo codec for performing iterative source-channel decoding (ISCD) in the context of DMVC for the sake of achieving a reduced bitrate. Against this background, our novel contributions are:

- 1) We conceive a novel mesh-structured trellis exemplified in Fig. 7 for exploiting the inter-view correlations inherent in the video signal. Furthermore, the corresponding decoding rules of this trellis are derived for the sake of performing turbo-like iterative decoding, again, by exploiting the inter-view correlations.
- 2) We apply the new-trellis based proposed technique in a DMVC system, which results in a substantial bitrate reduction.

This rest of this paper is organized as follows. Section II briefly reviews the first-order Markov modelling technique used for exploiting the intra-frame correlation. We present our DMVC system architecture in Section III. In Section IV we detail our mesh-structured source model conceived for exploiting the correlation among the inter-view pixels, which is applied in the distributed MVC system of Section III. The performance of the proposed scheme is quantified with the aid of simulations in Section V. Finally, we offer our conclusions in Section VI.

II. FIRST-ORDER MARKOV PROCESS MODEL

The *a-posteriori* probability determination technique conceived for first-order Markov processes was detailed in [31]–[33]. In this section, we will briefly introduce the technique of first-order Markov process aided decoding. Let us commence by stipulating the following assumptions:

- x_i : an m -bit pattern of pixels scanned from the original video pixels at time instant i , which is expressed as $\{x_i(0), \dots, x_i(m-1)\} = x_i \binom{m-1}{0}$;
- m : the number of bits in each m -bit pattern x_i of pixels;
- $X_m = \{0, 1, \dots, 2^m - 1\}$: the set of all possible values in an m -bit pattern x_i ;
- $x_0^t = x_0, \dots, x_t$: the bit patterns of the 1^{st} frame of the original video consisting of $(t+1)$ m -bit patterns during the time interval spanning from 0 to t ;
- $y_0^t = y_0, \dots, y_t$: potentially error-infested bit pattern of the 1^{st} frame;

The corresponding trellis of the first-order Markov process is displayed in Fig. 1, where the m -bit pattern x_i indicates the

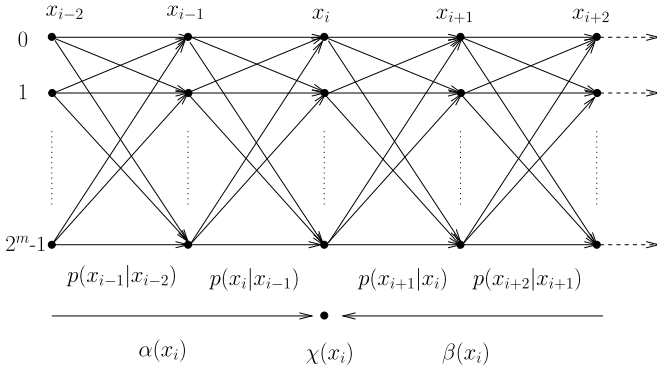


Fig. 1. Trellis of first-order Markov process for BCJR decoding, where $p(x_{i+1}|x_i)$ is the Markov transition probability.

trellis state at time instant i and the probability $p(x_{i+1}|x_i)$ indicates the transition from state x_i to state x_{i+1} . At the receiver, the *a-posteriori* probability of the m -bit pattern $x_i, x_i \in X_m$ conditioned on the specific received frame of m -bit patterns y_0, \dots, y_t may be expressed as

$$p(x_i|y_0^t) = \frac{p(x_i \wedge y_0^t)}{p(y_0^t)}, \quad (1)$$

where the joint probability $p(x_i \wedge y_0^t)$ of the m -bit pattern x_i and of the received frame y_0^t may be further formulated as [32]

$$p(x_i \wedge y_0^t) = \beta(x_i) \cdot \chi(x_i) \cdot \alpha(x_i). \quad (2)$$

It was shown in [32] that the bit-based *a-posteriori* LLR $L[x_i(k)|y_0^t]$ can be formulated as

$$L[x_i(k)|y_0^t] = \ln \frac{\sum_{\substack{x_i(k)=0 \\ x_i \in X_m}} \beta(x_i) \cdot \chi(x_i) \cdot \alpha(x_i)}{\sum_{\substack{x_i(k)=1 \\ x_i \in X_m}} \beta(x_i) \cdot \chi(x_i) \cdot \alpha(x_i)}, \quad (3)$$

where the components α, β, χ are derived in [32].

III. WYNER-ZIV CODING FOR MULTI-VIEW VIDEO: SYSTEM MODEL

Again, WZ compression techniques designed for monoscopic video have attracted substantial research attention [8]–[11], [14]–[18], [20], [25], [34], [35]. A number of contributions have also been proposed for DMVC [26]–[29]. In this section, we will detail the WZ compression philosophy shown in Fig. 2, which is invoked for distributed MVC employing our proposed MSSM-Turbo decoder. In the system of Fig. 2, there are N cameras capturing N views, respectively. We consider an array of N cameras, which employ N identical low-complexity video encoders for encoding the N camera-views independently at the transmitter and a potentially high-complexity video decoder for jointly decoding the N camera-views at the receiver. Each group of pictures (GOP) of each of the N camera views consists of a single I frame followed by a fixed number of WZ frames. The architecture of the investigated pixel-domain Wyner-Ziv

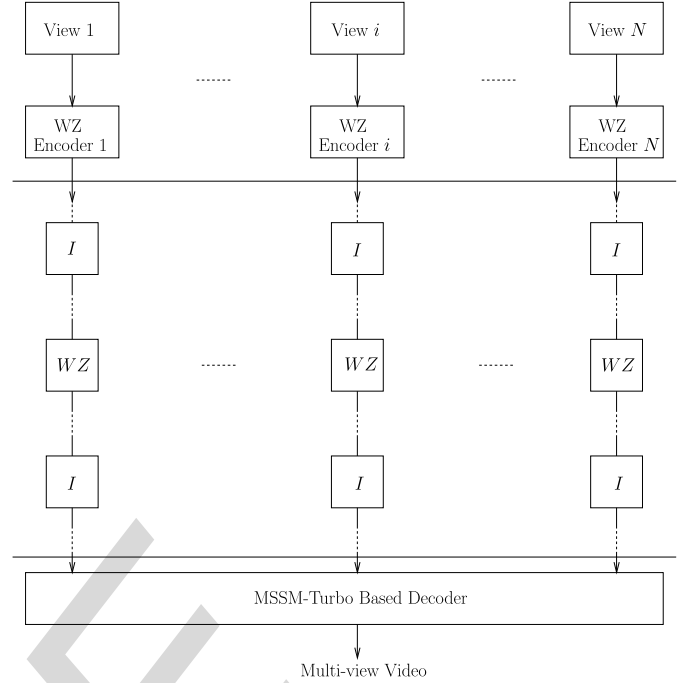


Fig. 2. The schematic compression process of Wyner-Ziv coding conceived for multi-view video.

coding system is displayed in Fig. 3, where the MSSM exploits the inter-view correlations, which cannot be removed by the multi-view Wyner-Ziv encoder, regardless, whether a pixel-domain architecture or a transform-domain architecture is employed. Hence the proposed MSSM is not limited to the family of pixel-domain architectures. The residual inter-view correlation encountered in transform-domain systems may potentially be exploited by appropriately designing the MSSM techniques, which may be part of our future research.

Generally, this treatise focuses on the MSSM-Turbo decoder of Fig. 3, while the rest of the techniques, including the WZ encoder, the motion-compensated frame interpolation (MCFI) etc., are detailed in [35]. Below, we will briefly introduce the system of Fig. 3, while the MSSM-Turbo decoder will be detailed in Section IV.

A. Transmitter

For each view, the frames are classified into two categories, namely the so-called key frames U_1, \dots, U_N of Fig. 3 and the WZ frames V_1, \dots, V_N of Fig. 3. The key frames U_1, \dots, U_N , also referred to as I frames, are intra-frame-coded by the H.264/AVC encoder and then transmitted to the receiver. Below we consider the monoscopic WZ frame V_i ($1 \leq i \leq N$) for introducing the encoding process of the WZ frames in Fig. 3. The encoding process is encapsulated in the following steps.

- The monoscopic frame V_i is quantized by a uniform quantizer at the transmitter of Fig. 3 generating the resultant m -bit monoscopic frame q_i .
- Each pixel of the m -bit quantized monoscopic frame q_i is decomposed into m bits. Then the bits from the same position of q_i constitute the most significant bit (MSB)

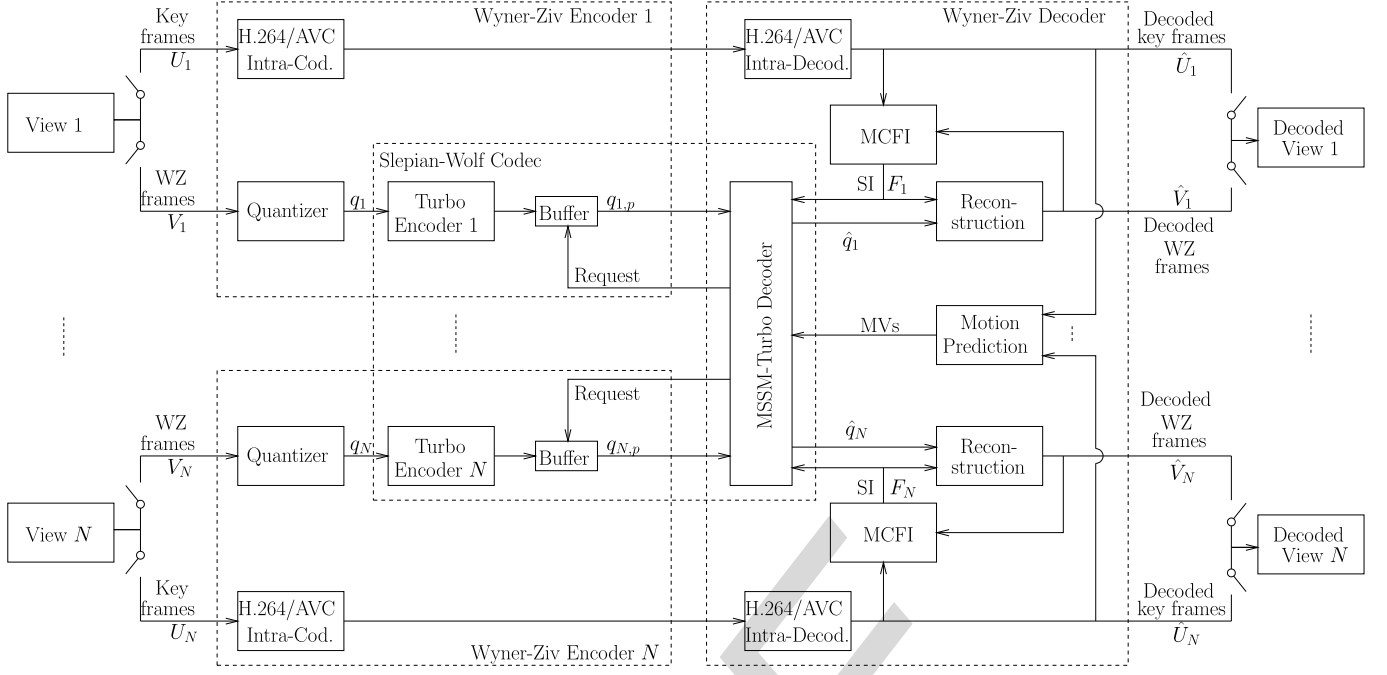


Fig. 3. System architecture of the Wyner-Ziv compression for MVC using MSSM-Turbo decoding.

plane resulting in m MSB planes. Afterwards, each of the m MSB planes will be considered as a block input, which is encoded by the turbo encoder i of Fig. 3.

- The systematic output of the turbo encoder i is directly transmitted to the receiver. The parity bits $q_{1,p}, \dots, q_{N,p}$ ¹ of Fig. 3 generated by the turbo encoder i for the different MSB planes of the WZ frame V_i may be independently buffered at the transmitter, which will then be transmitted to the receiver upon its request.

B. Receiver

The behavior of the receiver is described in Fig. 4, which will be further detailed in Section IV. At the receiver, the received bitstreams of the key frames representing the N camera views may be independently decoded for reconstructing the key frames $\hat{U}_1, \dots, \hat{U}_N$, which will then be utilized for estimating the SI for the corresponding WZ frames. Then the decoding process is listed as follows.

- For each monoscopic WZ frame, two temporally adjacent monoscopic key frames may be utilized for predicting the related soft-bit information. A number of algorithms have been proposed for this estimation process, such as for example the MCFI framework [35] shown in Fig. 3, which is invoked in our system for predicting the SI in the temporal direction.
- As observed from Fig. 4, the SI generated for the WZ frames and the received parity bits of the MSB planes related to the WZ frames will be used by the MSSM-Turbo decoder of Fig. 3 for reconstructing the WZ frames $\hat{V}_1, \dots, \hat{V}_N$.

- The flow-chart of Fig. 4 shows that when the MSSM-Turbo decoder fails to perfectly recover a MSB plane, the receiver will send a feedback flag to the transmitter for requesting more parity bits for this particular MSB plane. Again, this process is referred to as “request-and-decode”² process [9] in Fig. 4. The MSSM-Turbo decoder will be detailed in Section IV.
- As observed from Fig. 4, the “request-and-decode” process will terminate, when the bit error ratio (BER) of the MSB plane becomes lower than a preset threshold. Once all MSB planes have been decoded, the current multi-view frame may be readily reconstructed.

IV. INTER-VIEW CORRELATION MODELLING

The maximum *a-posteriori* probability (MAP) determination technique conceived for first-order Markov processes was briefly reviewed in Section II, where the intra-frame correlation was exploited using both horizontal and vertical Markov processes. However, the first-order Markov processes cannot be readily applied for modelling the inter-view correlation. In this section, we introduce the techniques invoked for exploiting the inter-view correlation of the N cameras by designing a novel trellis representation and derive its decoding rules. Below we focus on a specific multi-view WZ frame, which consists of N monoscopic video frames, whose SI is estimated by the MCFI of Fig. 3. Moreover, the trellis generation, MSSM-Turbo decoding and “request-and-decode” processes are illustrated in the flow-chart of Fig. 4. Let us commence by introducing the following notations:

- N : the number of camera views in the multi-view video;
- V_i : the original monoscopic frame of the camera view i ;

¹The subscript p indicates that $q_{1,p}, \dots, q_{N,p}$ are parity bits of quantized pixels q_1, \dots, q_N .

²Rate controller at the encoder side [18] may be investigated to avoid this feedback channel.

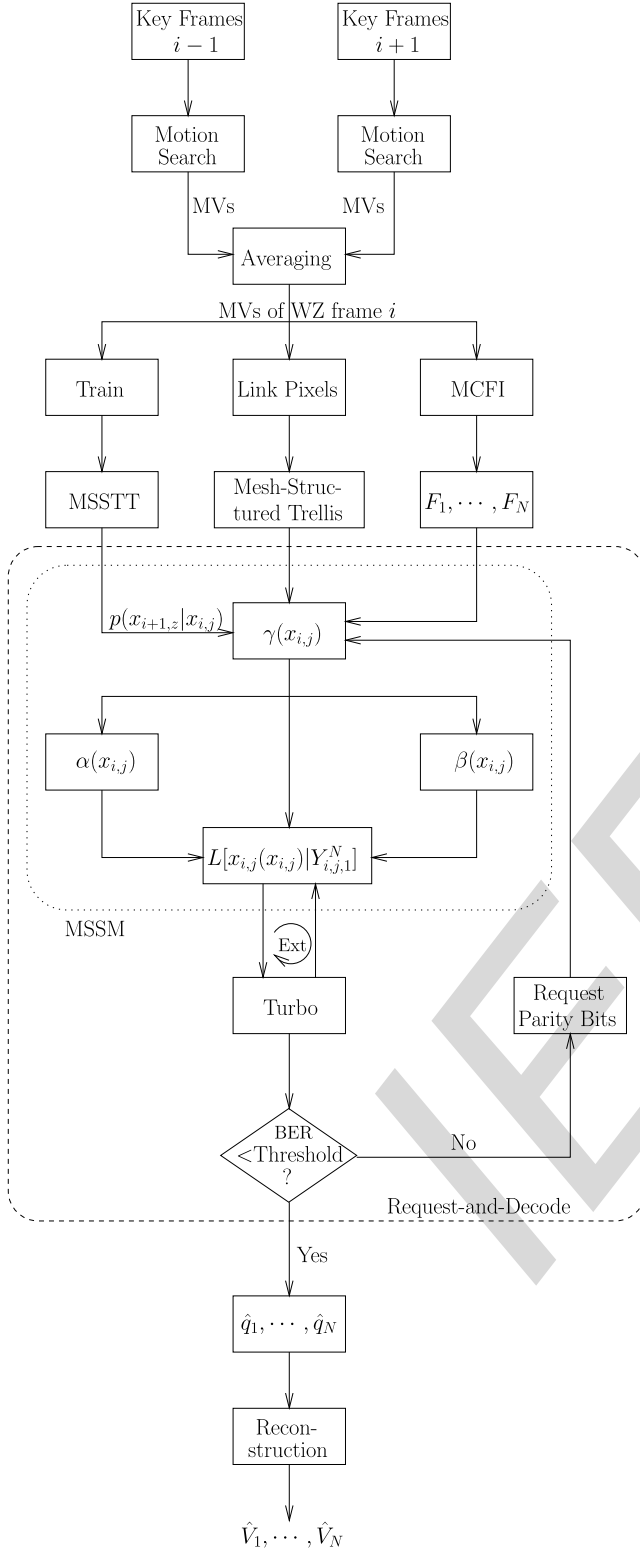
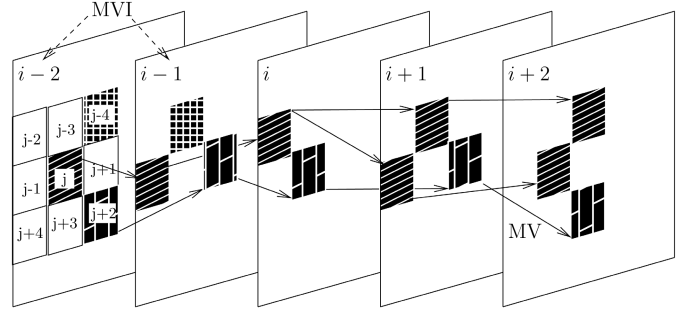


Fig. 4. Flow-chart of the receiver.

- F_i : the SI of the frame V_i , which is estimated by the MCFI block of Fig. 3;
- $x_{i,j}$: the j^{th} pixel³ in V_i , namely the pixel at position (i, j) ;

³The pixels of a 2D frame are indexed using a one-dimensional formulation, assuming that the 2D image is scanned into a single-dimensional vector.

Fig. 5. Motion based inter-view correlation, where $(j-4) \dots (j+4)$ indicate the indices of the (1×1) -pixel MBs.

- $x_{i,j}(k)$: the k^{th} bit of the pixel $x_{i,j}$.
- $y_{i,j}$: the SI of the pixel $x_{i,j}$, which is at position j of F_i ;
- m : the number of bits contained in the pixel $x_{i,j}$;
- $X_m = \{0, 1, \dots, 2^m - 1\}$: the set of legitimate values of an m -bit pattern $x_{i,j}$;
- $L_{i,j,k}$: the set of the original pixels linked with $x_{i,j}$ in camera view k .
- $L_{i,j,1}^N$: notation for the set $\bigcup_{t=1}^N L_{i,j,t}$;
- $Y_{i,j,k}$: the corresponding SI of $L_{i,j,k}$;
- $Y_{i,j,1}^N$: notation for the set $\bigcup_{t=1}^N Y_{i,j,t}$.

A. Mesh-Structured Trellis Representation

Again, each multi-view frame consists of N monoscopic camera view frames. Since the current multi-view WZ frame is not available at the receiver, the inter-view MVs of the current WZ frame are estimated from the adjacent key frames. Note that all the key frames are available at the receiver after the “H.264/AVC intra decoder” block of Fig. 3. Here we consider the scenario of $\text{GOP}=2$, but this technique may be readily extended to larger GOP scenarios. Specifically, the $(i-1)^{st}$ and $(i+1)^{st}$ key frames are utilized for estimating the MVs of the i^{th} WZ frame, as illustrated in the flow-chart of Fig. 4. The inter-view MVs of the WZ multi-view frames are generated as follows.

- The inter-view MVs of the $(i-1)^{st}$ and $(i+1)^{st}$ key frames are readily estimated using the traditional macroblock (MB) based motion search techniques [36], which is indicated by the “Motion Search” blocks of Fig. 4.
- The inter-view MVs of the i^{th} WZ frame is averaged based on that of the $(i-1)^{st}$ and $(i+1)^{st}$ key frames, as illustrated by the “Averaging” block of Fig. 4.

The structure of the inter-view “Motion Prediction” of Fig. 3 is shown in Fig. 5, where the MBs of the specific view associated with the Monoscopic View Index (MVI) i ($2 < i \leq I$) are estimated from the views $(i-1)$ and $(i+1)$. Note that in Fig. 5 we employ (1×1) -pixel “MBs” for the sake of simplifying the relevant descriptions, which the classic (8×8) -pixel MBs may be readily extended to.

Following the inter-view motion estimation process at the receiver, the pixels of a given MB of Fig. 5 may be linked to other pixels in the same position of the predicted MBs in the

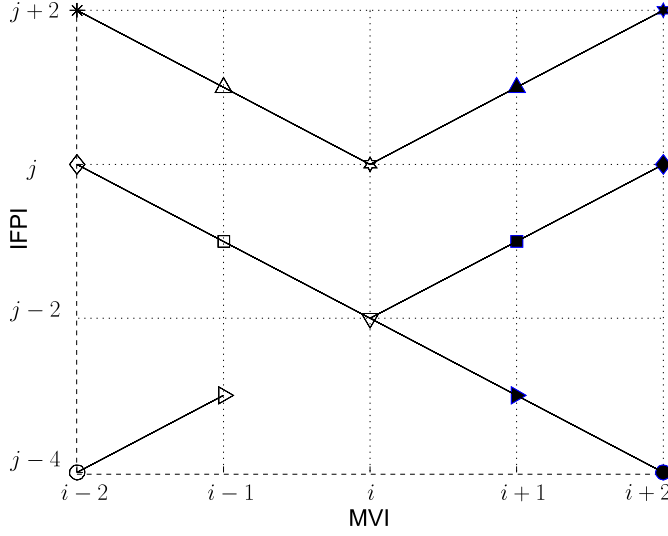


Fig. 6. The inter-view pixel correlation graph of Fig. 5, where each marker indicates a different pixel.

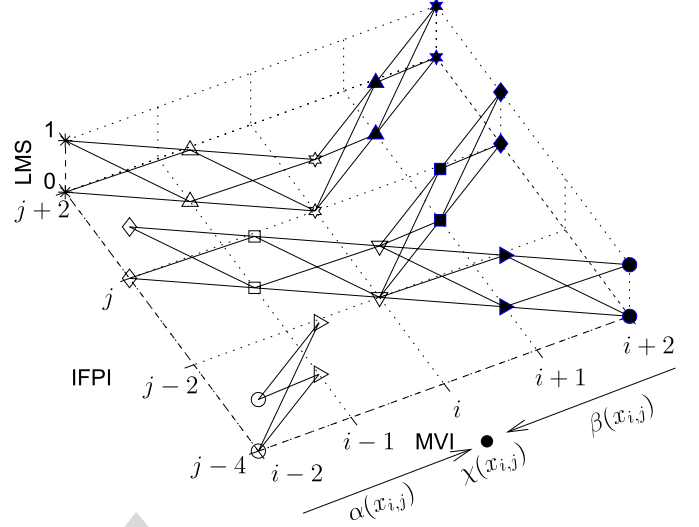


Fig. 7. Mesh-structured trellis representation of inter-view correlations for $m = 1$ -bit pixels, where the pixels of view $(i-2)$ to view $(i+2)$ are displayed.

other views, as exemplified in the inter-view pixel correlation graph of Fig. 6. Specifically, the graph of Fig. 6, indicated by the “Link Pixels” block of Fig. 4, may be created using the following steps:

- The pixels of Fig. 5 are scanned into one-dimensional Intra-Frame Pixel Indices (IFPI), namely $(j-4), \dots, (j+4)$ in view $(i-2)$ of Fig. 5. For example, each monoscopic frame having (352×288) -pixels will be indexed by $352 \cdot 288 = 72864$ one-dimensional scan-line indices.
- The coordinate axes of Fig. 6 are created by arranging for the x axis and y axis to indicate the MVI and the IFPI, respectively. Specifically, the MVI of Fig. 5 are in the range of $(i-2, \dots, i+2)$. Furthermore, we only consider the colored pixels of Fig. 5 for simplifying the related descriptions, resulting in IFPI of $(j-4, \dots, j+2)$.
- Connect all the correlated pixels portrayed in Fig. 6 using a link, where the presence of highest correlation is indicated by the MVs of Fig. 5. For example, in Fig. 6 the pixel at position (i, j) is correlated with the pixels at positions of $(i-1, j+1)$ and $(i+1, j+1)$ in Fig. 6, as indicated by the MVs of Fig. 5.

Based on the two-dimensional Markov-modelling based trellis representation developed in [32], the mesh-structured trellis of Fig. 7 may be derived from the correlation graph of Fig. 6. This trellis generation process is also indicated in Fig. 4, which is completed using the following steps:

- Each m -bit pixel has a value in the range of $[0, 2^m)$, which we refer to as the Legitimate Markov States (LMS). Then each pixel in Fig. 6 has 2^m LMS. Hence by introducing the z axis indicating the LMS, Fig. 7 may be created, where the x axis and y axis indicate the MVI and IFPI, respectively.
- Each pair of pixels connected by a direct link in Fig. 6 represents a correlated Markov-state transition. Hence we have to incorporate $(2^m \times 2^m)$ links indicating the

$(2^m \times 2^m)$ possible Markov-state transitions for these pixels in Fig. 6. Specifically, for the pixel $x_{i,j}$ within view i , the transition probability from state $x_{i,j}$ to the correlated successor state $x_{i+1,j+1}$ within view $(i+1)$ is represented by $p(x_{i+1,j+1}|x_{i,j})$, which is the state transition of the related Markov process.

Hence all nodes in Fig. 7 associated with identical MVI belong to the same view, while the nodes having both an identical MVI and SII values are Markov states for a same corresponding pixel. Furthermore, the trellis representation seen in Fig. 7 may be readily generalized both for arbitrary m -bit pixel multi-view signals and for an arbitrary number of views, which leads to 2^m LMS.

B. Trellis Decoding

Section III detailed how the SI of a multi-view WZ frame can be generated. The SI consists of floating-point values indicating the reliability of specific pixel values, which is estimated by the MCFI block of Fig. 3. For example, the SI of the bit $x_{i,j}(k)$ may be expressed in the log-likelihood ratio (LLR) format as $L[x_{i,j}(k)] = \ln \frac{L[x_{i,j}(k)=0]}{L[x_{i,j}(k)=1]}$, while the reliability of the pixel $x_{i,j}$ can be represented by $L[x_{i,j}(0)], \dots, L[x_{i,j}(m-1)]$. In Section IV-A, the pixel correlations of a multi-view WZ frame are modelled by the mesh-structured trellis of Fig. 7. Hence the SI of the multi-view WZ frame may be refined by decoding the trellis of Fig. 7, where the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [37] decoding principle may be applied.

Given the SI of the current multi-view WZ frame F_1, \dots, F_N , the pixel $x_{i,j}$ may be estimated by the *a-posteriori* probability (APP) $p(x_{i,j}|F_1, \dots, F_N)$. Furthermore, by employing our proposed MSSM detailed in Section IV-A, we have

$$p(x_{i,j}|F_1, \dots, F_N) \approx p(x_{i,j}|Y_{i,j,1}^N), \quad (4)$$

where $Y_{i,j,1}^N$ includes the SI of the pixels $L_{i,j,1}^N$, which are correlated with $x_{i,j}$. Consider the trellis of Fig. 7 for example, where we have $L_{i,j,i-2} = \{x_{i-2,j+2}\}$, $L_{i,j,i-1} = \{x_{i-1,j+1}\}$, $L_{i,j,i} = \{x_{i,j}\}$, $L_{i,j,i+1} = \{x_{i+1,j+1}\}$ and $L_{i,j,i+2} = \{x_{i+2,j+2}\}$, the pixels $L_{i,j,i-2}^{i+2}$ can be readily derived as $L_{i,j,i-2}^{i+2} = \bigcup_{t=i-2}^{i+2} L_{i,j,t}$. In the Appendix, we show that the log-likelihood ratio (LLR) version of the APP $p(x_{i,j}(k)|Y_{i,j,1}^N)$ for the bit $x_{i,j}(k)$, $0 \leq k < m$ may be formulated as

$$L[x_{i,j}(k)|Y_{i,j,1}^N] = \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}, \quad (5)$$

where $\beta(x_{i,j})$, $\chi(x_{i,j})$, $\alpha(x_{i,j})$ are defined in Eqs. (16), (10) (15). Specifically, $\beta(x_{i,j})$, $\alpha(x_{i,j})$ indicate the backward and forward oriented probability of the pixel $x_{i,j}$, while $\chi(x_{i,j})$ is the channel information of the pixel $x_{i,j}$.

C. Iterative MSSM-Turbo Decoding

A limitation of the formulas provided in Section IV-B is that they cannot be directly used for iterative decoding, since they cannot exploit the *a-priori* LLR $L[x_{i,j}(k)]$, which was generated from the *extrinsic* information gleaned from the other decoder components. To make use of the *a-priori* LLR $L[x_{i,j}(k)]$, the combined bit-wise LLR may be expressed as [31], [32]

$$\gamma_{i,j}(x_{i,j}) = \exp \sum_{k=0}^{m-1} \frac{\bar{x}_{i,j}(k)}{2} \cdot \{L[x_{i,j}(k)] + L[y_{i,j}(k)|x_{i,j}(k)]\}, \quad (6)$$

where the symbol-based m -bit information γ is the combination of the bit-wise *a-priori* LLR $L[x_{i,j}(k)]$ and of the channel information $L[y_{i,j}(k)|x_{i,j}(k)]$. We note in this context that γ of Eq. (6) contains more valuable information than the channel information χ . Hence Eq. (17) may be used for iterative joint source-channel decoding by replacing χ with γ in Eq. (6).

Similar to the BCJR decoding technique of classic turbo codes [38], the bit-based *a-posteriori* LLR $L[x_{i,j}(k)|Y_{i,j,1}^N]$ may be split into three components, namely the *a-priori* LLR $L[x_{i,j}(k)]$, the channel information $L[y_{i,j}(k)|x_{i,j}(k)]$ and the extrinsic information $L_e[x_{i,j}(k)]$. Specifically, the bit-based *a-posteriori* LLR $L[x_{i,j}(k)|Y_{i,j,1}^N]$ may be formulated as

$$L[x_{i,j}(k)|Y_{i,j,1}^N] = L[x_{i,j}(k)] + L[y_{i,j}(k)|x_{i,j}(k)] + \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \gamma^{[ext]}[x_{i,j}(k)] \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \gamma^{[ext]}[x_{i,j}(k)] \cdot \alpha(x_{i,j})}, \quad (7)$$

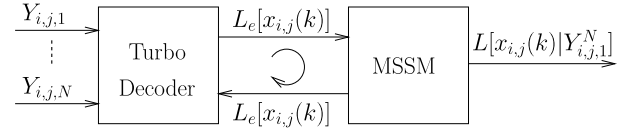


Fig. 8. The MSSM-Turbo decoding architecture of Fig. 3.

where the extrinsic information component $\gamma^{[ext]}[x_{i,j}(k)]$ may be expressed as

$$\gamma^{[ext]}[x_{i,j}(k)] = \exp \sum_{l=0, l \neq k}^{m-1} \frac{\bar{x}_{i,j}(l)}{2} \cdot \{L[x_{i,j}(l)] + L[y_{i,j}(l)|x_{i,j}(l)]\}.$$

Based on the above, the MSSM-Turbo decoding architecture of Fig. 3 is shown in Fig. 8, which terminates after I_{iter} iterations of extrinsic information exchange. Moreover, the MSSM-turbo decoding process is also detailed in the flow-chart of Fig. 4.

D. Training for Mesh-Structured State Transition

Again, the mesh-structured trellis of Fig. 7 is utilized for modelling the inter-view correlation at the receiver. Specifically, the component $p(x_{i+1,z}|x_{i,j})$ of Eqs. (15), (16) quantifies the inter-view correlation, where z and j indicate the pixel index of the views $(i-1)$ and i , respectively. Here we refer to $p(x_{i+1,z}|x_{i,j})$ as the Mesh-Structured State Transition Table (MSSTT) for simplicity. We initialize the $(2^m \times 2^m)$ -element MSSTT $T[0:2^m-1, 0:2^m-1]$ to zero values. Then we scan all the linked pixel pairs of Fig. 6. For example, when the pixel pairs s_{i-1} and s_i are scanned, the corresponding element $T[s_{i-1}, s_i]$ in the MSSTT is increased by 1. Finally, by normalizing the summation of all rows in the MSSTT $T[0:2^m-1, 0:2^m-1]$, the transition probabilities $p(x_{i+1,z}|x_{i,j})$ can be obtained, where we have $x_{i,j}, x_{i+1,z} \in [0, 2^m)$.

E. Complexity Analysis

The complexity of our proposed MSSM can be attributed to the calculation of $\gamma(x_{i,j})$ in Eq. (6), $\alpha(x_{i,j})$ in Eq. (15), $\beta(x_{i,j})$ in Eq. (16) and $L[x_{i,j}(k)|Y_{i,j,1}^N]$ in Eq. (5). As shown in Fig. 7, the trellis size is $(2^m \cdot S \cdot N)$, where S is the number of pixels in each monoscopic view. Similar to the BCJR decoding rules proposed in [37], the decoding of the $(2^m \cdot S \cdot N)$ -state trellis of Fig. 7 may be generalized into the following two stages:

- Calculation of γ , α and β : These operations are carried out across the entire trellis of Fig. 7, which imposes the complexity of m , 2^m , 2^m for each trellis state, as suggested by Eqs. (6), (15), (16), respectively. Hence the associated computational costs are on the order of $O(2^m \cdot S \cdot N \cdot m)$, $O(2^m \cdot S \cdot N)$ and $O(2^m \cdot S \cdot N)$ for γ , α and β , respectively.
- Calculation of $L[x_{i,j}(k)|Y_{i,j,1}^N]$: This operation is carried out for all the $(S \cdot N \cdot m)$ bits of a multi-view frame, which imposes a complexity of 2^m for each bit. Hence the computational cost is on the order of $O(2^m \cdot S \cdot N \cdot m)$.

TABLE I
TABLE OF PARAMETERS EMPLOYED FOR THE SIMULATIONS

Representation	YUV 4:2:0	Motion MB	8×8
Format	CIF	Generator of RSC	[11011,10011]
Bits Per Pixel	8	Turbo Code Rate	1/2
Number of Frames	100	Interleaver Length	352×288
GOP	2, 4	Puncturer	[1 1; 0 1; 1 0]
Number of Views	8	Intra-codec	H.264

Therefore, the overall complexity imposed by our proposed decoder is $O(2 \cdot 2^{2m} \cdot S \cdot N + 2 \cdot 2^m \cdot S \cdot N \cdot m)$, when decoding a multi-view frame.

V. PERFORMANCE STUDY

In this section, we present our simulation results for benchmarking the scheme introduced in Section III. Firstly, in Section V-A we will introduce the parameters of the scenario considered in our experiment. Then we will discuss our numerical results in Section V-B.

A. Scenario

In this section, we present our experimental parameters used for characterizing the convergence behavior of the proposed scheme introduced in Section III. Multi-view video sequences having 8 camera views represented in (352×288) -common intermediate format (CIF)⁴ and 4:2:0 YUV representation are employed. Moreover, the bitrate/PSNR of both the WZ and the key frames was taken into account in our average results. The distributed WZ coding scheme conceived for the multi-view video scheme of Fig. 3 operates on the basis of (352×288) -pixel blocks. More specifically, a specific MSB plane of each view is input to the turbo encoder, which consists of (352×288) bits. In other words, the interleaver length of our turbo codec is (352×288) bits. In [35], each bitplane of the MSB was transmitted separately and each bitplane was then refined based on the previously decoded bitplanes [39]. However, in our system, all MSB planes were transmitted together, which allowed us to reduce the number of “request-and-decode” processes defined in [8]. We employ a recursive systematic convolutional (RSC) encoder relying on the generator polynomials of $g_1 = 11011$, $g_2 = 10011$, which are represented as $G = [1, g_2/g_1]$, where g_1 is a feedback input and g_2 is feed-forward output. Moreover, two identical RSC encoders are employed for the turbo codec and the puncturing matrix of [1 1; 0 1; 1 0] is employed for the turbo code. The parameters employed are listed in Table I. The remaining parameters of our system were identical to those in [17], [35].

⁴We converted the multi-view video sequences into CIF representation for the sake of speeding up our simulations.

TABLE II
COMPARISON OF MSSM-TURBO AND THE BENCHMARKER TURBO FOR WYNER-ZIV CODING OF MULTIVIEW VIDEO, WHERE n_{MSB} INDICATES THE NUMBER OF MSB PLANES. $I_{\text{iter}} = 4$ IS EMPLOYED, WHERE INDICATES THE NUMBER OF DECODING ITERATIONS FOR THE MSSM-TURBO DECODER

	Turbo	MSSM-Turbo
Dimension	1	2
Number of Bits to Decode	352×288	$352 \times 288 \times 8 \times n_{\text{MSB}}$
Side Information	None	$1 \times \text{MMSTT}$

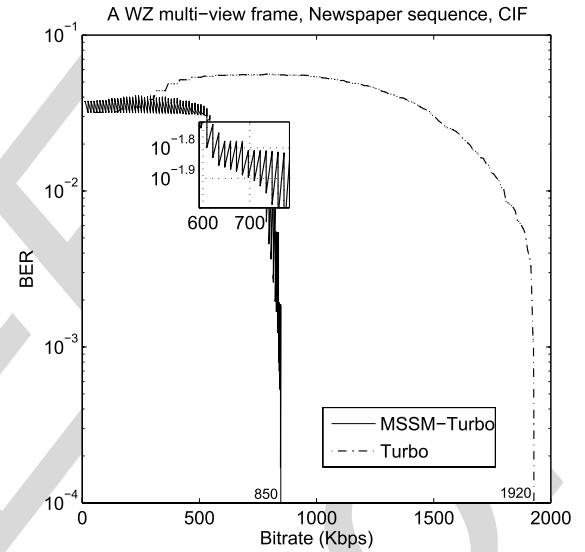


Fig. 9. BER versus bitrate comparison of the MSSM-turbo and of turbo codec for a multi-view WZ frame, where only the bits of the WZ frame are taken into account as the bitrate. Using the *Newspaper* sequence and $\text{GOP} = 2$.

TABLE III
BJNTEGAARD COMPARISON OF THE MSSM-PDWZ VERSUS THE PDWZ SCHEMES FOR THE CONSIDERED MULTI-VIEW SEQUENCES ORGANIZED IN GOPS OF 2 AND 4. “R” INDICATES “RATE”

Sequence	GOP=2		GOP=4	
	ΔR (%)	ΔPSNR (dB)	ΔR (%)	ΔPSNR (dB)
Newspaper	0.3	0.02	1.8	0.17
Leavinglaptop	6.7	0.42	15.6	0.92
Outdoor	12.6	0.97	30.7	2.27
Ballroom	13.4	0.80	27.2	1.58

Let us now compare the proposed MSSM-Turbo decoder to the classic turbo decoder. The turbo decoder is invoked for each MSB plane of each monoscopic frame, which carries (352×288) bits, scanned into a one-dimensional vector.

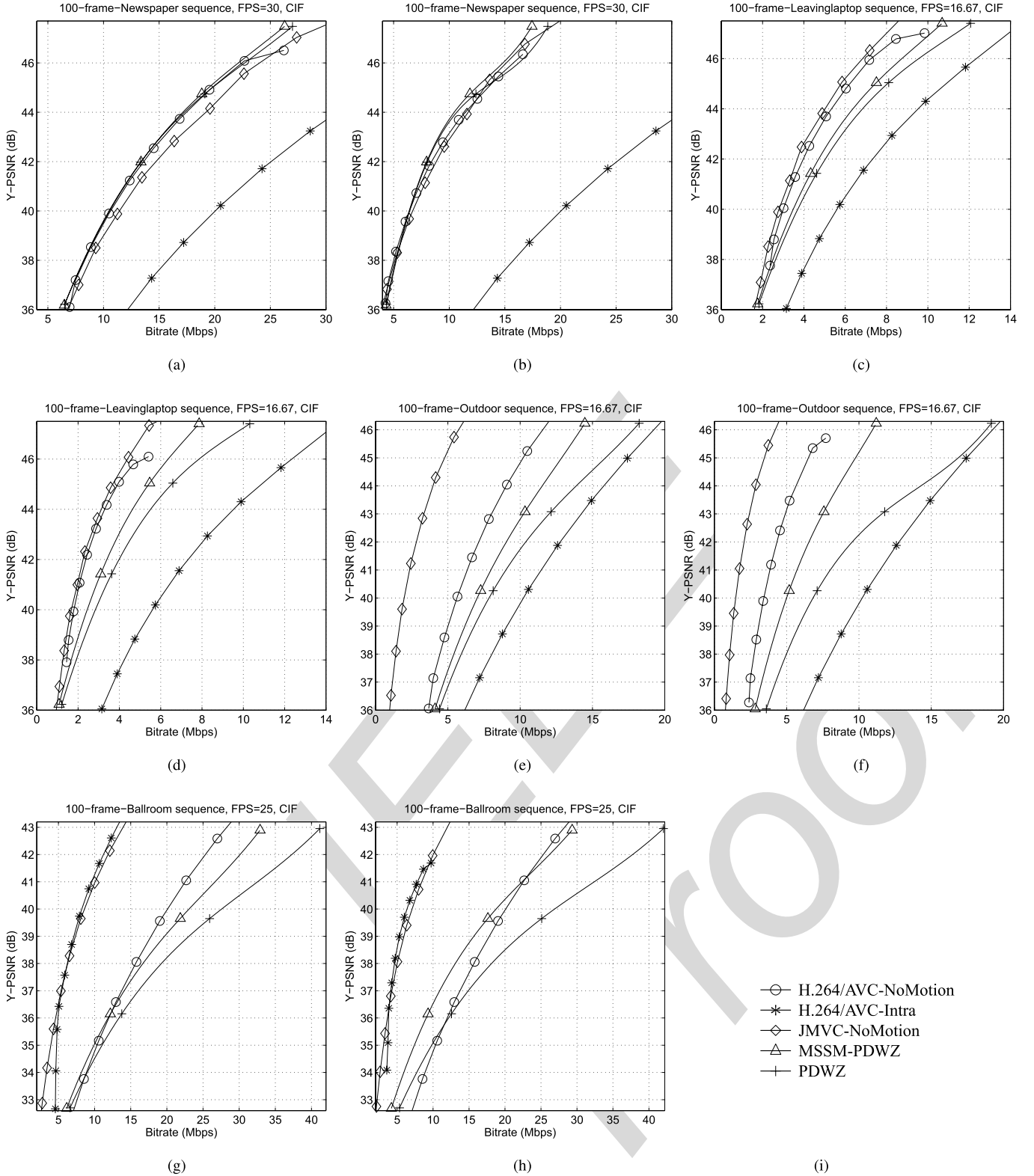


Fig. 10. Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. (a) *Newspaper*, GOP = 2. (b) *Newspaper*, GOP = 4. (c) *Leaving-laptop*, GOP = 2 (d) *Leaving-laptop*, GOP = 4. (e) *Outdoor*, GOP = 2. (f) *Outdoor*, GOP = 4. (g) *Ballroom*, GOP = 2. (h) *Ballroom*, GOP = 4. (i) Benchmarks.

Let n_{MSB} be the number of MSB planes. The MSSM-Turbo decoder is invoked for all MSB planes of each multi-view frame, which carries $(352 \times 288 \times 8 \times n_{\text{MSB}})$ bits and it is

arranged into $I = 8$ scanlines or vectors. Furthermore, the MSSM-Turbo scheme relies on the corresponding MSSTT, which has $(2^{n_{\text{MSB}}} \times 2^{n_{\text{MSB}}})$ elements. Specifically, for the



Fig. 11. Comparison of original WZ frames and the related estimated SI frames at the receiver. From left to right are *Newspaper*, *Leaving-laptop*, *Outdoor* and *Ballroom* sequences, respectively. The top row and the bottom rows list the original and the estimated SI frames, respectively.

simulations associated with $m = 4$, the size of the MSSTT SI was (16×16) floating-point values for 100 CIF frames. Our detailed comparisons are listed in Table II. Note that the MSSTT may be estimated from the adjacent I frames of the current WZ frame.

B. Numerical Results

In this section, we present our numerical results for benchmarking the proposed MSSM-PDWZ scheme's performance against that of the PDWZ system of [17] for four multi-view sequences associated with different motion features, namely the *Newspaper* sequence, the *Leaving-laptop* sequence [40], the *Outdoor* sequence [40] and the *Ballroom* sequence [41], scanned in FPSs of 30, 16.67, 16.67, 25, respectively. The rate-distortion (RD) results recorded for both the PDWZ and MSSM-PDWZ schemes in Fig. 3 were parameterized by the number of WZ coded bitplanes for $m = 1, 2, 3$ or 4 MSB planes, because this configuration has been widely adopted in the pixel-domain WZ video coding literature [17]. This was arranged by invoking the uniform quantizers shown in Fig. 3. The RD results of the test sequences coded by the H.264/AVC codec [42] are provided below as usual, both in the associated intra-frame encoding mode and in the motion compensation dispensed mode in conjunction with GOP periods of 2 and 4. Both these modes were selected by appropriately adjusting the encoding parameters of the H.264/AVC reference software JM [43]. Furthermore, the performance of the multi-view video codec JMVC operating without motion estimation and using GOP periods of 2 and 4 was also provided.

1) *BER Characteristics*: The BER comparison of the MSSM-turbo and turbo codec schemes is displayed in Fig. 9, where the x axis represents the bitrate of the multi-view WZ

frame. Note that the bitrate of a video sequence consists of the bitrate of the WZ frames and the bitrate of the key frames. Here we only count the bitrate of the WZ frames for the sake of providing further insights into our system's behavior.

For the MSSM-turbo decoder, the BERs of both the MSSM and of the turbo decoder components are provided. More specifically, multiple BER values are plotted for the MSSM-turbo decoder for each "Bitrate" abscissa value in Fig. 9. This results in a wave-shaped, fluctuating curve for the MSSM-turbo decoder. Alternatively, for each "Bitrate" value, the MSSM-turbo is capable of further reducing the BER upon the turbo codec during the ISCD process. Observe from Fig. 9 that BER of the WZ frame using the MSSM-turbo decoder becomes vanishingly low at about 850 Kbps, while that of the turbo decoder vanishes at 1920 Kbps. The MSSM-turbo decoder requires 1070 less Kbps for achieving a BER of 10^{-4} . This observation is due to the fact that MSSM is capable of further reducing the BER by exploiting the residual redundancy within the WZ frame by iteratively exchanging extrinsic information with the turbo decoder.

2) *Rate-Distortion Characteristics*: The simulation results recorded for the four sequences are displayed in Fig. 10, while the Bjntegaard comparison of the MSSM-PDWZ versus the PDWZ schemes for the four multi-view sequences considered and organized in GOPs of 2 and 4 is provided in Table III. For the MSSM-PDWZ, Fig. Fig. 10 shows that the $\text{GOP} = 4$ scenarios outperforms the $\text{GOP} = 2$ regime for the *Newspaper*, *Leavinglaptop* and *Outdoor* sequences, while the opposite trends were observed for the *Ballroom* sequence. This is due to the fact that less bits are required for reconstructing the WZ frames than that of the key frames, when the SI for the WZ frames can be accurately estimated. Table III shows

that the bitrate reduction ratio increases in the sequence order of Newspaper, Leavinglaptop, Outdoor and Ballroom. Similar trends are observed in terms of the PSNR reduction attained. The reason for this trend is that the receiver is more unlikely to be able to estimate the SI frame accurately from the received key frames, while our MSSM-PDWZ scheme is capable of effectively concealing the errors, which is an explicit benefit of our MSSM-Turbo decoder.

The comparison of the original frames to the corresponding estimated SI frames is displayed in Fig. 11 for the *Newspaper*, *Leavinglaptop*, *Outdoor* and *Ballroom* sequences, respectively. Observe that the receiver fails to estimate the SI frames of the higher-motion sequences, namely of the *Outdoor* and the *Ballroom* sequences, as precisely as for the lower-motion sequences, namely for the *Newspaper* and the *Leaving-laptop* sequences. We may conclude that our proposed MSSM technique is capable of reducing the bitrate more substantially for the higher-motion sequences, where the SI of key frames cannot be accurately estimated at the receiver.

VI. CONCLUSION

In this paper, we firstly extended the WZ coding techniques for monoscopic video into a Wyner-Ziv coded multi-view video system. Then we conceived the techniques for constructing a novel mesh-structured pixel correlation model from the inter-view MVs and derived its decoding rules. Furthermore, by incorporating the MSSM scheme into WZ video coding of multi-view video, we were able to substantially reduce the bitrate compared to that of the PDWZ benchmarker systems.

Our future work will focus on developing techniques for exploiting both the pixel correlation among different views and the pixel correlation within camera views for the sake of further reducing the required bitrate.

APPENDIX

Let us initially follow the procedure of the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [37] algorithm based determination rule of the MAP decoder for deriving the APP $p(x_{i,j}|Y_{i,j,1}^N)$ of Eq. (4). The APP $p(x_{i,j}|Y_{i,j,1}^N)$ of the m -bit pattern $x_{i,j}, x_{i,j} \in X_m$ conditioned on the specific $Y_{i,j,1}^N$ values may be expressed as

$$p(x_{i,j}|Y_{i,j,1}^N) = \frac{p(x_{i,j} \wedge Y_{i,j,1}^N)}{p(Y_{i,j,1}^N)}. \quad (8)$$

The joint probability $p(x_{i,j} \wedge Y_{i,j,1}^N)$ of the m -bit pattern $x_{i,j}$ in Eq. (8) and that of SI $Y_{i,j,1}^N$ may be further formulated as

$$\begin{aligned} & p(x_{i,j} \wedge Y_{i,j,1}^N) \\ &= p(x_{i,j} \wedge Y_{i,j,i} \wedge Y_{i,j,1}^{i-1} \wedge Y_{i,j,i+1}^N) \\ &= p(Y_{i,j,i+1}^N | x_{i,j}) \cdot p(Y_{i,j,i} | x_{i,j}) \cdot p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\ &= p(Y_{i,j,i+1}^N | x_{i,j}) \cdot p(y_{i,j} | x_{i,j}) \cdot p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\ &= \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j}). \end{aligned} \quad (9)$$

In Eq. (9), the symbol-based channel information $\chi(x_{i,j}) = p(y_{i,j}|x_{i,j})$ may be calculated from the bit-based channel information as

$$\chi(x_{i,j}) = C_{\chi_{i,j}} \cdot \exp \sum_{k=0}^{m-1} \frac{x_{i,j}(k)}{2} \cdot L[y_{i,j}(k)|x_{i,j}(k)], \quad (10)$$

where $C_{\chi_{i,j}}$ is a normalization factor, which solely depends on $y_{i,j}$. Furthermore, similar to the forward recursion calculation in the BCJR algorithm, the component $\alpha(x_{i,j})$ in Eq. (9) may be formulated as

$$\begin{aligned} & \alpha(x_{i,j}) \\ &= p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} p(Y_{i,j,i-1} \wedge Y_{i,j,1}^{i-2} \wedge x_{i,j} \wedge L_{i,j,i-1}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} p(Y_{i,j,i-1} \wedge x_{i,j} | L_{i,j,i-1}) \cdot p(Y_{i,j,1}^{i-2} \wedge L_{i,j,i-1}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} p(Y_{i,j,i-1} | L_{i,j,i-1}) \cdot p(x_{i,j} | L_{i,j,i-1}) \\ &\quad \cdot p(Y_{i,j,1}^{i-2} \wedge L_{i,j,i-1}). \end{aligned} \quad (11)$$

Note that, given the original pixel set $L_{i,j,i-1}$ the soft pixels within $Y_{i,j,i-1}$ are independent to each other, hence the item $p(Y_{i,j,i-1} | L_{i,j,i-1})$ in Eq. (11) may be expressed as

$$p(Y_{i,j,i-1} | L_{i,j,i-1}) = \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} \chi(x_{i-1,z}). \quad (12)$$

Then the forward recursion calculation may be further formulated as

$$\begin{aligned} & \alpha(x_{i,j}) = \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} [p(y_{i-1,z} | x_{i-1,z}) \\ &\quad \cdot p(Y_{i,j,i-1}^{i-2} \wedge x_{i-1,z})] \cdot p(x_{i,j} | L_{i,j,i-1}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} [\chi(x_{i-1,z}) \cdot \alpha(x_{i-1,z})] \cdot p(x_{i,j} | L_{i,j,i-1}). \end{aligned} \quad (13)$$

Furthermore, by assuming the pixels are independent to each other the item $p(x_{i,j} | L_{i,j,i-1})$ in Eq. (13) may be approximated as follows

$$\begin{aligned} & p(x_{i,j} | L_{i,j,i-1}) = \frac{p(L_{i,j,i-1} | x_{i,j}) \cdot p(x_{i,j})}{p(L_{i,j,i-1})} \\ &\approx \frac{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i-1,z} | x_{i,j}) \cdot p(x_{i,j})}{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i-1,z})} = \frac{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i,j} | x_{i-1,z})}{p(x_{i,j})^{|L_{i,j,i-1}|-1}}. \end{aligned} \quad (14)$$

Then by substituting Eq. (14) into the Eq. (13), $\alpha(x_{i,j})$ may be approximated as

$$\alpha(x_{i,j}) \approx \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} \left\{ \sum_{\substack{x_{i-1,z} \in \\ X_m}} \chi(x_{i-1,z}) \cdot \alpha(x_{i-1,z}) \cdot p(x_{i,j}|x_{i-1,z}) \right\} / p(x_{i,j})^{L_{i,j,i-1}-1}. \quad (15)$$

Similar to $\alpha(x_{i,j})$, the backward recursion calculation $\beta(x_{i,j})$ in Eq. (9) can be formulated as

$$\begin{aligned} \beta(x_{i,j}) &= p(Y_{i,j,i+1}^N \wedge x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m^{L_{i,j,i+1}}}} p(Y_{i,j,i+2}^N \wedge Y_{i,j,i+1} \wedge L_{i,j,i+1} | x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m^{L_{i,j,i+1}}}} p(Y_{i,j,i+2}^N | L_{i,j,i+1}) \cdot p(Y_{i,j,i+1} | L_{i,j,i+1}) \\ &\quad \cdot p(L_{i,j,i+1} | x_{i,j}) \\ &= \prod_{\substack{x_{i+1,z} \in \\ L_{i,j,i+1}}} \left\{ \sum_{\substack{x_{i+1,z} \in \\ X_m}} [\beta(x_{i+1,z}) \cdot \chi(x_{i+1,z}) \cdot p(x_{i+1,z} | x_{i,j})] \right\}. \end{aligned} \quad (16)$$

Finally, the determination of the bit-based APP LLRs $L[x_{i,j}(k) | Y_{i,j,1}^N]$ may be formulated as

$$\begin{aligned} L[x_{i,j}(k) | Y_{i,j,1}^N] &= \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} p(x_{i,j} | Y_{i,j,1}^N)}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} p(x_{i,j} | Y_{i,j,1}^N)} \\ &= \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}, \end{aligned} \quad (17)$$

where the Jacobian logarithm [38] can be readily applied for deriving the log-domain representation of our algorithm.

REFERENCES

- [1] M. Flierl and B. Girod, "Multiview video compression," *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 66–76, Nov. 2007.
- [2] Y. Chen, M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj, and H. Li, "Coding techniques in multiview video coding and joint multiview video model," in *Proc. PCS*, May 2009, pp. 1–4.
- [3] J. Zhang, M. Hannuksela, and H. Li, "Joint multiview video plus depth coding," in *Proc. 17th IEEE ICIP*, Sep. 2010, pp. 2865–2868.
- [4] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.
- [5] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.
- [6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM Int. Workshop WSN*, New York, NY, USA, Sep. 2002, pp. 88–97.
- [7] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.
- [8] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.
- [9] A. Aaron, R. Zhang, and B. Girod, "Wyner–Ziv coding of motion video," in *Proc. Conf. Rec. 36th Asilomar Conf. Signals, Syst. Comput.*, vol. 1, Nov. 2002, pp. 240–244.
- [10] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner–Ziv coding of video," in *Proc. ICIP*, vol. 3, Sep. 2003, pp. 869–872.
- [11] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner–Ziv coding for video: Applications to compression and error resilience," in *Proc. DCC*, Mar. 2003, pp. 93–102.
- [12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. ICC*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [13] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
- [14] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Process.*, vol. 86, no. 11, pp. 3123–3130, Nov. 2006.
- [15] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Oualet, "The DISCOVER codec: Architecture, techniques and evaluation," in *Proc. Picture Coding Symp.*, vol. 6, Lisbon, Portugal, Nov. 2007, pp. 1103–1120.
- [16] D. Varodayan, D. Chen, M. Flierl, and B. Girod, "Wyner–Ziv coding of video with unsupervised motion vector learning," *Signal Process., Image Commun.*, vol. 23, no. 5, pp. 369–378, Jun. 2008.
- [17] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1177–1190, Sep. 2008.
- [18] C. Brites and F. Pereira, "An efficient encoder rate control solution for transform domain Wyner–Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1278–1292, Sep. 2011.
- [19] X. Huang, C. Brites, J. Ascenso, F. Pereira, and S. Forchhammer, "Distributed video coding with multiple side information," in *Proc. PCS*, May 2009, pp. 1–4.
- [20] Y. Zhang, H. Xiong, Z. He, S. Yu, and C. W. Chen, "Reconstruction for distributed video coding: A context-adaptive Markov random field approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1100–1114, Aug. 2011.
- [21] H. V. Luong, L. Raket, X. Huang, and S. Forchhammer, "Side information and noise learning for distributed video coding using optical flow and clustering," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4782–4796, Dec. 2012.
- [22] N. Deligiannis, J. Barbarien, M. Jacobs, A. Munteanu, A. Skodras, and P. Schelkens, "Side-information-dependent correlation channel estimation in hash-based distributed video coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1934–1949, Apr. 2012.
- [23] F. Verbist, N. Deligiannis, M. Jacobs, J. Barbarien, P. Schelkens, and A. Munteanu, "Maximum likelihood motion compensation for distributed video coding," *Integr. Comput., Aided Eng.*, vol. 19, pp. 215–227, Jan. 2012.
- [24] F. Verbist, N. Deligiannis, M. Jacobs, J. Barbarien, P. Schelkens, A. Munteanu, et al., "Probabilistic motion-compensated prediction in distributed video coding," *Multimedia Tools Appl.*, vol. 66, no. 3, pp. 1–26, Mar. 2012.
- [25] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2436–2448, Oct. 2007.
- [26] N. Gehrig and P. Dragotti, "DIFFERENT—Distributed and fully flexible image encoders for camera sensor networks," in *Proc. IEEE ICIP*, vol. 2, Sep. 2005, pp. 690–693.
- [27] C. Yeo, J. Wang, and K. Ramchandran, "View synthesis for robust distributed video compression in wireless camera networks," in *Proc. IEEE ICIP*, vol. 3, Sep./Oct. 2007, pp. 21–24.
- [28] C. Yeo and K. Ramchandran, "Robust distributed multiview video compression for wireless camera networks," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 995–1008, Apr. 2010.
- [29] X. Guo, Y. Lu, F. Wu, D. Zhao, and W. Gao, "Wyner–Ziv-based multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 713–724, Jun. 2008.

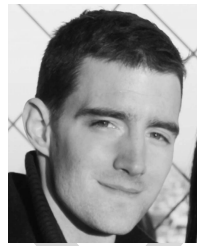
- [30] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.
- [31] M. Adrat and P. Vary, "Iterative source-channel decoding: Improved system design using EXIT charts," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 928–941, Jan. 2005.
- [32] Y. Huo, T. Wang, R. G. Maunder, and L. Hanzo, "Iterative source and channel decoding relying on correlation modelling for wireless video transmission," *IET Commun.*, vol. 7, no. 14, pp. 1465–1475, Sep. 2013.
- [33] Y. Huo, C. Zhu, and L. Hanzo, "Spatio-temporal iterative source-channel decoding aided video transmission," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1597–1609, May 2013.
- [34] A. Aaron, R. Zhang, and B. Girod, "Wyner–Ziv coding of motion video," in *Proc. 36th Asilomar Conf. Signals, Syst. Comput. Conf. Rec.*, vol. 1, Nov. 2002, pp. 240–244.
- [35] C. Brites, "Exploiting correlation noise modeling in Wyner–Ziv video coding," Ph.D. dissertation, Inst. Superior Técnico, Tech. Univ. Lisbon, Lisbon, Portugal, Mar. 2011.
- [36] Y.-W. Huang, C.-Y. Chen, C.-H. Tsai, C.-F. Shen, and L.-G. Chen, "Survey on block matching motion estimation algorithms and architectures with new results," *J. VLSI Signal Process. Syst.*, vol. 42, pp. 297–320, Mar. 2006.
- [37] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.
- [38] L. Hanzo, T. Liew, B. Yeap, and R. Tee, *Turbo Coding, Turbo Equalisation and Space-Time Coding*. New York, NY, USA: Wiley, 2011.
- [39] S. Cheng and Z. Xiong, "Successive refinement for the Wyner–Ziv problem and layered code design," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3269–3281, Aug. 2005.
- [40] *FhG-HHI 3DV Data* [Online]. Available: <http://sp.cs.tut.fi/mobile3dvtv/video-plus-depth/>
- [41] Mitsubishi Electric Research Lab. *MERL Multiview Video Sequence*, Cambridge, MA, USA [Online]. Available: <http://ftp.merl.com/pub/avetro/mvc-testseq>
- [42] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, *ITU-T Rec. H.264/ISO/IEC 14496-10 AVC: Advanced Video Coding for Generic Audiovisual Services*, Geneva, Switzerland, Mar. 2010.
- [43] *H.264/AVC Reference Software* [Online]. Available: <http://iphome.hhi.de/suehring/tml/>



Yongkai Huo received the B.Eng. degree with distinction in computer science and technology from the Hefei University of Technology, Hefei, China, in 2006, and the M.Eng. degree in computer software and theory from the University of Science and Technology of China, Hefei, in 2009. He is currently pursuing the Ph.D. degree with the Communications, Signal Processing and Control Group, School of Electronics and Computer Science, University of Southampton, Southampton, U.K. He received a scholarship under the China-U.K. Scholarships for Excellence Programme. His research interests include distributed video coding, multiview video coding, robust wireless video streaming, and joint source-channel decoding.



Tao Wang received the B.S. degree in information engineering from the University of Science and Technology of Beijing, Beijing, China, in 2006, and the M.Sc. degree in communication from the University of Southampton, Southampton, U.K., in 2008. He is currently pursuing the Ph.D. degree with the Communications Research Group, Electronics and Computer Science, University of Southampton, Southampton, U.K. His current research interests include joint source/channel coding and distributed video coding.



Robert G. Maunder has been studied with Electronics and Computer Science, University of Southampton, U.K., since October 2000. He received the B.Eng. (Hons.) degree in electronic engineering in July 2003, and the Ph.D. degree in wireless communications and a lectureship in December 2007. His research interests include joint source/channel coding, iterative decoding, irregular coding, and modulation techniques. He has published a number of IEEE papers.



Lajos Hanzo (F'04) received the Degree in electronics in 1976, the Doctorate degree in 1983, and the D.Sc. degree in 2004.

He has held various research and academic posts in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the Chair in telecommunications. He has successfully supervised 80+ Ph.D. students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totaling in excess of 10 000 pages, published 1380+ research entries at the IEEE Xplore, acted both as a TPC and General Chair of the IEEE conferences, presented keynote lectures, and has been awarded a number of distinctions. Currently, he is directing a 100-strong academic research team, involved on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council U.K., the European IST Programme, and the Mobile Virtual Centre of Excellence, U.K. He is an enthusiastic supporter of industrial and academic liaison and he offers a range of industrial courses. He is a Governor of the IEEE VTS. From 2008 to 2012, he was the Editor-in-Chief of the IEEE PRESS and since 2009, he has been a Chaired Professor with Tsinghua University, Beijing. He is a fellow of the Royal Academy of Engineering, the Wolfson Fellow of the Royal Society as well as a fellow of the EURASIP and the IET. He has 18 000+ citations.

AUTHOR QUERIES

- AQ:1 = Please supply index terms/keywords for your paper. To download the IEEE Taxonomy, go to http://www.ieee.org/documents/taxonomy_v101.pdf.
- AQ:2 = Please confirm the volume no. for ref. [31].
- AQ:3 = Please provide the accessed date for refs. [40]–[41] and [43].

Motion-Aware Mesh-Structured Trellis for Correlation Modelling Aided Distributed Multi-View Video Coding

Yongkai Huo, Tao Wang, Robert G. Maunder, *Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

Abstract—A joint source-channel coding has attracted substantial attention with the aim of further exploiting the residual correlation residing in the encoded video signals for the sake of improving the reconstructed video quality. In our previous paper, a first-order Markov process model was utilized as an error concealment tool for exploiting the intra-frame correlation residing in the Wyner-Ziv (WZ) frame in the context of pixel-domain distributed video coding. In this contribution, we exploit the inter-view correlation with the aid of an inter-view motion search in distributed multi-view video coding (DMVC). Initially, we rely on the system architecture of WZ coding invoked for multi-view video. Then, we construct a novel mesh-structured pixel-correlation model from the inter-view motion vectors and derive its decoding rules for joint source-channel decoding. Finally, we benchmark the attainable system performance against the existing pixel-domain WZ coding based DMVC scheme, where the classic turbo codec is employed. Our simulation results show that substantial bitrate reductions are achieved by employing the proposed motion-aware mesh-structured correlation modelling technique in a DMVC scheme.

Index Terms—XXXXX.

I. INTRODUCTION

MULTI-VIEW Video Coding (MVC) [1] has recently attracted substantial attention in the context of both sophisticated 3D-TV and low-complexity wireless sensor network scenarios. A number of video coding techniques [2]–[4] have been developed for MVC, which typically rely on a high-complexity encoder and a low-complexity decoder. The constraint of the inter-view prediction based MVC is that all cameras of the MVC must exchange their monoscopic views with each other for inter-view prediction. What is even more challenging is that the communication between cameras must have a low-latency in delay-sensitive lip-synchronized interactive applications. These requirements are unrealistic in many applications, such as wireless video sensor networks

(WVSN) [5] for example, where both the energy and the computational complexity are constrained [6]. Hence it is necessary to shift the computationally complex tasks from the sensor to the base station (BS) or the server of the network, in addition to limiting the data exchange among the sensors. In theory, the Wyner-Ziv (WZ) [7] video coding philosophy, also known as distributed video coding (DVC) [8], is capable of assisting the sensors in relocating the computational burden to the BS, whilst simultaneously limiting the data exchange among themselves. Below, we will firstly review the WZ coding techniques for monoscopic video and for multi-view video, followed by the motivation of our proposed algorithm.

Two basic types of WZ coding structures [8] have been proposed for monoscopic video. In [9]–[11], the authors advocated a WZ codec, which is composed of an inner turbo-code-based [12] Slepian-Wolf (SW) codec [13] concatenated with an outer quantization-reconstruction component pair. More specifically, the odd-indexed video frames, namely the so-called key frames are intra-coded, while the even frames, namely the WZ frames are encoded by the WZ codec. At the receiver, the side information of the WZ frames will be estimated from their adjacent key frames for joint inter-frame decoding. However, a specific impediment of this structure is that the turbo decoder has to invoke a “request-and-decode” [9] process for the transmission of the WZ frames’ parity bits, which precludes its application in delay-sensitive services. Low density parity check (LDPC) codes were employed for distributed source coding in [14], which were the so-called rate-adaptive LDPC accumulate (LDPCA) codes and the sum-LDPC-accumulate (SLDPCA) codes. It was shown in [14] that the LDPCA codes are capable of approaching the capacity of a variety of communication channels more closely - including that of the virtual channel in DVC - than the family of turbo codes. Based on the WZ video coding structure of [11], in the European project DIStributed CODing for Video sERVICES (DISCOVER) [15] the transform-domain of the WZ frames was encoded by the LDPCA code of [14] at the transmitter. A so-called unsupervised motion learning technique was proposed in [16], which estimates the Motion Vectors (MVs) of the next video frame during the decoding of the current frame with reference to the previous reconstructed frame. This technique may be readily applied for both pixel-domain and transform-domain coding. The authors of [17] proposed a more realistic WZ video coding approach, which performs online estimation of the channel-induced noise (CN)

Manuscript received April 27, 2013; revised August 30, 2013 and October 23, 2013; accepted October 27, 2013. This work was supported in part by the EU’s Concerto Project of European Research Council’s Senior Fellow Grant and in part by the RC-U.K. under the India-U.K. Advanced Technology Centre. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stefano Tubaro.

The authors are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: yh3g09@ecs.soton.ac.uk; tw08r@ecs.soton.ac.uk; rm@ecs.soton.ac.uk; lh@ecs.soton.ac.uk).

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

Digital Object Identifier 10.1109/TIP.2013.2288913

model parameters at the decoder, which can be invoked for both pixel-domain and for transform domain WZ video codecs. Moreover, three levels of granularity were proposed by Brites and Pereira for pixel-domain WZ (PDWZ) video coding in [17], namely frame-, block- and pixel-level granularity, while both DCT-band and DCT-coefficient level granularity was proposed for transform-domain WZ (TDWZ) video coding. Then in [18] the same authors proposed an efficient encoder rate control (ERC) solution for feedback free transform-domain WZ (TDWZ) video coding. The approach of multiple side information (SI) components was proposed in [19] for improving the accuracy of SI using a single estimation mode. As a further advance, a more accurate parity rate estimator (PRE) is employed for more closely estimating the parity rate necessitated. A context-adaptive Markov random field reconstruction algorithm was proposed in [20], which exploits the spatio-temporal correlation by modelling the WZ frames. The so-called optical flow was proposed in [21] to improve the side information generation, which is exploited at the decoder side to compensate for weaknesses of block-based methods. A successive bit-plane-by-bit-plane refinement of the SI estimation algorithm was investigated in [22], leading to successively improved SI. The same authors proposed a motion-compensated multi-hypothesis prediction technique for medical imaging applications. Moreover, techniques conceived for multiple SI generation were proposed in [23], [24], where additional information may be used for improving the estimated SI of WZ frames. Apart from the WZ coding architecture mentioned above, another DVC architecture was proposed in [25], which allows the flexible sharing of complexity between the encoder and decoder.

Below, we review the WZ coding techniques designed for multi-view video sequences. In [26], the authors proposed a novel framework for the distributed compression of multi-view images, which was based on a tree-structured compression algorithm that guaranteed an optimal rate-distortion performance for specific video signals. Yeo and Ramchandran extended their previous PRISM framework detailed in [25] into distributed MVC in [27], [28], where the achievable error-resilience was studied in wireless scenarios. Two alternative models were proposed for exploiting the inter-view correlation, namely the view-synthesis-based correlation model and the disparity-based correlation model. The view-synthesis-based correlation model requires at least two other camera views and relies on both disparity estimation and view interpolation, while the disparity-based correlation model requires only a single additional camera view. In [29], the authors extended the WZ framework proposed in [11] into distributed MVC. At the encoder side, a wavelet-based WZ scheme was proposed for compressing each camera's view independently, where all coefficients were organized as proposed in the SPIHT scheme of [30] on a bitplane by bitplane basis. At the decoder side, a flexible prediction technique was proposed for generating the required SI, which jointly exploited both the temporal and inter-view correlations. The common benefit of the frameworks advocated in [28], [29] is that inter-camera communication is completely avoided and the computational complexity of the encoder was shifted to the decoder.

The iterative source-channel decoding (ISCD) [31] principle can be utilized for improving the system's performance by exploiting the residual correlation within the source signals. Moreover, a first-order Markov model based error concealment technique was developed in our previous work [32], where the intra-frame correlation was exploited for achieving an improved reconstructed video quality. However, the ISCD principle has not been conceived for family of the WZ multi-view video codecs. Since the inter-view correlation is not removed by the WZ multi-view video encoder, it is beneficial to exploit the residual correlation at the receiver for the sake of reducing the required bitrate. In this treatise, we develop a novel mesh-structured source model (MSSM) based decoder for exploiting the correlation among the inter-view pixels, which will be combined with a turbo codec for performing iterative source-channel decoding (ISCD) in the context of DMVC for the sake of achieving a reduced bitrate. Against this background, our novel contributions are:

- 1) We conceive a novel mesh-structured trellis exemplified in Fig. 7 for exploiting the inter-view correlations inherent in the video signal. Furthermore, the corresponding decoding rules of this trellis are derived for the sake of performing turbo-like iterative decoding, again, by exploiting the inter-view correlations.
- 2) We apply the new-trellis based proposed technique in a DMVC system, which results in a substantial bitrate reduction.

This rest of this paper is organized as follows. Section II briefly reviews the first-order Markov modelling technique used for exploiting the intra-frame correlation. We present our DMVC system architecture in Section III. In Section IV we detail our mesh-structured source model conceived for exploiting the correlation among the inter-view pixels, which is applied in the distributed MVC system of Section III. The performance of the proposed scheme is quantified with the aid of simulations in Section V. Finally, we offer our conclusions in Section VI.

II. FIRST-ORDER MARKOV PROCESS MODEL

The *a-posteriori* probability determination technique conceived for first-order Markov processes was detailed in [31]–[33]. In this section, we will briefly introduce the technique of first-order Markov process aided decoding. Let us commence by stipulating the following assumptions:

- x_i : an m -bit pattern of pixels scanned from the original video pixels at time instant i , which is expressed as $\{x_i(0), \dots, x_i(m-1)\} = x_i \binom{m-1}{0}$;
- m : the number of bits in each m -bit pattern x_i of pixels;
- $X_m = \{0, 1, \dots, 2^m - 1\}$: the set of all possible values in an m -bit pattern x_i ;
- $x_0^t = x_0, \dots, x_t$: the bit patterns of the 1^{st} frame of the original video consisting of $(t+1)$ m -bit patterns during the time interval spanning from 0 to t ;
- $y_0^t = y_0, \dots, y_t$: potentially error-infested bit pattern of the 1^{st} frame;

The corresponding trellis of the first-order Markov process is displayed in Fig. 1, where the m -bit pattern x_i indicates the

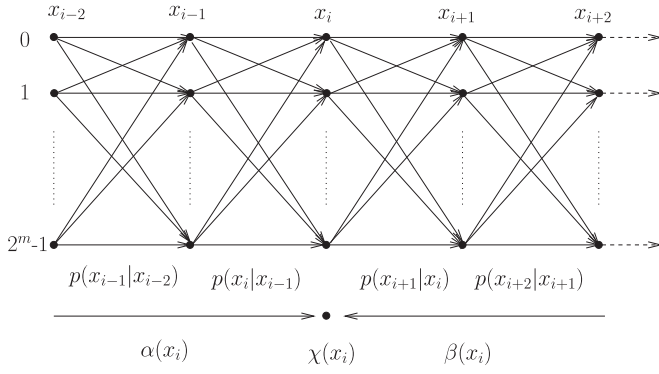


Fig. 1. Trellis of first-order Markov process for BCJR decoding, where $p(x_{i+1}|x_i)$ is the Markov transition probability.

trellis state at time instant i and the probability $p(x_{i+1}|x_i)$ indicates the transition from state x_i to state x_{i+1} . At the receiver, the *a-posteriori* probability of the m -bit pattern $x_i, x_i \in X_m$ conditioned on the specific received frame of m -bit patterns y_0, \dots, y_t may be expressed as

$$p(x_i|y_0^t) = \frac{p(x_i \wedge y_0^t)}{p(y_0^t)}, \quad (1)$$

where the joint probability $p(x_i \wedge y_0^t)$ of the m -bit pattern x_i and of the received frame y_0^t may be further formulated as [32]

$$p(x_i \wedge y_0^t) = \beta(x_i) \cdot \chi(x_i) \cdot \alpha(x_i). \quad (2)$$

It was shown in [32] that the bit-based *a-posteriori* LLR $L[x_i(k)|y_0^t]$ can be formulated as

$$L[x_i(k)|y_0^t] = \ln \frac{\sum_{\substack{x_i(k)=0 \\ x_i \in X_m}} \beta(x_i) \cdot \chi(x_i) \cdot \alpha(x_i)}{\sum_{\substack{x_i(k)=1 \\ x_i \in X_m}} \beta(x_i) \cdot \chi(x_i) \cdot \alpha(x_i)}, \quad (3)$$

where the components α, β, χ are derived in [32].

III. WYNER-ZIV CODING FOR MULTI-VIEW VIDEO: SYSTEM MODEL

Again, WZ compression techniques designed for monoscopic video have attracted substantial research attention [8]–[11], [14]–[18], [20], [25], [34], [35]. A number of contributions have also been proposed for DMVC [26]–[29]. In this section, we will detail the WZ compression philosophy shown in Fig. 2, which is invoked for distributed MVC employing our proposed MSSM-Turbo decoder. In the system of Fig. 2, there are N cameras capturing N views, respectively. We consider an array of N cameras, which employ N identical low-complexity video encoders for encoding the N camera-views independently at the transmitter and a potentially high-complexity video decoder for jointly decoding the N camera-views at the receiver. Each group of pictures (GOP) of each of the N camera views consists of a single I frame followed by a fixed number of WZ frames. The architecture of the investigated pixel-domain Wyner-Ziv

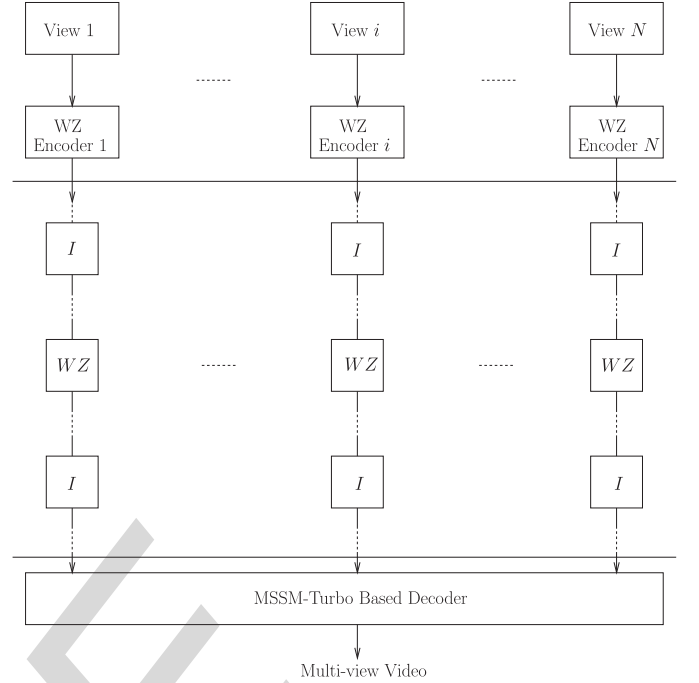


Fig. 2. The schematic compression process of Wyner-Ziv coding conceived for multi-view video.

coding system is displayed in Fig. 3, where the MSSM exploits the inter-view correlations, which cannot be removed by the multi-view Wyner-Ziv encoder, regardless, whether a pixel-domain architecture or a transform-domain architecture is employed. Hence the proposed MSSM is not limited to the family of pixel-domain architectures. The residual inter-view correlation encountered in transform-domain systems may potentially be exploited by appropriately designing the MSSM techniques, which may be part of our future research.

Generally, this treatise focuses on the MSSM-Turbo decoder of Fig. 3, while the rest of the techniques, including the WZ encoder, the motion-compensated frame interpolation (MCFI) etc., are detailed in [35]. Below, we will briefly introduce the system of Fig. 3, while the MSSM-Turbo decoder will be detailed in Section IV.

A. Transmitter

For each view, the frames are classified into two categories, namely the so-called key frames U_1, \dots, U_N of Fig. 3 and the WZ frames V_1, \dots, V_N of Fig. 3. The key frames U_1, \dots, U_N , also referred to as I frames, are intra-frame-coded by the H.264/AVC encoder and then transmitted to the receiver. Below we consider the monoscopic WZ frame V_i ($1 \leq i \leq N$) for introducing the encoding process of the WZ frames in Fig. 3. The encoding process is encapsulated in the following steps.

- The monoscopic frame V_i is quantized by a uniform quantizer at the transmitter of Fig. 3 generating the resultant m -bit monoscopic frame q_i .
- Each pixel of the m -bit quantized monoscopic frame q_i is decomposed into m bits. Then the bits from the same position of q_i constitute the most significant bit (MSB)

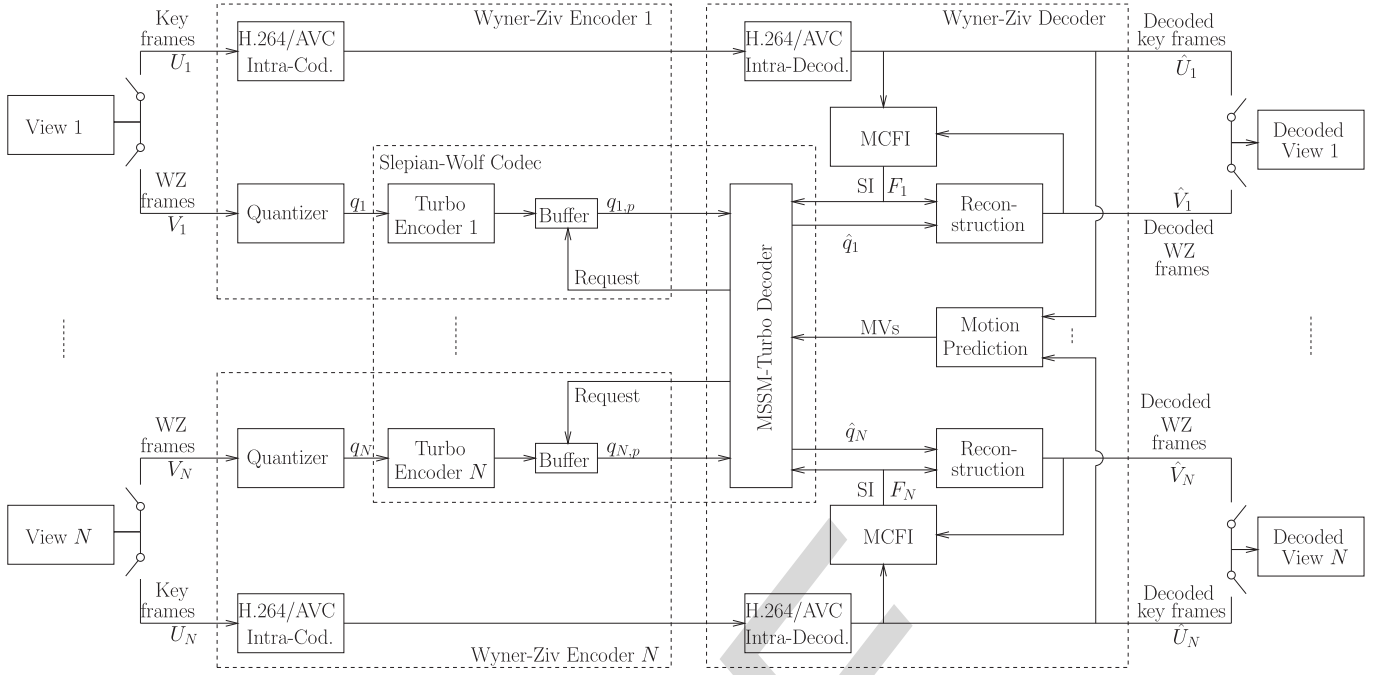


Fig. 3. System architecture of the Wyner-Ziv compression for MVC using MSSM-Turbo decoding.

plane resulting in m MSB planes. Afterwards, each of the m MSB planes will be considered as a block input, which is encoded by the turbo encoder i of Fig. 3.

- The systematic output of the turbo encoder i is directly transmitted to the receiver. The parity bits $q_{1,p}, \dots, q_{N,p}$ ¹ of Fig. 3 generated by the turbo encoder i for the different MSB planes of the WZ frame V_i may be independently buffered at the transmitter, which will then be transmitted to the receiver upon its request.

B. Receiver

The behavior of the receiver is described in Fig. 4, which will be further detailed in Section IV. At the receiver, the received bitstreams of the key frames representing the N camera views may be independently decoded for reconstructing the key frames $\hat{U}_1, \dots, \hat{U}_N$, which will then be utilized for estimating the SI for the corresponding WZ frames. Then the decoding process is listed as follows.

- For each monoscopic WZ frame, two temporally adjacent monoscopic key frames may be utilized for predicting the related soft-bit information. A number of algorithms have been proposed for this estimation process, such as for example the MCFI framework [35] shown in Fig. 3, which is invoked in our system for predicting the SI in the temporal direction.
- As observed from Fig. 4, the SI generated for the WZ frames and the received parity bits of the MSB planes related to the WZ frames will be used by the MSSM-Turbo decoder of Fig. 3 for reconstructing the WZ frames $\hat{V}_1, \dots, \hat{V}_N$.

- The flow-chart of Fig. 4 shows that when the MSSM-Turbo decoder fails to perfectly recover a MSB plane, the receiver will send a feedback flag to the transmitter for requesting more parity bits for this particular MSB plane. Again, this process is referred to as “request-and-decode”² process [9] in Fig. 4. The MSSM-Turbo decoder will be detailed in Section IV.
- As observed from Fig. 4, the “request-and-decode” process will terminate, when the bit error ratio (BER) of the MSB plane becomes lower than a preset threshold. Once all MSB planes have been decoded, the current multi-view frame may be readily reconstructed.

IV. INTER-VIEW CORRELATION MODELLING

The maximum *a-posteriori* probability (MAP) determination technique conceived for first-order Markov processes was briefly reviewed in Section II, where the intra-frame correlation was exploited using both horizontal and vertical Markov processes. However, the first-order Markov processes cannot be readily applied for modelling the inter-view correlation. In this section, we introduce the techniques invoked for exploiting the inter-view correlation of the N cameras by designing a novel trellis representation and derive its decoding rules. Below we focus on a specific multi-view WZ frame, which consists of N monoscopic video frames, whose SI is estimated by the MCFI of Fig. 3. Moreover, the trellis generation, MSSM-Turbo decoding and “request-and-decode” processes are illustrated in the flow-chart of Fig. 4. Let us commence by introducing the following notations:

- N : the number of camera views in the multi-view video;
- V_i : the original monoscopic frame of the camera view i ;

¹The subscript p indicates that $q_{1,p}, \dots, q_{N,p}$ are parity bits of quantized pixels q_1, \dots, q_N .

²Rate controller at the encoder side [18] may be investigated to avoid this feedback channel.

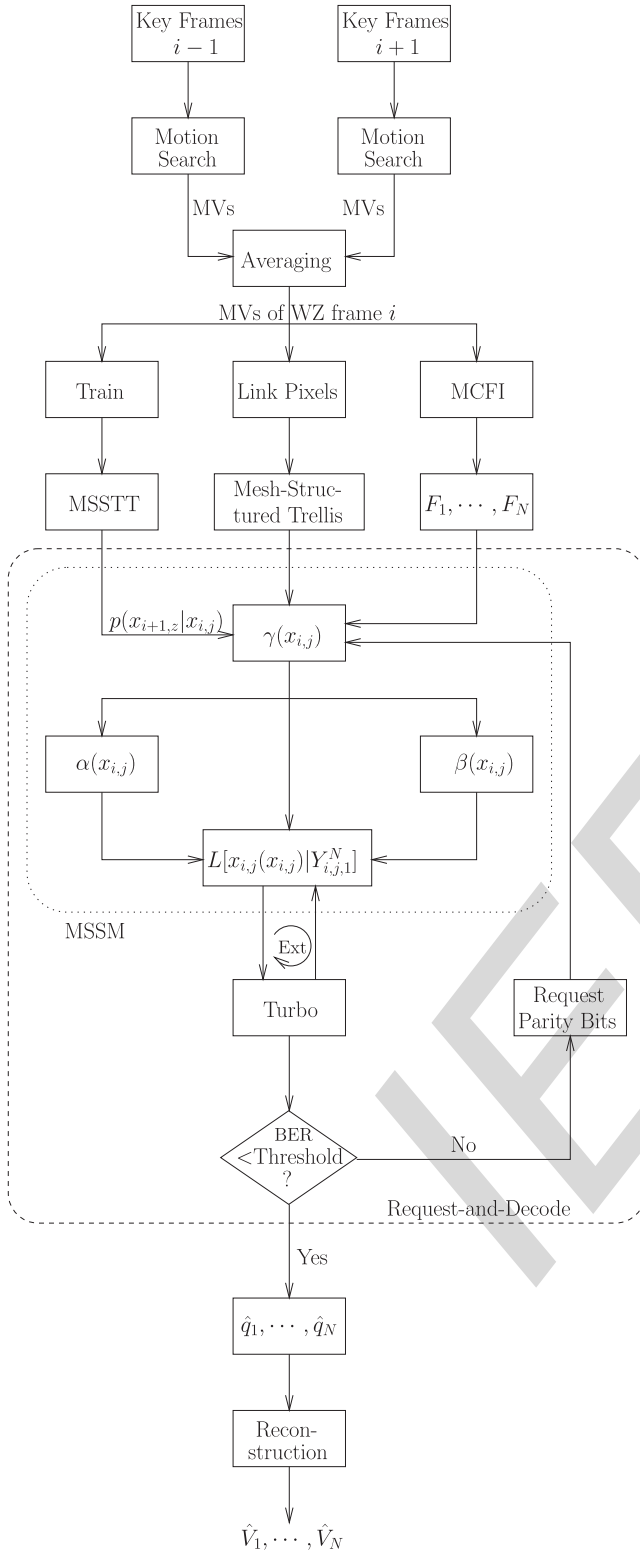
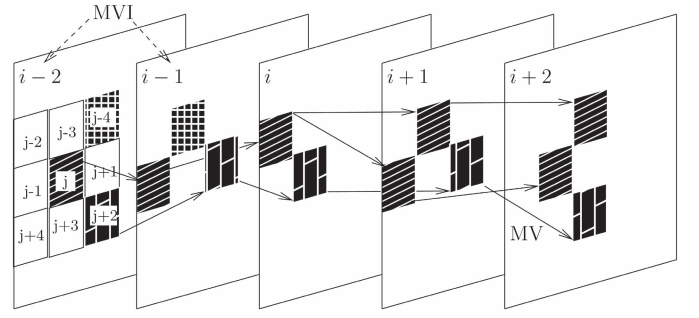


Fig. 4. Flow-chart of the receiver.

- F_i : the SI of the frame V_i , which is estimated by the MCFI block of Fig. 3;
- $x_{i,j}$: the j^{th} pixel³ in V_i , namely the pixel at position (i, j) ;

³The pixels of a 2D frame are indexed using a one-dimensional formulation, assuming that the 2D image is scanned into a single-dimensional vector.

Fig. 5. Motion based inter-view correlation, where $(j-4) \dots (j+4)$ indicate the indices of the (1×1) -pixel MBs.

- $x_{i,j}(k)$: the k^{th} bit of the pixel $x_{i,j}$.
- $y_{i,j}$: the SI of the pixel $x_{i,j}$, which is at position j of F_i ;
- m : the number of bits contained in the pixel $x_{i,j}$;
- $X_m = \{0, 1, \dots, 2^m - 1\}$: the set of legitimate values of an m -bit pattern $x_{i,j}$;
- $L_{i,j,k}$: the set of the original pixels linked with $x_{i,j}$ in camera view k .
- $L_{i,j,1}^N$: notation for the set $\bigcup_{t=1}^N L_{i,j,t}$;
- $Y_{i,j,k}$: the corresponding SI of $L_{i,j,k}$;
- $Y_{i,j,1}^N$: notation for the set $\bigcup_{t=1}^N Y_{i,j,t}$.

A. Mesh-Structured Trellis Representation

Again, each multi-view frame consists of N monoscopic camera view frames. Since the current multi-view WZ frame is not available at the receiver, the inter-view MVs of the current WZ frame are estimated from the adjacent key frames. Note that all the key frames are available at the receiver after the “H.264/AVC intra decoder” block of Fig. 3. Here we consider the scenario of $\text{GOP}=2$, but this technique may be readily extended to larger GOP scenarios. Specifically, the $(i-1)^{st}$ and $(i+1)^{st}$ key frames are utilized for estimating the MVs of the i^{th} WZ frame, as illustrated in the flow-chart of Fig. 4. The inter-view MVs of the WZ multi-view frames are generated as follows.

- The inter-view MVs of the $(i-1)^{st}$ and $(i+1)^{st}$ key frames are readily estimated using the traditional macroblock (MB) based motion search techniques [36], which is indicated by the “Motion Search” blocks of Fig. 4.
- The inter-view MVs of the i^{th} WZ frame is averaged based on that of the $(i-1)^{st}$ and $(i+1)^{st}$ key frames, as illustrated by the “Averaging” block of Fig. 4.

The structure of the inter-view “Motion Prediction” of Fig. 3 is shown in Fig. 5, where the MBs of the specific view associated with the Monoscopic View Index (MVI) i ($2 < i \leq I$) are estimated from the views $(i-1)$ and $(i+1)$. Note that in Fig. 5 we employ (1×1) -pixel “MBs” for the sake of simplifying the relevant descriptions, which the classic (8×8) -pixel MBs may be readily extended to.

Following the inter-view motion estimation process at the receiver, the pixels of a given MB of Fig. 5 may be linked to other pixels in the same position of the predicted MBs in the

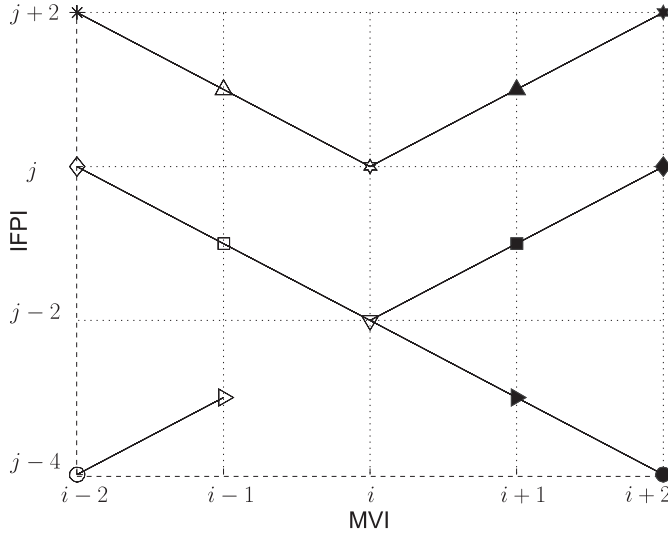


Fig. 6. The inter-view pixel correlation graph of Fig. 5, where each marker indicates a different pixel.

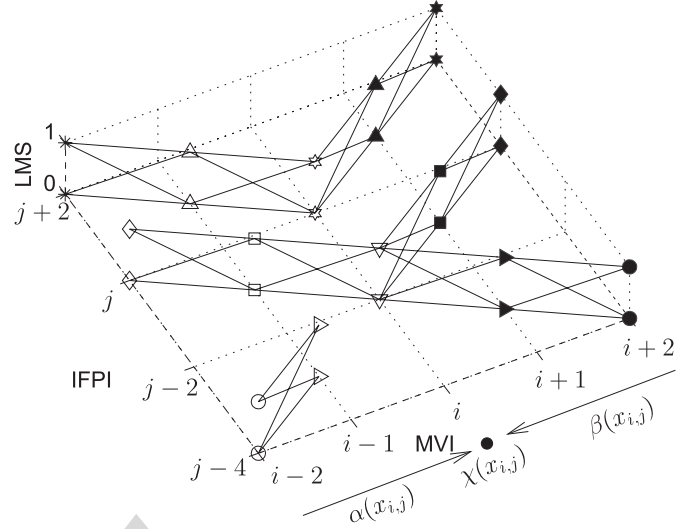


Fig. 7. Mesh-structured trellis representation of inter-view correlations for $m = 1$ -bit pixels, where the pixels of view $(i-2)$ to view $(i+2)$ are displayed.

other views, as exemplified in the inter-view pixel correlation graph of Fig. 6. Specifically, the graph of Fig. 6, indicated by the “Link Pixels” block of Fig. 4, may be created using the following steps:

- The pixels of Fig. 5 are scanned into one-dimensional Intra-Frame Pixel Indices (IFPI), namely $(j-4), \dots, (j+4)$ in view $(i-2)$ of Fig. 5. For example, each monoscopic frame having (352×288) -pixels will be indexed by $352 \cdot 288 = 72864$ one-dimensional scan-line indices.
- The coordinate axes of Fig. 6 are created by arranging for the x axis and y axis to indicate the MVI and the IFPI, respectively. Specifically, the MVI of Fig. 5 are in the range of $(i-2, \dots, i+2)$. Furthermore, we only consider the colored pixels of Fig. 5 for simplifying the related descriptions, resulting in IFPI of $(j-4, \dots, j+2)$.
- Connect all the correlated pixels portrayed in Fig. 6 using a link, where the presence of highest correlation is indicated by the MVs of Fig. 5. For example, in Fig. 6 the pixel at position (i, j) is correlated with the pixels at positions of $(i-1, j+1)$ and $(i+1, j+1)$ in Fig. 6, as indicated by the MVs of Fig. 5.

Based on the two-dimensional Markov-modelling based trellis representation developed in [32], the mesh-structured trellis of Fig. 7 may be derived from the correlation graph of Fig. 6. This trellis generation process is also indicated in Fig. 4, which is completed using the following steps:

- Each m -bit pixel has a value in the range of $[0, 2^m)$, which we refer to as the Legitimate Markov States (LMS). Then each pixel in Fig. 6 has 2^m LMS. Hence by introducing the z axis indicating the LMS, Fig. 7 may be created, where the x axis and y axis indicate the MVI and IFPI, respectively.
- Each pair of pixels connected by a direct link in Fig. 6 represents a correlated Markov-state transition. Hence we have to incorporate $(2^m \times 2^m)$ links indicating the

$(2^m \times 2^m)$ possible Markov-state transitions for these pixels in Fig. 6. Specifically, for the pixel $x_{i,j}$ within view i , the transition probability from state $x_{i,j}$ to the correlated successor state $x_{i+1,j+1}$ within view $(i+1)$ is represented by $p(x_{i+1,j+1}|x_{i,j})$, which is the state transition of the related Markov process.

Hence all nodes in Fig. 7 associated with identical MVI belong to the same view, while the nodes having both an identical MVI and SII values are Markov states for a same corresponding pixel. Furthermore, the trellis representation seen in Fig. 7 may be readily generalized both for arbitrary m -bit pixel multi-view signals and for an arbitrary number of views, which leads to 2^m LMS.

B. Trellis Decoding

Section III detailed how the SI of a multi-view WZ frame can be generated. The SI consists of floating-point values indicating the reliability of specific pixel values, which is estimated by the MCFI block of Fig. 3. For example, the SI of the bit $x_{i,j}(k)$ may be expressed in the log-likelihood ratio (LLR) format as $L[x_{i,j}(k)] = \ln \frac{L[x_{i,j}(k)=0]}{L[x_{i,j}(k)=1]}$, while the reliability of the pixel $x_{i,j}$ can be represented by $L[x_{i,j}(0)], \dots, L[x_{i,j}(m-1)]$. In Section IV-A, the pixel correlations of a multi-view WZ frame are modelled by the mesh-structured trellis of Fig. 7. Hence the SI of the multi-view WZ frame may be refined by decoding the trellis of Fig. 7, where the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [37] decoding principle may be applied.

Given the SI of the current multi-view WZ frame F_1, \dots, F_N , the pixel $x_{i,j}$ may be estimated by the *a-posteriori* probability (APP) $p(x_{i,j}|F_1, \dots, F_N)$. Furthermore, by employing our proposed MSSM detailed in Section IV-A, we have

$$p(x_{i,j}|F_1, \dots, F_N) \approx p(x_{i,j}|Y_{i,j,1}^N), \quad (4)$$

where $Y_{i,j,1}^N$ includes the SI of the pixels $L_{i,j,1}^N$, which are correlated with $x_{i,j}$. Consider the trellis of Fig. 7 for example, where we have $L_{i,j,i-2} = \{x_{i-2,j+2}\}$, $L_{i,j,i-1} = \{x_{i-1,j+1}\}$, $L_{i,j,i} = \{x_{i,j}\}$, $L_{i,j,i+1} = \{x_{i+1,j+1}\}$ and $L_{i,j,i+2} = \{x_{i+2,j+2}\}$, the pixels $L_{i,j,i-2}^{i+2}$ can be readily derived as $L_{i,j,i-2}^{i+2} = \bigcup_{t=i-2}^{i+2} L_{i,j,t}$. In the Appendix, we show that the log-likelihood ratio (LLR) version of the APP $p(x_{i,j}(k)|Y_{i,j,1}^N)$ for the bit $x_{i,j}(k)$, $0 \leq k < m$ may be formulated as

$$L[x_{i,j}(k)|Y_{i,j,1}^N] = \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}, \quad (5)$$

where $\beta(x_{i,j})$, $\chi(x_{i,j})$, $\alpha(x_{i,j})$ are defined in Eqs. (16), (10) (15). Specifically, $\beta(x_{i,j})$, $\alpha(x_{i,j})$ indicate the backward and forward oriented probability of the pixel $x_{i,j}$, while $\chi(x_{i,j})$ is the channel information of the pixel $x_{i,j}$.

C. Iterative MSSM-Turbo Decoding

A limitation of the formulas provided in Section IV-B is that they cannot be directly used for iterative decoding, since they cannot exploit the *a-priori* LLR $L[x_{i,j}(k)]$, which was generated from the *extrinsic* information gleaned from the other decoder components. To make use of the *a-priori* LLR $L[x_{i,j}(k)]$, the combined bit-wise LLR may be expressed as [31], [32]

$$\gamma_{i,j}(x_{i,j}) = \exp \sum_{k=0}^{m-1} \frac{\bar{x}_{i,j}(k)}{2} \cdot \{L[x_{i,j}(k)] + L[y_{i,j}(k)|x_{i,j}(k)]\}, \quad (6)$$

where the symbol-based m -bit information γ is the combination of the bit-wise *a-priori* LLR $L[x_{i,j}(k)]$ and of the channel information $L[y_{i,j}(k)|x_{i,j}(k)]$. We note in this context that γ of Eq. (6) contains more valuable information than the channel information χ . Hence Eq. (17) may be used for iterative joint source-channel decoding by replacing χ with γ in Eq. (6).

Similar to the BCJR decoding technique of classic turbo codes [38], the bit-based *a-posteriori* LLR $L[x_{i,j}(k)|Y_{i,j,1}^N]$ may be split into three components, namely the *a-priori* LLR $L[x_{i,j}(k)]$, the channel information $L[y_{i,j}(k)|x_{i,j}(k)]$ and the extrinsic information $L_e[x_{i,j}(k)]$. Specifically, the bit-based *a-posteriori* LLR $L[x_{i,j}(k)|Y_{i,j,1}^N]$ may be formulated as

$$L[x_{i,j}(k)|Y_{i,j,1}^N] = L[x_{i,j}(k)] + L[y_{i,j}(k)|x_{i,j}(k)] + \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \gamma^{[ext]}[x_{i,j}(k)] \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \gamma^{[ext]}[x_{i,j}(k)] \cdot \alpha(x_{i,j})}, \quad (7)$$

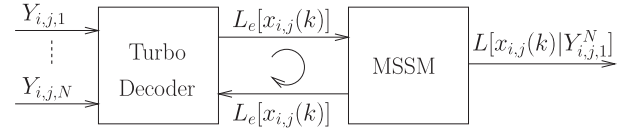


Fig. 8. The MSSM-Turbo decoding architecture of Fig. 3.

where the extrinsic information component $\gamma^{[ext]}[x_{i,j}(k)]$ may be expressed as

$$\gamma^{[ext]}[x_{i,j}(k)] = \exp \sum_{l=0, l \neq k}^{m-1} \frac{\bar{x}_{i,j}(l)}{2} \cdot \{L[x_{i,j}(l)] + L[y_{i,j}(l)|x_{i,j}(l)]\}.$$

Based on the above, the MSSM-Turbo decoding architecture of Fig. 3 is shown in Fig. 8, which terminates after I_{iter} iterations of extrinsic information exchange. Moreover, the MSSM-turbo decoding process is also detailed in the flow-chart of Fig. 4.

D. Training for Mesh-Structured State Transition

Again, the mesh-structured trellis of Fig. 7 is utilized for modelling the inter-view correlation at the receiver. Specifically, the component $p(x_{i+1,z}|x_{i,j})$ of Eqs. (15), (16) quantifies the inter-view correlation, where z and j indicate the pixel index of the views $(i-1)$ and i , respectively. Here we refer to $p(x_{i+1,z}|x_{i,j})$ as the Mesh-Structured State Transition Table (MSSTT) for simplicity. We initialize the $(2^m \times 2^m)$ -element MSSTT $T[0:2^m-1, 0:2^m-1]$ to zero values. Then we scan all the linked pixel pairs of Fig. 6. For example, when the pixel pairs s_{i-1} and s_i are scanned, the corresponding element $T[s_{i-1}, s_i]$ in the MSSTT is increased by 1. Finally, by normalizing the summation of all rows in the MSSTT $T[0:2^m-1, 0:2^m-1]$, the transition probabilities $p(x_{i+1,z}|x_{i,j})$ can be obtained, where we have $x_{i,j}, x_{i+1,z} \in [0, 2^m)$.

E. Complexity Analysis

The complexity of our proposed MSSM can be attributed to the calculation of $\gamma(x_{i,j})$ in Eq. (6), $\alpha(x_{i,j})$ in Eq. (15), $\beta(x_{i,j})$ in Eq. (16) and $L[x_{i,j}(k)|Y_{i,j,1}^N]$ in Eq. (5). As shown in Fig. 7, the trellis size is $(2^m \cdot S \cdot N)$, where S is the number of pixels in each monoscopic view. Similar to the BCJR decoding rules proposed in [37], the decoding of the $(2^m \cdot S \cdot N)$ -state trellis of Fig. 7 may be generalized into the following two stages:

- Calculation of γ , α and β : These operations are carried out across the entire trellis of Fig. 7, which imposes the complexity of m , 2^m , 2^m for each trellis state, as suggested by Eqs. (6), (15), (16), respectively. Hence the associated computational costs are on the order of $O(2^m \cdot S \cdot N \cdot m)$, $O(2^{2m} \cdot S \cdot N)$ and $O(2^{2m} \cdot S \cdot N)$ for γ , α and β , respectively.
- Calculation of $L[x_{i,j}(k)|Y_{i,j,1}^N]$: This operation is carried out for all the $(S \cdot N \cdot m)$ bits of a multi-view frame, which imposes a complexity of 2^m for each bit. Hence the computational cost is on the order of $O(2^m \cdot S \cdot N \cdot m)$.

TABLE I
TABLE OF PARAMETERS EMPLOYED FOR THE SIMULATIONS

Representation	YUV 4:2:0	Motion MB	8×8
Format	CIF	Generator of RSC	[11011,10011]
Bits Per Pixel	8	Turbo Code Rate	1/2
Number of Frames	100	Interleaver Length	352×288
GOP	2, 4	Puncturer	[1 1; 0 1; 1 0]
Number of Views	8	Intra-codec	H.264

Therefore, the overall complexity imposed by our proposed decoder is $O(2 \cdot 2^{2m} \cdot S \cdot N + 2 \cdot 2^m \cdot S \cdot N \cdot m)$, when decoding a multi-view frame.

V. PERFORMANCE STUDY

In this section, we present our simulation results for benchmarking the scheme introduced in Section III. Firstly, in Section V-A we will introduce the parameters of the scenario considered in our experiment. Then we will discuss our numerical results in Section V-B.

A. Scenario

In this section, we present our experimental parameters used for characterizing the convergence behavior of the proposed scheme introduced in Section III. Multi-view video sequences having 8 camera views represented in (352×288) -common intermediate format (CIF)⁴ and 4:2:0 YUV representation are employed. Moreover, the bitrate/PSNR of both the WZ and the key frames was taken into account in our average results. The distributed WZ coding scheme conceived for the multi-view video scheme of Fig. 3 operates on the basis of (352×288) -pixel blocks. More specifically, a specific MSB plane of each view is input to the turbo encoder, which consists of (352×288) bits. In other words, the interleaver length of our turbo codec is (352×288) bits. In [35], each bitplane of the MSB was transmitted separately and each bitplane was then refined based on the previously decoded bitplanes [39]. However, in our system, all MSB planes were transmitted together, which allowed us to reduce the number of “request-and-decode” processes defined in [8]. We employ a recursive systematic convolutional (RSC) encoder relying on the generator polynomials of $g_1 = 11011$, $g_2 = 10011$, which are represented as $G = [1, g_2/g_1]$, where g_1 is a feedback input and g_2 is feed-forward output. Moreover, two identical RSC encoders are employed for the turbo codec and the puncturing matrix of [1 1; 0 1; 1 0] is employed for the turbo code. The parameters employed are listed in Table I. The remaining parameters of our system were identical to those in [17], [35].

⁴We converted the multi-view video sequences into CIF representation for the sake of speeding up our simulations.

TABLE II
COMPARISON OF MSSM-TURBO AND THE BENCHMARKER TURBO FOR WYNER-ZIV CODING OF MULTIVIEW VIDEO, WHERE n_{MSB} INDICATES THE NUMBER OF MSB PLANES. $I_{\text{iter}} = 4$ IS EMPLOYED, WHERE INDICATES THE NUMBER OF DECODING ITERATIONS FOR THE MSSM-TURBO DECODER

	Turbo	MSSM-Turbo
Dimension	1	2
Number of Bits to Decode	352×288	$352 \times 288 \times 8 \times n_{\text{MSB}}$
Side Information	None	$1 \times \text{MMSTT}$

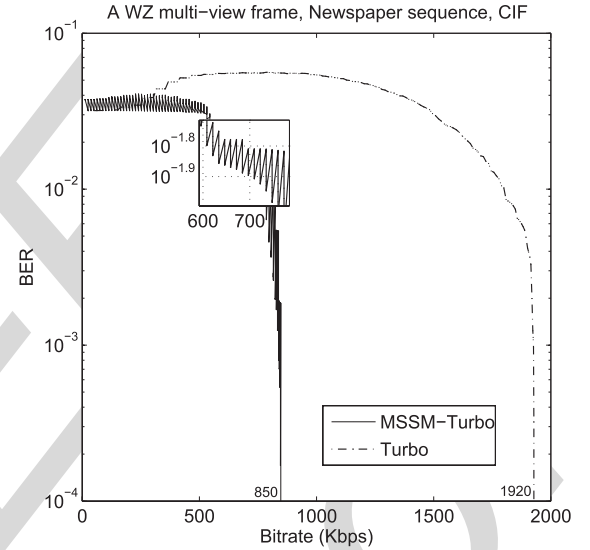


Fig. 9. BER versus bitrate comparison of the MSSM-turbo and of turbo codec for a multi-view WZ frame, where only the bits of the WZ frame are taken into account as the bitrate. Using the *Newspaper* sequence and $\text{GOP} = 2$.

TABLE III
BJNTEGAARD COMPARISON OF THE MSSM-PDWZ VERSUS THE PDWZ SCHEMES FOR THE CONSIDERED MULTI-VIEW SEQUENCES ORGANIZED IN GOPS OF 2 AND 4. “R” INDICATES “RATE”

Sequence	GOP=2		GOP=4	
	ΔR (%)	ΔPSNR (dB)	ΔR (%)	ΔPSNR (dB)
Newspaper	0.3	0.02	1.8	0.17
Leavinglaptop	6.7	0.42	15.6	0.92
Outdoor	12.6	0.97	30.7	2.27
Ballroom	13.4	0.80	27.2	1.58

Let us now compare the proposed MSSM-Turbo decoder to the classic turbo decoder. The turbo decoder is invoked for each MSB plane of each monoscopic frame, which carries (352×288) bits, scanned into a one-dimensional vector.

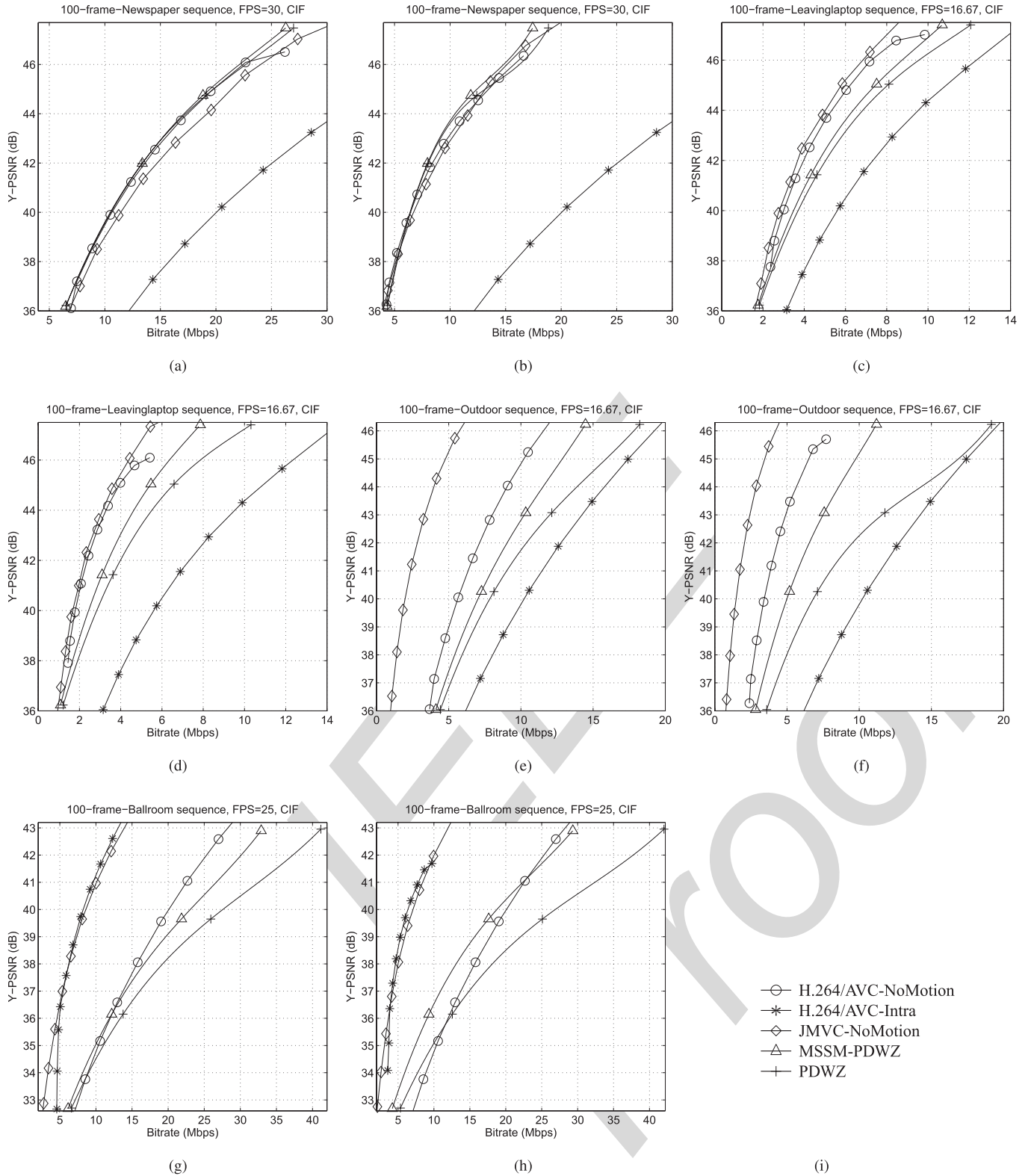


Fig. 10. Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. (a) *Newspaper*, GOP = 2. (b) *Newspaper*, GOP = 4. (c) *Leaving-laptop*, GOP = 2 (d) *Leaving-laptop*, GOP = 4. (e) *Outdoor*, GOP = 2. (f) *Outdoor*, GOP = 4. (g) *Ballroom*, GOP = 2. (h) *Ballroom*, GOP = 4. (i) Benchmarks.

Let n_{MSB} be the number of MSB planes. The MSSM-Turbo decoder is invoked for all MSB planes of each multi-view frame, which carries $(352 \times 288 \times 8 \times n_{\text{MSB}})$ bits and it is

arranged into $I = 8$ scanlines or vectors. Furthermore, the MSSM-Turbo scheme relies on the corresponding MSSTT, which has $(2^{n_{\text{MSB}}} \times 2^{n_{\text{MSB}}})$ elements. Specifically, for the



Fig. 11. Comparison of original WZ frames and the related estimated SI frames at the receiver. From left to right are *Newspaper*, *Leaving-laptop*, *Outdoor* and *Ballroom* sequences, respectively. The top row and the bottom rows list the original and the estimated SI frames, respectively.

simulations associated with $m = 4$, the size of the MSSTT SI was (16×16) floating-point values for 100 CIF frames. Our detailed comparisons are listed in Table II. Note that the MSSTT may be estimated from the adjacent I frames of the current WZ frame.

B. Numerical Results

In this section, we present our numerical results for benchmarking the proposed MSSM-PDWZ scheme's performance against that of the PDWZ system of [17] for four multi-view sequences associated with different motion features, namely the *Newspaper* sequence, the *Leaving-laptop* sequence [40], the *Outdoor* sequence [40] and the *Ballroom* sequence [41], scanned in FPSs of 30, 16.67, 16.67, 25, respectively. The rate-distortion (RD) results recorded for both the PDWZ and MSSM-PDWZ schemes in Fig. 3 were parameterized by the number of WZ coded bitplanes for $m = 1, 2, 3$ or 4 MSB planes, because this configuration has been widely adopted in the pixel-domain WZ video coding literature [17]. This was arranged by invoking the uniform quantizers shown in Fig. 3. The RD results of the test sequences coded by the H.264/AVC codec [42] are provided below as usual, both in the associated intra-frame encoding mode and in the motion compensation dispensed mode in conjunction with GOP periods of 2 and 4. Both these modes were selected by appropriately adjusting the encoding parameters of the H.264/AVC reference software JM [43]. Furthermore, the performance of the multi-view video codec JMVC operating without motion estimation and using GOP periods of 2 and 4 was also provided.

1) *BER Characteristics*: The BER comparison of the MSSM-turbo and turbo codec schemes is displayed in Fig. 9, where the x axis represents the bitrate of the multi-view WZ

frame. Note that the bitrate of a video sequence consists of the bitrate of the WZ frames and the bitrate of the key frames. Here we only count the bitrate of the WZ frames for the sake of providing further insights into our system's behavior.

For the MSSM-turbo decoder, the BERs of both the MSSM and of the turbo decoder components are provided. More specifically, multiple BER values are plotted for the MSSM-turbo decoder for each "Bitrate" abscissa value in Fig. 9. This results in a wave-shaped, fluctuating curve for the MSSM-turbo decoder. Alternatively, for each "Bitrate" value, the MSSM-turbo is capable of further reducing the BER upon the turbo codec during the ISCD process. Observe from Fig. 9 that BER of the WZ frame using the MSSM-turbo decoder becomes vanishingly low at about 850 Kbps, while that of the turbo decoder vanishes at 1920 Kbps. The MSSM-turbo decoder requires 1070 less Kbps for achieving a BER of 10^{-4} . This observation is due to the fact that MSSM is capable of further reducing the BER by exploiting the residual redundancy within the WZ frame by iteratively exchanging extrinsic information with the turbo decoder.

2) *Rate-Distortion Characteristics*: The simulation results recorded for the four sequences are displayed in Fig. 10, while the Bjntegaard comparison of the MSSM-PDWZ versus the PDWZ schemes for the four multi-view sequences considered and organized in GOPs of 2 and 4 is provided in Table III. For the MSSM-PDWZ, Fig. Fig. 10 shows that the GOP = 4 scenarios outperforms the GOP = 2 regime for the Newspaper, Leavinglaptop and Outdoor sequences, while the opposite trends were observed for the Ballroom sequence. This is due to the fact that less bits are required for reconstructing the WZ frames than that of the key frames, when the SI for the WZ frames can be accurately estimated. Table III shows

that the bitrate reduction ratio increases in the sequence order of Newspaper, Leavinglaptop, Outdoor and Ballroom. Similar trends are observed in terms of the PSNR reduction attained. The reason for this trend is that the receiver is more unlikely to be able to estimate the SI frame accurately from the received key frames, while our MSSM-PDWZ scheme is capable of effectively concealing the errors, which is an explicit benefit of our MSSM-Turbo decoder.

The comparison of the original frames to the corresponding estimated SI frames is displayed in Fig. 11 for the *Newspaper*, *Leavinglaptop*, *Outdoor* and *Ballroom* sequences, respectively. Observe that the receiver fails to estimate the SI frames of the higher-motion sequences, namely of the *Outdoor* and the *Ballroom* sequences, as precisely as for the lower-motion sequences, namely for the *Newspaper* and the *Leaving-laptop* sequences. We may conclude that our proposed MSSM technique is capable of reducing the bitrate more substantially for the higher-motion sequences, where the SI of key frames cannot be accurately estimated at the receiver.

VI. CONCLUSION

In this paper, we firstly extended the WZ coding techniques for monoscopic video into a Wyner-Ziv coded multi-view video system. Then we conceived the techniques for constructing a novel mesh-structured pixel correlation model from the inter-view MVs and derived its decoding rules. Furthermore, by incorporating the MSSM scheme into WZ video coding of multi-view video, we were able to substantially reduce the bitrate compared to that of the PDWZ benchmarker systems.

Our future work will focus on developing techniques for exploiting both the pixel correlation among different views and the pixel correlation within camera views for the sake of further reducing the required bitrate.

APPENDIX

Let us initially follow the procedure of the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [37] algorithm based determination rule of the MAP decoder for deriving the APP $p(x_{i,j}|Y_{i,j,1}^N)$ of Eq. (4). The APP $p(x_{i,j}|Y_{i,j,1}^N)$ of the m -bit pattern $x_{i,j}, x_{i,j} \in X_m$ conditioned on the specific $Y_{i,j,1}^N$ values may be expressed as

$$p(x_{i,j}|Y_{i,j,1}^N) = \frac{p(x_{i,j} \wedge Y_{i,j,1}^N)}{p(Y_{i,j,1}^N)}. \quad (8)$$

The joint probability $p(x_{i,j} \wedge Y_{i,j,1}^N)$ of the m -bit pattern $x_{i,j}$ in Eq. (8) and that of SI $Y_{i,j,1}^N$ may be further formulated as

$$\begin{aligned} & p(x_{i,j} \wedge Y_{i,j,1}^N) \\ &= p(x_{i,j} \wedge Y_{i,j,i-1}^{i-1} \wedge Y_{i,j,i+1}^N) \\ &= p(Y_{i,j,i+1}^N | x_{i,j}) \cdot p(Y_{i,j,i} | x_{i,j}) \cdot p(Y_{i,j,i-1}^{i-1} \wedge x_{i,j}) \\ &= p(Y_{i,j,i+1}^N | x_{i,j}) \cdot p(y_{i,j} | x_{i,j}) \cdot p(Y_{i,j,i-1}^{i-1} \wedge x_{i,j}) \\ &= \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j}). \end{aligned} \quad (9)$$

In Eq. (9), the symbol-based channel information $\chi(x_{i,j}) = p(y_{i,j}|x_{i,j})$ may be calculated from the bit-based channel information as

$$\chi(x_{i,j}) = C_{\chi_{i,j}} \cdot \exp \sum_{k=0}^{m-1} \frac{x_{i,j}(k)}{2} \cdot L[y_{i,j}(k)|x_{i,j}(k)], \quad (10)$$

where $C_{\chi_{i,j}}$ is a normalization factor, which solely depends on $y_{i,j}$. Furthermore, similar to the forward recursion calculation in the BCJR algorithm, the component $\alpha(x_{i,j})$ in Eq. (9) may be formulated as

$$\begin{aligned} & \alpha(x_{i,j}) \\ &= p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} p(Y_{i,j,i-1} \wedge Y_{i,j,1}^{i-2} \wedge x_{i,j} \wedge L_{i,j,i-1}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} p(Y_{i,j,i-1} \wedge x_{i,j} | L_{i,j,i-1}) \cdot p(Y_{i,j,1}^{i-2} \wedge L_{i,j,i-1}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} p(Y_{i,j,i-1} | L_{i,j,i-1}) \cdot p(x_{i,j} | L_{i,j,i-1}) \\ &\quad \cdot p(Y_{i,j,1}^{i-2} \wedge L_{i,j,i-1}). \end{aligned} \quad (11)$$

Note that, given the original pixel set $L_{i,j,i-1}$ the soft pixels within $Y_{i,j,i-1}$ are independent to each other, hence the item $p(Y_{i,j,i-1} | L_{i,j,i-1})$ in Eq. (11) may be expressed as

$$p(Y_{i,j,i-1} | L_{i,j,i-1}) = \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} \chi(x_{i-1,z}). \quad (12)$$

Then the forward recursion calculation may be further formulated as

$$\begin{aligned} & \alpha(x_{i,j}) = \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} [p(y_{i-1,z} | x_{i-1,z}) \\ &\quad \cdot p(Y_{i,j,i-1}^{i-2} \wedge x_{i-1,z})] \cdot p(x_{i,j} | L_{i,j,i-1}) \\ &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{|L_{i,j,i-1}|}}} \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} [\chi(x_{i-1,z}) \cdot \alpha(x_{i-1,z})] \cdot p(x_{i,j} | L_{i,j,i-1}). \end{aligned} \quad (13)$$

Furthermore, by assuming the pixels are independent to each other the item $p(x_{i,j} | L_{i,j,i-1})$ in Eq. (13) may be approximated as follows

$$\begin{aligned} & p(x_{i,j} | L_{i,j,i-1}) = \frac{p(L_{i,j,i-1} | x_{i,j}) \cdot p(x_{i,j})}{p(L_{i,j,i-1})} \\ &= \frac{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i-1,z} | x_{i,j}) \cdot p(x_{i,j})}{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i-1,z})} = \frac{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i,j} | x_{i-1,z})}{p(x_{i,j})^{|L_{i,j,i-1}|-1}}. \end{aligned} \quad (14)$$

Then by substituting Eq. (14) into the Eq. (13), $\alpha(x_{i,j})$ may be approximated as

$$\alpha(x_{i,j}) \approx \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} \left\{ \sum_{\substack{x_{i-1,z} \in \\ X_m}} \chi(x_{i-1,z}) \cdot \alpha(x_{i-1,z}) \cdot p(x_{i,j}|x_{i-1,z}) \right\} / p(x_{i,j})^{L_{i,j,i-1}-1}. \quad (15)$$

Similar to $\alpha(x_{i,j})$, the backward recursion calculation $\beta(x_{i,j})$ in Eq. (9) can be formulated as

$$\begin{aligned} \beta(x_{i,j}) &= p(Y_{i,j,i+1}^N \wedge x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m}} p(Y_{i,j,i+2}^N \wedge Y_{i,j,i+1} \wedge L_{i,j,i+1} | x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m}} p(Y_{i,j,i+2}^N | L_{i,j,i+1}) \cdot p(Y_{i,j,i+1} | L_{i,j,i+1}) \\ &\quad \cdot p(L_{i,j,i+1} | x_{i,j}) \\ &= \prod_{\substack{x_{i+1,z} \in \\ L_{i,j,i+1}}} \left\{ \sum_{\substack{x_{i+1,z} \in \\ X_m}} [\beta(x_{i+1,z}) \cdot \chi(x_{i+1,z}) \cdot p(x_{i+1,z} | x_{i,j})] \right\}. \end{aligned} \quad (16)$$

Finally, the determination of the bit-based APP LLRs $L[x_{i,j}(k) | Y_{i,j,1}^N]$ may be formulated as

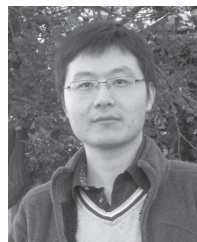
$$\begin{aligned} L[x_{i,j}(k) | Y_{i,j,1}^N] &= \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} p(x_{i,j} | Y_{i,j,1}^N)}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} p(x_{i,j} | Y_{i,j,1}^N)} \\ &= \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}, \end{aligned} \quad (17)$$

where the Jacobian logarithm [38] can be readily applied for deriving the log-domain representation of our algorithm.

REFERENCES

- [1] M. Flierl and B. Girod, "Multiview video compression," *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 66–76, Nov. 2007.
- [2] Y. Chen, M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj, and H. Li, "Coding techniques in multiview video coding and joint multiview video model," in *Proc. PCS*, May 2009, pp. 1–4.
- [3] J. Zhang, M. Hannuksela, and H. Li, "Joint multiview video plus depth coding," in *Proc. 17th IEEE ICIP*, Sep. 2010, pp. 2865–2868.
- [4] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.
- [5] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.
- [6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM Int. Workshop WSN*, New York, NY, USA, Sep. 2002, pp. 88–97.
- [7] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.
- [8] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.
- [9] A. Aaron, R. Zhang, and B. Girod, "Wyner–Ziv coding of motion video," in *Proc. Conf. Rec. 36th Asilomar Conf. Signals, Syst. Comput.*, vol. 1, Nov. 2002, pp. 240–244.
- [10] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner–Ziv coding of video," in *Proc. ICIP*, vol. 3, Sep. 2003, pp. 869–872.
- [11] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner–Ziv coding for video: Applications to compression and error resilience," in *Proc. DCC*, Mar. 2003, pp. 93–102.
- [12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. ICC*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [13] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
- [14] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Process.*, vol. 86, no. 11, pp. 3123–3130, Nov. 2006.
- [15] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Oualet, "The DISCOVER codec: Architecture, techniques and evaluation," in *Proc. Picture Coding Symp.*, vol. 6, Lisbon, Portugal, Nov. 2007, pp. 1103–1120.
- [16] D. Varodayan, D. Chen, M. Flierl, and B. Girod, "Wyner–Ziv coding of video with unsupervised motion vector learning," *Signal Process., Image Commun.*, vol. 23, no. 5, pp. 369–378, Jun. 2008.
- [17] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1177–1190, Sep. 2008.
- [18] C. Brites and F. Pereira, "An efficient encoder rate control solution for transform domain Wyner–Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1278–1292, Sep. 2011.
- [19] X. Huang, C. Brites, J. Ascenso, F. Pereira, and S. Forchhammer, "Distributed video coding with multiple side information," in *Proc. PCS*, May 2009, pp. 1–4.
- [20] Y. Zhang, H. Xiong, Z. He, S. Yu, and C. W. Chen, "Reconstruction for distributed video coding: A context-adaptive Markov random field approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1100–1114, Aug. 2011.
- [21] H. V. Luong, L. Raket, X. Huang, and S. Forchhammer, "Side information and noise learning for distributed video coding using optical flow and clustering," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4782–4796, Dec. 2012.
- [22] N. Deligiannis, J. Barbarien, M. Jacobs, A. Munteanu, A. Skodras, and P. Schelkens, "Side-information-dependent correlation channel estimation in hash-based distributed video coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1934–1949, Apr. 2012.
- [23] F. Verbist, N. Deligiannis, M. Jacobs, J. Barbarien, P. Schelkens, and A. Munteanu, "Maximum likelihood motion compensation for distributed video coding," *Integr. Comput., Aided Eng.*, vol. 19, pp. 215–227, Jan. 2012.
- [24] F. Verbist, N. Deligiannis, M. Jacobs, J. Barbarien, P. Schelkens, A. Munteanu, et al., "Probabilistic motion-compensated prediction in distributed video coding," *Multimedia Tools Appl.*, vol. 66, no. 3, pp. 1–26, Mar. 2012.
- [25] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2436–2448, Oct. 2007.
- [26] N. Gehrig and P. Dragotti, "DIFFERENT—Distributed and fully flexible image encoders for camera sensor networks," in *Proc. IEEE ICIP*, vol. 2, Sep. 2005, pp. 690–693.
- [27] C. Yeo, J. Wang, and K. Ramchandran, "View synthesis for robust distributed video compression in wireless camera networks," in *Proc. IEEE ICIP*, vol. 3, Sep./Oct. 2007, pp. 21–24.
- [28] C. Yeo and K. Ramchandran, "Robust distributed multiview video compression for wireless camera networks," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 995–1008, Apr. 2010.
- [29] X. Guo, Y. Lu, F. Wu, D. Zhao, and W. Gao, "Wyner–Ziv-based multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 713–724, Jun. 2008.

- [30] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.
- [31] M. Adrat and P. Vary, "Iterative source-channel decoding: Improved system design using EXIT charts," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 928–941, Jan. 2005.
- [32] Y. Huo, T. Wang, R. G. Maunder, and L. Hanzo, "Iterative source and channel decoding relying on correlation modelling for wireless video transmission," *IET Commun.*, vol. 7, no. 14, pp. 1465–1475, Sep. 2013.
- [33] Y. Huo, C. Zhu, and L. Hanzo, "Spatio-temporal iterative source-channel decoding aided video transmission," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1597–1609, May 2013.
- [34] A. Aaron, R. Zhang, and B. Girod, "Wyner–Ziv coding of motion video," in *Proc. 36th Asilomar Conf. Signals, Syst. Comput. Conf. Rec.*, vol. 1, Nov. 2002, pp. 240–244.
- [35] C. Brites, "Exploiting correlation noise modeling in Wyner–Ziv video coding," Ph.D. dissertation, Inst. Superior Técnico, Tech. Univ. Lisbon, Lisbon, Portugal, Mar. 2011.
- [36] Y.-W. Huang, C.-Y. Chen, C.-H. Tsai, C.-F. Shen, and L.-G. Chen, "Survey on block matching motion estimation algorithms and architectures with new results," *J. VLSI Signal Process. Syst.*, vol. 42, pp. 297–320, Mar. 2006.
- [37] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.
- [38] L. Hanzo, T. Liew, B. Yeap, and R. Tee, *Turbo Coding, Turbo Equalization and Space-Time Coding*. New York, NY, USA: Wiley, 2011.
- [39] S. Cheng and Z. Xiong, "Successive refinement for the Wyner–Ziv problem and layered code design," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3269–3281, Aug. 2005.
- [40] *FhG-HHI 3DV Data* [Online]. Available: <http://sp.cs.tut.fi/mobile3dvtv/video-plus-depth/>
- [41] Mitsubishi Electric Research Lab. *MERL Multiview Video Sequence*, Cambridge, MA, USA [Online]. Available: <http://ftp.merl.com/pub/avetro/mvc-testseq>
- [42] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, *ITU-T Rec. H.264/ISO/IEC 14496-10 AVC: Advanced Video Coding for Generic Audiovisual Services*, Geneva, Switzerland, Mar. 2010.
- [43] *H.264/AVC Reference Software* [Online]. Available: <http://iphome.hhi.de/suehring/tml/>



Yongkai Huo received the B.Eng. degree with distinction in computer science and technology from the Hefei University of Technology, Hefei, China, in 2006, and the M.Eng. degree in computer software and theory from the University of Science and Technology of China, Hefei, in 2009. He is currently pursuing the Ph.D. degree with the Communications, Signal Processing and Control Group, School of Electronics and Computer Science, University of Southampton, Southampton, U.K. He received a scholarship under the China-U.K. Scholarships for Excellence Programme. His research interests include distributed video coding, multiview video coding, robust wireless video streaming, and joint source-channel decoding.



Tao Wang received the B.S. degree in information engineering from the University of Science and Technology of Beijing, Beijing, China, in 2006, and the M.Sc. degree in communication from the University of Southampton, Southampton, U.K., in 2008. He is currently pursuing the Ph.D. degree with the Communications Research Group, Electronics and Computer Science, University of Southampton, Southampton, U.K. His current research interests include joint source/channel coding and distributed video coding.



Robert G. Maunder has been studied with Electronics and Computer Science, University of Southampton, U.K., since October 2000. He received the B.Eng. (Hons.) degree in electronic engineering in July 2003, and the Ph.D. degree in wireless communications and a lectureship in December 2007. His research interests include joint source/channel coding, iterative decoding, irregular coding, and modulation techniques. He has published a number of IEEE papers.



Lajos Hanzo (F'04) received the Degree in electronics in 1976, the Doctorate degree in 1983, and the D.Sc. degree in 2004.

He has held various research and academic posts in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the Chair in telecommunications. He has successfully supervised 80+ Ph.D. students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totaling in excess of 10 000 pages, published 1380+ research entries at the IEEE Xplore, acted both as a TPC and General Chair of the IEEE conferences, presented keynote lectures, and has been awarded a number of distinctions. Currently, he is directing a 100-strong academic research team, involved on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council U.K., the European IST Programme, and the Mobile Virtual Centre of Excellence, U.K. He is an enthusiastic supporter of industrial and academic liaison and he offers a range of industrial courses. He is a Governor of the IEEE VTS. From 2008 to 2012, he was the Editor-in-Chief of the IEEE PRESS and since 2009, he has been a Chaired Professor with Tsinghua University, Beijing. He is a fellow of the Royal Academy of Engineering, the Wolfson Fellow of the Royal Society as well as a fellow of the EURASIP and the IET. He has 18 000+ citations.

AUTHOR QUERIES

- AQ:1 = Please supply index terms/keywords for your paper. To download the IEEE Taxonomy, go to http://www.ieee.org/documents/taxonomy_v101.pdf.
- AQ:2 = Please confirm the volume no. for ref. [31].
- AQ:3 = Please provide the accessed date for refs. [40]–[41] and [43].

IEEE
Proof