

Iterative source and channel decoding relying on correlation modelling for wireless video transmission

Yongkai Huo, Tao Wang, Robert G. Maunder, Lajos Hanzo

School of ECS, University of Southampton, UK
 E-mail: lh@ecs.soton.ac.uk

Abstract: Since joint source-channel decoding is capable of exploiting the residual redundancy in the source signals for improving the attainable error resilience, it has attracted substantial attention. In this treatise, the authors study iterative source-channel decoding (ISCD) aided video communications, where the video signal redundancy is modelled by a first-order Markov process. Firstly, the authors derive reduced-complexity formulas for the first-order Markov modelling (FOMM) aided source decoding. Then they propose a bit-based iterative horizontal-vertical scanline model (IHVSM) aided source decoding algorithm, where a horizontal and a vertical source decoder are employed for exchanging their extrinsic information using the iterative decoding philosophy. The iterative IHVSM aided decoder is then employed in a forward error correction (FEC) encoded uncompressed video transmission scenario, where the IHVSM and the FEC decoder exchange softbit-information for performing turbo-like ISCD for the sake of improving the reconstructed video quality. Finally, the authors benchmark the attainable system performance against a near-lossless H.264/AVC video communication system and the existing FOMM-based softbit source decoding scheme. The authors simulation results show that E_b/N_0 improvements in excess of 2.8 dB are attainable by the proposed technique in uncompressed video applications.

1 Introduction

Shannon's separation theorem [1] states that reliable transmission can be accomplished by separate source coding using lossless entropy codes and channel coding under the idealised assumption of Gaussian channels and potentially infinite encoding/decoding delay and complexity. However, a finite-delay, finite-complexity source encoder fails to remove all the redundancy residing in the source signals, such as audio and video. Hence, joint source-channel coding (JSCC) [2] was proposed for additionally exploiting the source statistics for the sake of improved iterative source-channel decoding. Furthermore, Görtz [3, 4] proposed the iterative source-channel decoding (ISCD) philosophy, which performed turbo-like iterative decoding by exchanging extrinsic information between the source encoder and decoder. Softbit source decoding (SBSD) was proposed for error concealment of speech signals by Fingscheidt and Vary [5] using softbit information, where the correlation of adjacent speech source samples were modelled by a first-order Markov process. Afterwards, Adrat *et al.* [6, 7] developed SBSB for iterative source-channel decoding using both forward- and backward-oriented calculations.

Similar to speech, joint source-channel decoding of image and video signals also attracted substantial attention. For example, Kliever *et al.* [8, 9] proposed an ISCD scheme for images modelled by a Markov random field (MRF) by exploiting that any pixel can exchange extrinsic information with its eight neighbours for exploiting the residual spatial

correlations residing in the encoded image. Kliever and Thobaben [10] proposed a novel symbol-based soft-input a-posteriori probability (APP) decoder for packetised variable length coded [11] source signals transmitted over wireless channels, where the Markov-modelled residual redundancy generated after source encoding was exploited for achieving an improved error protection. In the context of distributed source coding, the authors of [12] proposed an error-resilient JSCC scheme using a Slepian-Wolf (SW) codec, which exploited the knowledge of both the channel statistics and the correlation between specific video frames and their reference frames. In [13], an H.264 [14] video telephone scheme was proposed using ISCD. Firstly, the H.264 bitstream was segmented into the partitions A, B and C [14]. Then the three partitions were encoded by variable code-rate short block codes (SBC), which artificially imposed redundancy on the H.264-encoded bitstreams. This artificial redundancy was then exploited by the softbit source decoder, which performs iterative decoding by exchanging extrinsic information with the channel decoder. In [15], a double low-density parity-check (LDPC) code was proposed for joint source and channel coding, where two concatenated LDPC codes were employed as the source LDPC and channel LDPC, respectively. At the receiver, the source LDPC and channel LDPC performed joint decoding by exchanging extrinsic information.

The traditional lossy video coding methods such as MPEG and the ITU-T H.26x codecs have been researched for decades [16]. However, they impose a high encoder complexity, which may become excessive in wireless

personal area networks (WPAN) [17, 18], for example. Secondly, the intrinsic latency may violate the delay budget of delay-sensitive applications, such as interactive gaming [19]. Thirdly, some video quality degradation is inevitable and remains unrecoverable at the receiver, which may be unacceptable in high-quality medical applications [20, 21]. Last but not the least, compressed video streaming is limited to certain devices, categories where the matching encoding/decoding techniques are available. A transcodec converting between compressed video formats is required, when a device has to relay the received video stream to another device employing a different compression technique, which may increase both the cost and complexity. Uncompressed video transmission [17, 19] meets the requirements of high-quality home networking and quality/delay-sensitive medical applications, where low-complexity transmitters deliver unimpaired video signals, which may be processed by the receivers without any video decoder. The state-of-the-art techniques are reviewed below.

The emerging 60 GHz WPAN within the IEEE 802.15.3c standard family [22, 23] is designed for short-range (<10 m) transmission of very-high-speed (>2 Gb/s) multimedia data to computer terminals and consumer appliances centered around an individual person's workspace, such as in residential rooms, offices etc. The WirelessHD specification [24, 25], as another WPAN standard, increases the maximum data rate to 28 Gb/s, which supports the transmission of either compressed or uncompressed digital high definition (HD) multimedia signals. In recent years, numerous investigations have been conducted in uncompressed video transmissions. Singh *et al.* [17, 26, 27] developed a system, where both unequal error protection (UEP) and automatic repeat request (ARQ) protocols were conceived for achieving an improved video quality. The authors of [28] investigated the specific technical challenges imposed by mm-wave systems supporting reliable video streaming using multi-beam transmissions. A flexible UEP method was proposed for the uncompressed video context in [29], which offers an improved visual quality and resource efficiency over both conventional UEP and equal error protection.

A high amount of redundancy is present in the transmitted uncompressed video, which manifests itself in terms of a high adjacent-pixel correlation and should be exploited at the receiver for concealing the pixel errors in the reconstructed video. Motivated by the congenial principle of ISCD, we design an ISCD technique for video communication, where the video-domain redundancy is exploited for iteratively correcting video impairments. Specifically, we propose a sophisticated iterative horizontal-vertical scanline model (IHVSM) aided iterative source decoding model, which iteratively exchanges extrinsic information with the forward error correction (FEC) codec for the sake of exploiting the intra-frame pixel-domain redundancy. Against this background, our novel contributions are

1. We conceive a two-dimensional (2D) IHVSM technique, which exchanges extrinsic information between the rows and columns of a video frame.
2. We apply our proposed technique in an uncompressed video transmission scenario for improving the reconstructed video quality.
3. Finally, we reduce the complexity of the 1D decoding algorithms derived by Vary and co-workers [6, 7] by

designing a novel trellis representation of the Markov process and by deriving its decoding rules.

This rest of this paper is organised as follows. In Section 2, we introduce the general architecture of the iterative source decoding model, which uses the proposed IHVSM decoder. In Section 3, we present the technique of decoding the first-order Markov process as well as the associated iterative decoding principle, which will be employed in the IHVSM decoder introduced in Section 2. Then our IHVSM aided iterative source decoding model will be combined with the FEC codec in an uncompressed video communication scenario in Section 4. Finally, our conclusions are offered in Section 5.

2 Iterative source decoding model

A 1D iterative system model has been proposed and analysed by Vary and co-workers [4–7] in the context of audio signals. Since their model was designed for 1D signals, it cannot directly exploit the 2D correlation of practical video signals. In this section, the 2D video correlation is modelled and exploited by our proposed IHVSM, which consists of multiple horizontal and vertical scanline-based Markov processes. Moreover, the IHVSM exchanges extrinsic information between the horizontal and vertical scanlines by performing iterative source decoding, thereby correcting the errors within the 2D video frames. We initially focus our attention on the receiver side. The details of applying the 2D IHVSM decoder in FEC encoded uncompressed video transmission will then be illustrated in Section 4. Let us commence by stipulating the following assumptions:

- x_i : the m -bit pattern of pixel 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 1st 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 1st frame.

The 2D iterative source decoding model is displayed in Fig. 1, which only accepts the soft bit input and generates the corresponding hard-decision-based pixel as output. We will discuss the details of 2D iterative source decoding techniques below.

2.1 Softbit input

Let us now focus our attention on the 1st original video frame carrying $(t+1)$ consecutive and hence correlated bit patterns, for example, x_0, \dots, x_t . Owing to the channel-impairments, the receiver can only reconstruct the error-infested version of x_0, \dots, x_t . Again, the 2D iterative source decoder shown in Fig. 1 only accepts softbit [Softbit source decoding was proposed in [5]. Instead of expressing a bit as 0 or 1, a softbit represents one bit of information using a floating-point number,

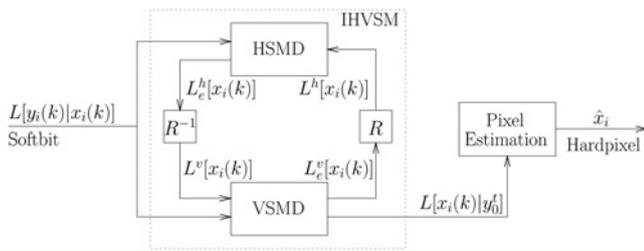


Fig. 1 Iterative source decoding architecture using IHVSM, where R represents reordering of the video pixels

indicating our confidence in that bit.] information as its input. Then, the receiver has to estimate the softbit information of each bit representing the original pixels x_0, \dots, x_t , namely y_0, \dots, y_t , which carry the error-infested bit sequence $y_0 \binom{m-1}{0}, \dots, y_t \binom{m-1}{0}$. The softbit information represented by the log-likelihood ratios (LLR) may be readily derived from the softbit patterns y_0, \dots, y_t , yielding $L[y_i(k)|x_i(k)]$ in Fig. 1, which indicates the receiver's confidence in the original m -bit pixel $x_i(k)$.

2.2 Iterative source decoding

Given the softbit LLR input $L[y_i(k)|x_i(k)]$ of Fig. 1, our 2D iterative source decoder may be invoked for correcting the effects of channel-errors on the error-infested m -bit sequence of $(t+1)$ pixels $y_0 \binom{m-1}{0}, \dots, y_t \binom{m-1}{0}$.

Two stages are involved in the iterative source decoding process, namely the IHVSM-based decoding and the related pixel estimation, as seen in Fig. 1.

2.2.1 IHVSM-based decoding: At the first stage, two similar source decoders are employed, namely the horizontal scanline model decoder (HSMD) modelling the video correlation in the horizontal direction and the vertical scanline model decoder (VSMD) modelling the video correlation in the vertical direction. Let us now consider H horizontal scanlines and V vertical scanlines, which are modelled by H HSMDs and V VSMDs, respectively. Then the IHVSM decoding may be performed based on the $(H \times V)$ -pixel block, which is represented by the $(H \cdot V)$ - m -softbit patterns y_i . Here, we consider the HSMD as an example for highlighting the decoding process. The m -softbit-based pixel value y_i is used for generating the systematic LLR information $L[y_i(k)|x_i(k)]$, which is then input to the HSMD without the assistance of any channel decoding. However, our technique may be conceived with arbitrary channel codecs. The HSMD also exploits the a-priori LLR information $L^h[x_i(k)]$ generated from the extrinsic information $L_e^v[x_i(k)]$ [The first-order Markov process aided decoding algorithm will be detailed in Section 3, along with the associated iterative decoding principle.] provided by the vertical decoder. The horizontal decoder independently performs source-modelling aided soft decoding of each of the H horizontal scanlines. Hence, following horizontal decoding, the extrinsic LLR information $L_e^h[x_i(k)]$ may be generated for all H scanlines. Then the relevant $(H \times V)$ -line extrinsic information block will be appropriately reordered in the block R^{-1} of Fig. 1 for generating the a-priori information $L^v[x_i(k)]$ for the vertical decoder.

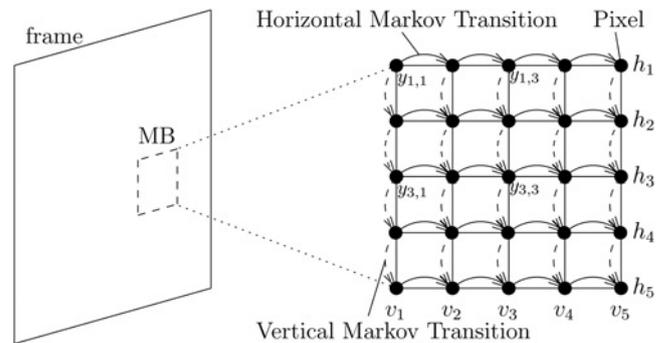


Fig. 2 Structure of horizontal and vertical scanlines and their corresponding Markov processes $y_{i,j}$ represents the received pixel at the position (i, j) in the MB, which corresponds to the error-infested version of original pixel $x_{i,j}$

For further illustration of the IHVSM decoding process, consider the example of the horizontal and vertical scanlines displayed in Fig. 2, where the pixels connected by the solid/dashed arrows belong to a horizontal/vertical Markov process. Specifically, the five pixels of the first row in Fig. 2 are modelled by a five-pixel Markov process h_1 . Let us assume that the five pixel values available at the receiver may be expressed as $y_{1,1}, \dots, y_{1,5}$ of Fig. 2, each of which consists of m soft values indicating the decoder's confidence in each of the m systematic bits. [The m soft values may be acquired in different ways in different applications. In the application of distributed video coding (DVC), the m soft values of the pixels in the Wyner-Ziv (WZ) frames are estimated from the key frames, which are intra-coded and transmitted to the receiver [30].] The corresponding a-priori LLR information

$$L^h \left[x_{1,1} \binom{m-1}{0} \right], \dots, L^h \left[x_{1,5} \binom{m-1}{0} \right]$$

for the five pixels is provided by the VSMD decoder. The $(5 \times m)$ soft values representing the five pixels can exchange extrinsic information with each other, when the HSMD decoder is processing scanline h_1 . Similarly, the pixel confidences of all five horizontal scanlines may be improved by performing HSMD on h_1, \dots, h_5 , respectively. Note that the pixels only shared their information within the horizontal scanlines in the HSMD process. After HSMD decoding, the extrinsic information provided by the (5×5) pixels, namely the $(5 \times 5 \times m)$ soft values of

$$L_e^h \left[x_{1,1} \binom{m-1}{0} \right], \dots, L_e^h \left[x_{5,5} \binom{m-1}{0} \right]$$

are generated. Then, these $(5 \times 5 \times m)$ extrinsic information values will be reordered into vertical scanlines by the block R^{-1} of Fig. 1, which will be utilised as a-priori information by the VSMD decoder. Then the VSMD decoder will decode the vertical Markov process v_1 based on the systematic information $y_{1,1}, \dots, y_{5,1}$, whereas also exploiting the a-priori LLR information

$$L^v \left[x_{1,1} \binom{m-1}{0} \right], \dots, L^v \left[x_{5,1} \binom{m-1}{0} \right]$$

reordered by R^{-1} . Similar to the HSMD, the VSMD will improve the decoder's confidence related to the pixels by exchanging extrinsic information in the vertical direction. After VSMD processing of the vertical scanlines v_1, \dots, v_5 , respectively, a full iteration of the IHVSM scheme is completed and the HSMD will exploit the extrinsic information fed back by the VSMD decoder in order to start the next iteration. This iterative process will terminate after the affordable number of iterations. Observe that any two pixels of an MB are related to each other according to our Markov model and can exchange information with each other with the aid of the IHVSM decoder. Moreover, the IHVSM decoder performs decoding on a block-by-block basis, as defined in the system.

2.2.2 Pixel estimation: Following the IHVSM decoding process, the relevant a-posteriori information is generated at the parameter estimation stage by summing the systematic LLR information $L[y_i(k)/x_i(k)]$, the a-priori LLR information $L^v[x_i(k)]$ and the extrinsic LLR information $L_e^h[x_i(k)]$, where the a-posteriori information $L[x_i | y_0^t]$ may be exploited either by the minimum mean-square error (MMSE) decoder or the maximum a-posteriori probability (MAP) decoder for estimating the m -bit pattern x_i and ultimately to output the estimated original pixel \hat{x}_i [5], which may be formulated as

- MAP estimator

$$\hat{x}_i = \arg \max_{x_i \in X_m} p(x_i | y_0^t) \quad (1)$$

- MMSE estimator

$$\hat{x}_i = \sum_{x_i \in X_m} x_i p(x_i | y_0^t) \quad (2)$$

Both the MMSE estimator of (2) and the MAP estimator of (1) may be deemed to be symbol-based estimators, since they are based on the APP $p(x_i | y_0^t)$ of the m -bit pattern x_i , which is conditioned on the received frame of bit patterns y_0^t . However, since the specific bits $x_i(0), \dots, x_i(m-1)$ of the pixels are independent of each other, if the bit-based a-posteriori probability $p[x_i(k) | y_0^t]$ is provided by the iterative decoding process, the symbol-based a-posteriori probability $p(x_i | y_0^t)$ in (1) and (2) may be derived from the bit-based information as their product

$$p(x_i | y_0^t) = \prod_{k=0}^{m-1} p[x_i(k) | y_0^t] \quad (3)$$

Furthermore, the symbol-based MMSE estimation rule may also be derived from the bit-based probability $p[x_i(k) | y_0^t]$ as

$$L[x_i(k) | y_0^t] = \log \frac{\sum_{x_i \in X_m} \beta_i(x_i) \gamma_i(x_i) \sum_{x_{i-1} \in X_m} p(x_i | x_{i-1}) \alpha_{i-1}(x_{i-1})}{\sum_{x_i \in X_m} \beta_i(x_i) \gamma_i(x_i) \sum_{x_{i-1} \in X_m} p(x_i | x_{i-1}) \alpha_{i-1}(x_{i-1})} \quad (5)$$

follows:

$$\begin{aligned} \hat{x}_i &= \sum_{x_i \in X_m} x_i p(x_i | y_0^t) = \sum_{x_i \in X_m} x_i \prod_{k=0}^{m-1} p[x_i(k) | y_0^t] \\ &= \sum_{k=0}^{m-1} 2^k p[x_i(k) = 1 | y_0^t] \sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \prod_{\substack{l=0 \\ l \neq k}}^{m-1} p[x_i(l) | y_0^t] \end{aligned}$$

It may be readily shown that

$$\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \prod_{\substack{l=0 \\ l \neq k}}^{m-1} p[x_i(l) | y_0^t] = 1$$

hence the symbol-based MMSE estimator \hat{x}_i can be derived from the bit-based probabilities as

$$\hat{x}_i = \sum_{k=0}^{m-1} 2^k p[x_i(k) = 1 | y_0^t] \quad (4)$$

Finally, the hard pixel \hat{x}_i of Fig. 1 may be returned as the final estimate of the original pixel x_i .

3 Markov modelled softbit source decoding with reduced complexity

In this section, we focus our attention on the decoding of the first-order Markov process as well as on its iterative decoding principle. The employment of first-order Markov modelling aided softbit source decoding (FOMM-SBSD) was discussed in [5–7]. The a-posteriori log-likelihood ratio of bit $x_i(k)$ conditioned on y_0^t was derived in [6, 7], which may be expressed as in (5), where the components γ , α and β are defined by Adrat and Vary [7] (see (5)).

Below, we consider the determination of the pixels x_0, \dots, x_t based on the soft pixel information y_0, \dots, y_t . Firstly, we propose our novel trellis representation of the first-order Markov process in Section 3.1, namely the pixel sequence x_0, \dots, x_t . We will then detail the decoding rules of our trellis in Section 3.2, which is a reduced complexity version of the technique proposed by Adrat and Vary's team [7]. Finally, we will present our extrinsic information formula invoked for iterative decoding in Section 3.3, whereas the computational complexity imposed is analysed in Section 3.4.

3.1 Trellis representation of first-order Markov chain

The traditional MAP decoder trellis for an RSC code is detailed in Figs. 4.6 of [31], where the nodes indicate the memory states of the RSC encoder, whereas the lines indicate the state transitions. Specifically, there are four states, each of which has two possible transitions indicating the input bits of 0 and 1. All the transitions are determined by the generator polynomials of the RSC encoder.

In contrast to the traditional one, the trellis of the first-order Markov process is shown in Fig. 3, where the nodes are the

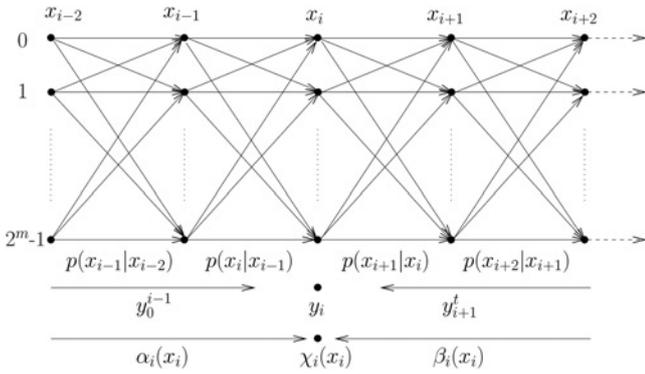


Fig. 3 Trellis of first-order Markov process x_0, \dots, x_t for BCJR decoding, where $p(x_{i+1}|x_i)$ is the Markov transition probability

trellis states, namely the m -bit pixels x_{i-2}, \dots, x_{i+2} , while the lines indicate the state transitions. Moreover, the probability of a transition from state x_i to state x_{i+1} is represented by $p(x_{i+1}|x_i)$, which is the state transition probability of the related Markov process. There are 2^m states for the m -bit pixels x_{i-2}, \dots, x_{i+2} , each of which has 2^m possible transitions.

Our proposed trellis is different from the traditional trellis in the following aspects. Firstly, the states of the traditional trellis indicate the memory states of the RSC encoder, whereas the states of our trellis indicate the legitimate m -bit pixel values. Secondly, the transitions of the traditional trellis indicate the binary inputs 0, 1, whereas the transitions of our trellis indicate the Markov state transitions. Finally, there are two possible transitions for each state of the traditional trellis, whereas there are 2^m possible transitions for each state of our trellis.

3.2 BCJR decoding of first-order Markov chain

The existing APP generation rule conceived for first-order Markov modelled SBSB is shown in (5). However, we will demonstrate that the complexity imposed may be reduced by invoking our novel trellis introduced in Section 3.1.

As illustrated in Section 2, the soft Markov processes y_0, \dots, y_t [As detailed in Section 2, each soft pixel y_i consists of m floating values $y_i(0), \dots, y_i(m-1)$, representing the confidences of the original m bits $x_i(0), \dots, x_i(m-1)$.] are known at the Markov decoder, which represented the error-infested soft information of the original Markov process x_0, \dots, x_t . The objective of the Markov decoder is to estimate the quantised pixels x_0, \dots, x_t based on the soft pixels. Given the soft information y_0, \dots, y_t of the Markov process, the quantised pixel x_i may be determined by the APP

$$p(x_i|y_0, \dots, y_t) = p(x_i|y_0^t) \quad (6)$$

where we have $x_i \in X_m, 0 \leq i \leq t$. In the Appendix, we show that the bit-based LLR version of the APP $p(x_i|y_0^t)$ may be expressed as

$$L[x_i(k)|y_0^t] = \ln \frac{\sum_{\substack{x_i(k)=0 \\ x_i \in X_m}} \beta_i(x_i) \chi_i(x_i) \alpha_i(x_i)}{\sum_{\substack{x_i(k)=1 \\ x_i \in X_m}} \beta_i(x_i) \chi_i(x_i) \alpha_i(x_i)} \quad (7)$$

and based on this, the decoder may determine the bit $x_i(k)$.

We have hence derived the final rule of determining the bit-based a-posteriori probability LLR, which is represented

in (20). For the conventional determination rule formulated in (5), the components β, χ and α are defined in [7], which impose a similar complexity to our definitions in (16)–(18). However, the inner summation $\sum p(x_i|x_{i-1}) \alpha_{i-1}(x_{i-1})$ of (5) is avoided in our decoding rule of (20), which reduces the computational complexity imposed.

Furthermore, the Jacobian logarithm [31] may be readily applied for deriving the log-domain representation of our algorithm.

3.3 Extrinsic information exchange for iterative decoding

A limitation of the formulas provided in Section 3.2 is that they cannot be directly used for iterative decoding, since they cannot exploit the a-priori LLR information $L[x_i(k)]$, which was generated from the ‘extrinsic’ information gleaned from the other decoder involved in the turbo-like iterative decoding process [32]. The rules of iterative source and channel decoding were derived by Vary and co-workers [6, 7]. To make use of the a-priori LLR information $L[x_i(k)]$, the combined bit-based log-likelihood information [Similar to (16), a constant normalisation factor is neglected since it will be canceled during the calculation.] may be utilised as [7]

$$\gamma_i(x_i) = \exp \sum_{k=0}^{m-1} \frac{\bar{x}_i(k)}{2} \{L[x_i(k)] + L[y_i(k)|x_i(k)]\} \quad (8)$$

where the symbol-based m -bit information γ is the combination of the bit-based log-likelihood a-priori information $L[x_i(k)]$ and of the channel information $L[y_i(k)|x_i(k)]$. We note in this context that γ of (8) contains more valuable information than the channel information χ . By replacing χ of (20) with γ of (8), we have the following formula:

$$L[x_i(k)|y_0^t] = \ln \frac{\sum_{\substack{x_i \in X_m \\ x_i(k)=0}} \beta_i(x_i) \gamma_i(x_i) \alpha_i(x_i)}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \beta_i(x_i) \gamma_i(x_i) \alpha_i(x_i)} \quad (9)$$

Moreover, the symbol-based a-bretonposteriori probability of the first and last m -bit patterns may be expressed as follows:

$$\begin{aligned} p(x_0 \wedge y_0^t) &= \beta_0(x_0) \gamma_0(x_0) p(x_0) \\ p(x_t \wedge y_0^t) &= \gamma_t(x_t) \alpha_t(x_t) \end{aligned} \quad (10)$$

Similar to the BCJR decoding technique of classic turbo codes [31], the bit-based a-bretonposteriori LLR $L[x_i(k)|y_0^t]$ may be split into three components, namely the a-priori information $L[x_i(k)]$, the channel information $L[y_i(k)|x_i(k)]$ and the extrinsic information $L_e[x_i(k)]$. Specifically, the bit-based a-posteriori information $L[x_i(k)|y_0^t]$ may be formulated as in (11)

$$\begin{aligned} L[x_i(k)|y_0^t] &= L[x_i(k)] + L[y_i(k)|x_i(k)] \\ &+ \ln \frac{\sum_{\substack{x_i \in X_m \\ x_i(k)=0}} \beta_i(x_i) \gamma_i^{\text{[ext]}}[x_i(k)] \alpha_i(x_i)}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \beta_i(x_i) \gamma_i^{\text{[ext]}}[x_i(k)] \alpha_i(x_i)} \\ &= L[x_i(k)] + L[y_i(k)|x_i(k)] + L_e[x_i(k)] \end{aligned} \quad (11)$$

where the extrinsic information component $\gamma_i^{\text{[ext]}}[x_i(k)]$ may

be expressed as

$$\gamma_i^{[\text{ext}]}[x_i(k)] = \exp \sum_{l=0, l \neq k}^{m-1} \frac{\bar{x}_i(l)}{2} \{L[x_i(l)] + L[y_i(l)|x_i(l)]\} \quad (12)$$

3.4 Complexity analysis

The complexity of our proposed first-order Markov process-based decoder can be attributed to the calculation of $\gamma_i(x_i)$ in (8), $\alpha_i(x_i)$ in (17), $\beta_i(x_i)$ in (18) and $L[x_i(k)|y_0^t]$ in (9). As shown in Fig. 3, the trellis size is $(2^{m \cdot t})$, where t is the length of the Markov process. Similar to the BCJR decoding rules proposed in [33], the decoding of the $(2^{m \cdot t})$ -state trellis of Fig. 3 may be generalised into the following two stages:

- *Calculation of γ , α and β* : These operations are carried out across the entire Markov trellis of Fig. 3, which imposes the complexity of m , 2^m , 2^m for each trellis state, as suggested by (8), (17), (18), respectively. Hence the associated computational costs are on the order of $O(2^{m \cdot t} \cdot m)$, $O(2^{2m \cdot t})$ and $O(2^{2m \cdot t})$ for γ , α and β , respectively.
- *Calculation of $L[x_i(k)|y_0^t]$* : This operation is carried out for all the $t \cdot m$ bits of the t -pixel Markov process, which imposes a complexity of 2^m for each bit. Hence the computational cost is on the order of $O(2^{m \cdot t} \cdot m)$.

Therefore the overall complexity imposed by our proposed decoder is $O(2 \cdot 2^{2m \cdot t} + 2 \cdot 2^{2m \cdot t} \cdot m)$, when decoding the t -pixel Markov process.

Similarly, for the conventional Markov process-based decoding technique of (5) [7], the associated computational costs are on the order of $O(2^{2m \cdot t} \cdot m)$, $O(2^{2m \cdot t})$, $O(2^{2m \cdot t})$ for γ , α , β , respectively, where the relevant definitions of γ , α , β [7] are not included in this treatise for the sake of space economy. However, the inner summation $\sum p(x_i|x_{i-1}) \alpha_{i-1}(x_{i-1})$ in (5) imposes $2^{2m \cdot t}$ more operations than (9), which is calculated in advance. Therefore the complexity of the conventional Markov process-based decoding algorithm is on the order of $O(3 \cdot 2^{2m \cdot t} + 2 \cdot 2^{2m \cdot t} \cdot m)$.

Based on the discussions above, the percentage of complexity reduction achieved by our proposed decoder may be expressed as $2^m / (3 \cdot 2^m + 2 \cdot m)$, which increases upon increasing m . Specifically, complexity reductions of 25 and 33% may be achieved for $(m = 1)$ and $(m = 8)$, respectively. Note that the decoding complexity of the first-order Markov process increases exponentially with the number of bits per symbol m . Hence, a quantiser may be employed for striking a tradeoff between the complexity imposed and the attainable performance.

4 IHVSM aided uncompressed video transmission

Note that the peak to-signal-noise ratio (PSNR) metric is employed for quantifying the reconstructed video quality in this section. To avoid having infinite PSNR values when a video is perfectly reconstructed, we artificially set the total averaged mean-squared error (MSE) value between the reconstructed and the original frame to a minimum value of 1. This is justified, since the same technique is employed in the H.264 reference software JM. Hence the maximum

unimpaired video PSNR that may be obtained at the receiver is about 48.1 dB.

In this section, we consider the transmission of RSC encoded uncompressed video through a Rayleigh channel. The demodulator estimates the softbit information at the receiver, thereby meeting the essential prerequisites of Section 2 for the application of our proposed iterative source decoding technique. Furthermore, the IHVSM and the RSC codec will exchange extrinsic information for concealing the errors within the video frames.

4.1 System configuration

In this section, we consider the scenario of uncompressed video transmission, which may be employed for home networking [17]. The system's architecture is displayed in Fig. 4. At time instant i , the transmitter has to convey a video pixel x_i , which is mapped to the m -bit pattern $x_i \binom{m-1}{0}$. This pixel-to-bit mapper may include a quantisation function [16]. We treat $(t + 1)$ consecutive and hence correlated m -bit patterns eg. x_0, \dots, x_t as a frame. Let us consider the first 2D video frame, for example, which will be interleaved by a bit-based interleaver of length $(t + 1) \cdot m$. Then the signals are RSC encoded and transmitted to the receiver through a Rayleigh channel using BPSK modulation. At the receiver, the demodulator will generate the softbit information y_0, \dots, y_t of the video pixels, which will be input to our proposed ISCD model, namely to the 'IHVSM-RSC' and 'pixel estimation' blocks of Fig. 4. Then, the HSMD, VSMD component decoders of our IHVSM and the RSC codec will perform a typical three-stage [34] iterative source-channel decoding for concealing the error effects imposed on the video frames. Specifically, in Fig. 4 the RSC-HSMD consists of the inner decoder, whereas the VSMD is the outer decoder. The reconstructed hard-decision-based pixel \hat{x}_i can then be acquired after the pixel estimation stage of Fig. 4. Although the system employs an RSC channel encoder, for our design-example, arbitrary systematic FEC codecs may be utilised.

We compare the performance of our scheme recorded for the video sequences Akiyo, Coastguard and Football against that of the existing system model. Video sequences stored in (176×144) -pixel quarter common intermediate format (QCIF) and 4:2:0 YUV representation are employed. Moreover, the horizontal and vertical decoders perform

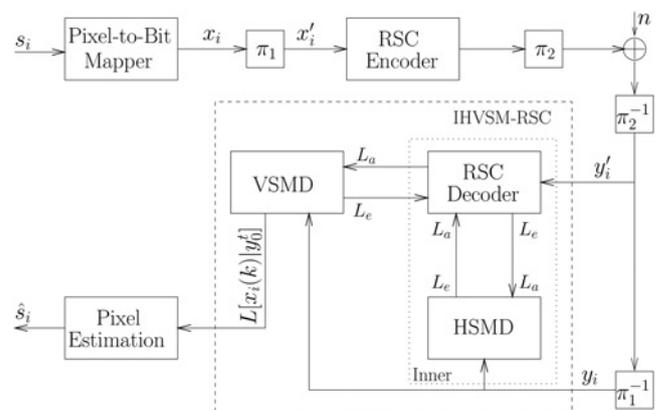


Fig. 4 System architecture of IHVSM-RSC for communication of RSC encoded uncompressed video through wireless channel

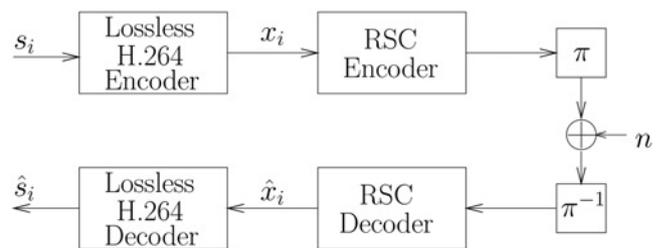
Table 1 Table of parameters employed for the video sequences

	Akiyo	Coastguard	Football
representation format	YUV 4:2:0 QCIF	YUV 4:2:0 QCIF	YUV 4:2:0 QCIF
bits per pixel	8	8	8
FPS	30	15	15
number of frames	100	100	100
bitrate, Mbps	9.1	4.6	4.6
'natural' code rate	1/8.4	1/2.4	1/1.8
channel modulation	Unc-Ray BPSK	Unc-Ray BPSK	Unc-Ray BPSK

Unc-Ray stands for uncorrelated Rayleigh

iterative decoding based on (88×72) -pixel blocks. For the video sequences represented in QCIF, each QCIF luminance frame is divided into (2×2) blocks and each QCIF chroma frame is transmitted in a single block. Note that for simplicity, the uncompressed video bits are transmitted through an uncorrelated non-dispersive Rayleigh channel using BPSK modulation. These parameters are summarised in Table 1. The MMSTT of the first-order Markov model was derived from the original video pixels, which may be utilised by the horizontal and vertical source decoders as side-information for improving the achievable error resilience. We mainly rely on two types of curves for characterising the video quality, namely the PSNR against the channel SNR per bit, namely E_b/N_0 curves and the bit-error ratio (BER) [For the uncompressed video scenario, BER defines the BER of the reconstructed video compared to the original video. For the H.264 compressed bitstream, BER defines the bit error ratio of the bitstream decoded at the receiver.] against E_b/N_0 curves.

Shannon's channel capacity theorem [1] was proposed for the transmission of i.i.d source. Hence, to be in line with the channel capacity theory, we have to consider the true entropy of the video sequence, when calculating the energy efficiency per bit. More explicitly, any redundancy inherent in the encoded sequence has to be taken into account by shifting the BER against E_b/N_0 curves to the right, regardless, whether the redundancy is natural source redundancy or whether it was artificially imposed by channel coding. In our case, substantial redundancy resides in the video source signal, since here we do not employ any video encoder. Hence, the true amount of non-redundant information transmitted to the receiver is given by the entropy of the video sequence. Assuming that the total uncompressed size of a video file is S_r bits and the entropy of this video source file is S_e , we might interpret the raw video file as being 'naturally' losslessly encoded from S_e i.i.d bits, to generate S_r bits where the code rate is $r = S_e/S_r$. According to Shannon's source coding theorem [1], the entropy of the

**Fig. 5** Architecture of the Lossless-H.264-RSC system, where the H.264 codec operates in the near-lossless encoding mode

video source file represents the lowest achievable rate at which the source may be losslessly represented. Hence, in our simulations the E_b/N_0 (dB) value is calculated as $E_b/N_0 = 10 \log_{10}((E_b S_r)/(N_0 S_e))$. However, no widely recognised technique exists for quantifying the entropy of a realistic video source. As a practical solution, we opted for using the near-lossless coding mode of the H.264 codec [14, 16] to encode the source video for the sake of approximating its entropy. The 'natural' code rates (NCR) of the Akiyo, Coastguard and Football sequences used in our simulations are listed in Table 1. Quantitatively, we found that the 'natural' code rates of the three sequences were 1/8.4, 1/2.4 and 1/1.8 for the QCIF scenario, which corresponds to the maximum achievable compression ratios of 8.4, 2.4 and 1.8, respectively. The corresponding parameters for all the scenarios considered are shown in Table 1, where the PSNR values correspond to the maximum quality of the error-free source decoded video at the receiver.

We will benchmark the performance of our system against three schemes, namely against that of the MAP-based and MMSE-based hard decoding schemes (MAP-HD/MMSE-HD), where no softbit source decoding is employed, as well as against the FOMM-SBSD scheme, where no IHVSM decoding is employed. In the latter case only one of the two decoders is activated. A brief comparison of the four schemes is shown in Table 2.

4.2 Benchmarks

To provide sufficiently deep insights into the performance of our proposed system, let us now describe the benchmarks. Firstly, we benchmark our IHVSM-RSC scheme against the RSC aided uncompressed video transmission system, where no source correlation is exploited at the receiver. We refer to this as uncompressed-RSC scheme for simplicity.

Then, to analyse the benefits of the RSC codec in the IHVSM-RSC scheme, the results of the IHVSM scheme will also be provided, which is a non-channel-encoded version of the IHVSM-RSC regime. We will also benchmark the performance recorded for the three video sequences against that of the MAP-based and MMSE-based

Table 2 Comparison of IHVSM-RSC and the benchmarks: MMSE-HD [5], Uncompressed-RSC, Lossless-H.264-RSC, FOMM-SBSD [7], FOMM-RSC [7], IHVSM, where MMSE estimator is employed

	MMSE-HD	Uncompressed-RSC	L-H.264-RSC	FOMM-SBSD	FOMM-RSC	IHVSM	IHVSM-RSC
dimension	1	1	1	1	1	2	2
bits num to decode	8	10 000	10 000	704	50 688	50 688	50 688
side information	none	none	none	1 × MMSTT	1 × MMSTT	1 × MMSTT	1 × MMSTT
code rate	NCR	$(1/2) \times \text{NCR}$	$(1/2)$	NCR	$(1/2) \times \text{NCR}$	NCR	$(1/2) \times \text{NCR}$
delay (frames)	0	0	8	0	0	0	0
col (trans./rec.)	low/low	low/low	high/low	low/high	low/high	low/high	low/high

hard decoding schemes (MAP-HD/MMSE-HD), where no source correlation is exploited by the receiver. As further benchmarks, both the FOMM-SBSD and the FOMM-SBSD aided RSC scheme (FOMM-RSC) were also invoked.

Finally, the system employing the near-lossless H.264 codec of Fig. 5 is invoked. Specifically, the H.264 codec [14] is configured using the smallest quantisation index. Furthermore, both predicted (P) and bidirectional predicted

(B) frames are enabled. More specifically, the 30-frame sequences were encoded into an intra-coded frame (I), followed by the periodically repeated PBBBBBBB frames. Again, this enables the H.264 codec to generate a near-lossless video bitstream. However, a delay of eight frames was introduced by the employment of B frames. As shown in Fig. 5, the RSC codec of the IHVSM-RSC scheme was utilised as our FEC codec for protecting the losslessly encoded bitstream. We refer to this system as the

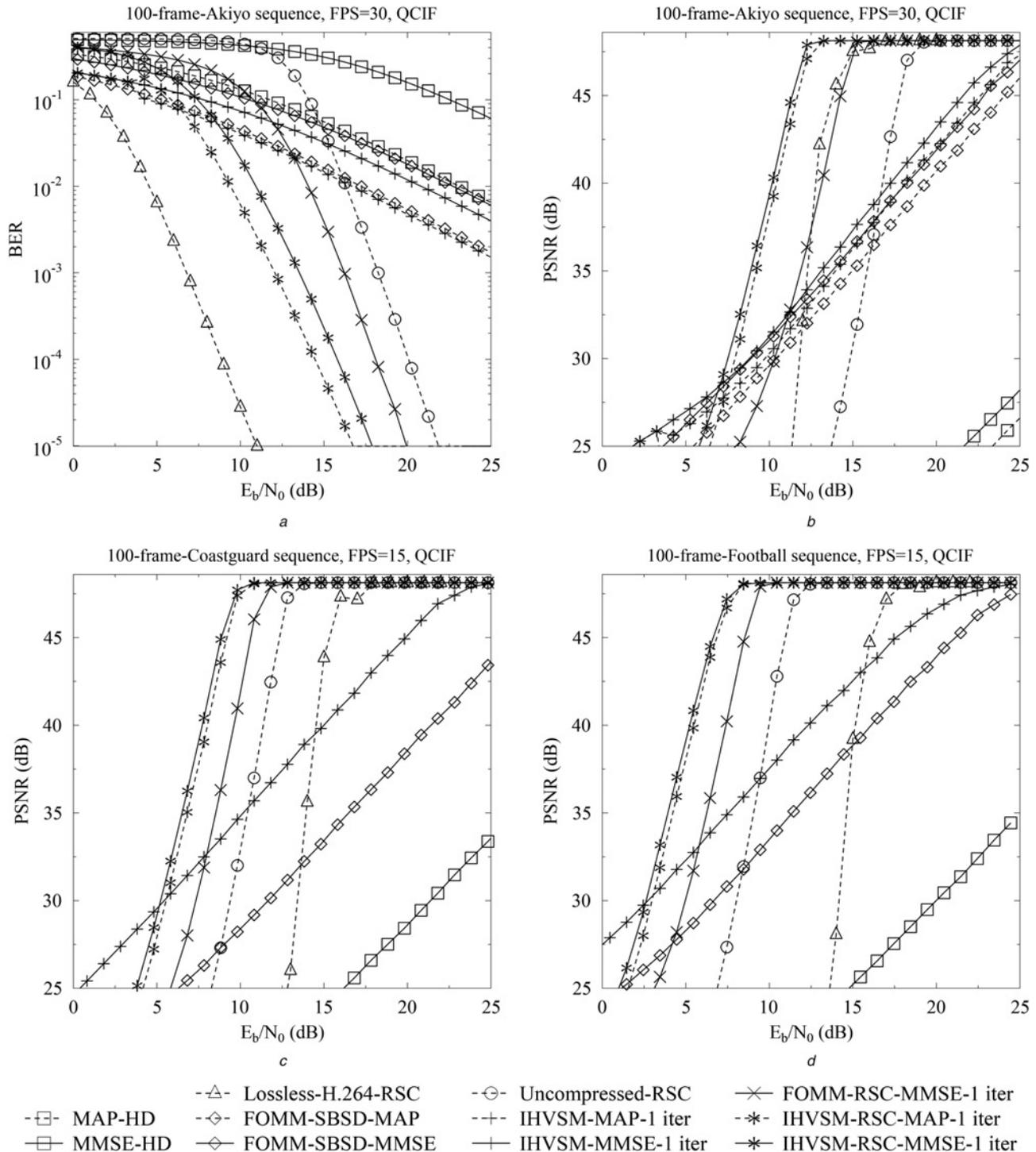


Fig. 6 BER/PSNR against E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel

a Akiyo, QCIF
 b Akiyo, QCIF
 c Coastguard, QCIF
 d Football, QCIF

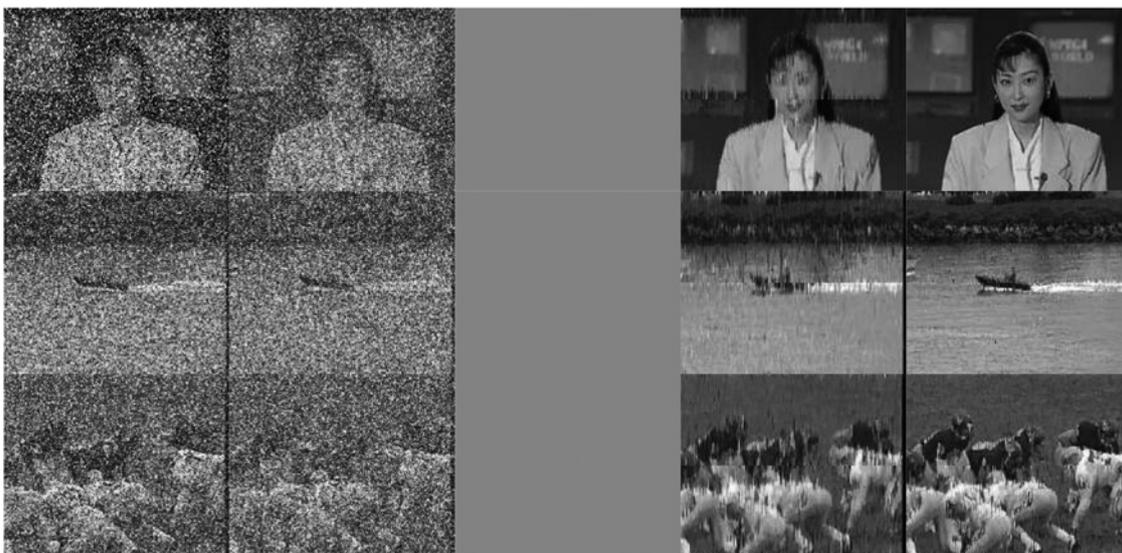


Fig. 7 Comparison of the first QCIF frames recorded at E_b/N_0 of 12.3, 3.8 and 2.5 dB for the Akiyo, Coastguard and Football sequences, respectively, decoded by MMSE-HD [5], Uncompressed-RSC, Lossless-H.264-RSC, FOMM-RSC [7] and IHVSM-RSC

MMSE-based pixel estimation is employed

Lossless-H.264-RSC arrangement for simplicity. Note that the Lossless-H.264-RSC system imposes a high complexity at the transmitter, but a low complexity at the receiver. By contrast, our system imposes low complexity at the transmitter and a high complexity at the receiver.

A brief comparison of all the benchmark schemes is shown in Table 2, where L-H.264-RSC represents the Lossless-H.264-RSC scheme. The Row Col (Trans./Rec.) compares the complexity imposed at the transmitter and receiver, respectively.

4.3 Numerical results

Firstly, the BER against E_b/N_0 performance comparison of the MAP-based and MMSE-based pixel estimation using the Akiyo-QCIF sequence is presented in Fig. 6a, whereas the corresponding Y-PSNR against E_b/N_0 results are presented in Fig. 6b. As suggested by Fig. 6a, the IHVSM-RSC scheme substantially outperforms both the FOMM-SBSD and the uncompressed-RSC schemes. Specifically, at a BER of 10^{-5} the IHVSM-RSC-MMSE system outperforms the FOMM-RSC-MMSE and the uncompressed-RSC systems by about 2 and 4.6 dB, respectively. Moreover, even though the Lossless-H.264-RSC achieves the best BER performance, the compressed bits are rather sensitive to bit errors. As suggested by Fig. 5, at a PSNR of 47.5 dB the IHVSM-RSC-MMSE system outperforms the FOMM-RSC-MMSE, the uncompressed-RSC and the Lossless-H.264-RSC systems by about 2.8, 6.6 and 2.8 dB, respectively. In other words, at E_b/N_0 of 12.1 dB the IHVSM-RSC-MMSE system outperforms the FOMM-RSC-MMSE and the Lossless-H.264-RSC systems by about 12.1 and 15 dB, in terms of the video PSNR achieved, respectively.

Observe from Fig. 6a that the MAP-based systems outperform the schemes employing the MMSE estimator in terms of the BER attained. On the other hand, Fig. 6b suggests that the MMSE-based systems outperform the schemes employing the MAP estimator in terms of the PSNR attained. We may conclude that the MAP estimator is capable of achieving a lower BER, whereas the MMSE

estimator may achieve an increased PSNR. Hence, to achieve an increased PSNR, the MMSE-based pixel estimation should be employed.

To provide further insights, we present PSNR against E_b/N_0 results in Figs. 6c and 6d for the high-dynamic Coastguard-QCIF and Football-QCIF sequences, respectively. Again, the MMSE-based estimator is capable of achieving a higher PSNR than the MAP-based estimator. We mainly present the Y-PSNR against E_b/N_0 curves in Figs. 6c and 6d using the MMSE-based estimator. As suggested by Figs. 6c and 6d, the IHVSM-RSC-MMSE substantially outperforms the FOMM-RSC-MMSE, the uncompressed-RSC and the Lossless-H.264-RSC schemes. Specifically, at a Y-PSNR of 47.5 dB the IHVSM-RSC-MMSE outperforms the FOMM-RSC-MMSE, the uncompressed-RSC and the Lossless-H.264-RSC schemes by about 1.8, 3.3 and 6.4 dB, respectively, when considering the Coastguard-QCIF sequence. Similar trends are observed for the Football-QCIF sequence. Our visual comparisons of the decoded frames for the Akiyo, Coastguard and Football sequences are shown in Fig. 7.

5 Conclusions

In this paper, we proposed iterative source decoding for video-quality enhancements, which may be combined with arbitrary FEC codecs. Furthermore, a reduced-complexity first-order Markov model-based source decoder was derived. Iterative decoding was performed by exchanging extrinsic information among the FEC codec and two source decoders, namely the HSMD and the VSMD. Our simulation results showed that the proposed system substantially improves the achievable system performance and may facilitate a transmit power reduction in excess of 2.8 dB in uncompressed video coding.

Our future work will focus on iterative decoding exchanging extrinsic information between the source decoder and channel decoder.

6 Acknowledgments

The financial support of the RC-UK under the auspices of the India-UK Advanced Technology Centre (IU-ATC) and that of the EU under the CONCERTO project as well as that of the European Research Council's Advanced Fellow Grant is gratefully acknowledged.

7 References

- 1 Shannon, C.E.: 'A mathematical theory of communication', *Bell Syst. Tech. J.*, 1948, **27**, pp. 379–423 and 623–656
- 2 Sayood, K., Borkenhagen, J.: 'Use of residual redundancy in the design of joint source/channel coders', *IEEE Trans. Commun.*, 1991, **39**, pp. 838–846
- 3 Görtz, N.: 'Joint source channel decoding using bit-reliability information and source statistics'. Int. Symp. on Information Theory, August 1998, pp. 9
- 4 Görtz, N.: 'On the iterative approximation of optimal joint source-channel decoding', *IEEE J. Sel. Areas Commun.*, 2001, **19**, pp. 1662–1670
- 5 Fingscheidt, T., Vary, P.: 'Softbit speech decoding: a new approach to error concealment', *IEEE Trans. Speech Audio Process.*, 2001, **9**, pp. 240–251
- 6 Adrat, M., Vary, R., Spittka, J.: 'Iterative source-channel decoder using extrinsic information from softbit-source decoding'. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, May 2001, vol. **4**, pp. 2653–2656
- 7 Adrat, M., Vary, P.: 'Iterative source-channel decoding: improved system design using EXIT charts', *EURASIP J. Appl. Signal Process.*, 2005, **2005**, pp. 928–941
- 8 Kliewer, J., Görtz, N., Mertins, A.: 'On iterative source-channel image decoding with Markov random field source models'. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, August 2004, vol. **4**, pp. iv-661–iv-664
- 9 Kliewer, J., Görtz, N., Mertins, A.: 'Iterative source-channel decoding with Markov random field source models', *IEEE Trans. Signal Process.*, 2006, **54**, pp. 3688–3701
- 10 Kliewer, J., Thobaben, R.: 'Iterative joint source-channel decoding of variable-length codes using residual source redundancy', *IEEE Trans. Wirel. Commun.*, 2005, **4**, pp. 919–929
- 11 Hanzo, L., Maunder, R.G., Wang, J., Yang, L.-L.: 'Near-capacity variable-length coding: regular and exit-chart aided irregular designs' (John Wiley & Sons Ltd, 2010)
- 12 Zhang, Y., Zhu, C., Yap, K.-H.: 'A joint source-channel video coding scheme based on distributed source coding', *IEEE Trans. Multimed.*, 2008, **10**, pp. 1648–1656
- 13 Nasruminallah, Hanzo, L.: 'EXIT-chart optimized short block codes for iterative joint source and channel decoding in H.264 video telephony', *IEEE Trans. Veh. Technol.*, 2009, **58**, pp. 4306–4315
- 14 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, March 2010
- 15 Fresia, M., Pérez-Cruz, F., Poor, H., Verdú, S.: 'Joint source and channel coding', *IEEE Signal Process. Mag.*, 2010, **27**, pp. 104–113
- 16 Hanzo, L., Cherriman, P., Streit, J.: 'Video compression and communications: from basics to H.261, H.263, H.264, MPEG2, MPEG4 for DVB and HSDPA-style adaptive turbo-transceivers' (John Wiley, New York, 2007)
- 17 Singh, H., Oh, J., Kweon, C., Qin, X., Shao, H.-R., Ngo, C.: 'A 60 GHz wireless network for enabling uncompressed video communication', *IEEE Commun. Mag.*, 2008, **46**, pp. 71–78
- 18 Liu, K., Ling, X., Shen, X., Mark, J.: 'Performance analysis of prioritized MAC in UWB WPAN with bursty multimedia traffic', *IEEE Trans. Veh. Technol.*, 2008, **57**, pp. 2462–2473
- 19 Gilbert, J., Doan, C., Emami, S., Shung, C.: 'A 4 Gbps uncompressed wireless HD A/V transceiver chipset', *IEEE Micro*, 2008, **28**, pp. 56–64
- 20 Hutanu, A., Paruchuri, R., Eiland, D., *et al.*: 'Uncompressed HD video for collaborative teaching – an experiment'. Int. Conf. on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2007, November 2007, pp. 253–261
- 21 Wei, S.-T., Tien, C.-W., Liu, B.-D., Yang, J.-F.: 'Adaptive truncation algorithm for Hadamard-transformed H.264/AVC lossless video coding', *IEEE Trans. Circuits Syst. Video Technol.*, 2011, **21**, pp. 538–549
- 22 Fisher, R.: '60 GHz WPAN standardization within IEEE 802.15.3c'. Int. Symp. on Signals, Systems and Electronics, 2007 (ISSSE'07), 30 July–2 August 2007, pp. 103–105
- 23 Park, C., Rappaport, T.: 'Short-range wireless communications for next-generation networks: UWB, 60 GHz millimeter-wave WPAN, and ZigBee', *IEEE Wirel. Commun.*, 2007, **14**, pp. 70–78
- 24 <http://www.wirelessHD.org>, WirelessHD Specification Overview, October 2007
- 25 Pepe, D., Zito, D.: '60 GHz transceivers for wireless HD uncompressed video communication in nano-era CMOS technology'. Proc. 15th IEEE Mediterranean Electrotechnical Conf., (MELECON), April 2010, pp. 1237–1240
- 26 Singh, H., Qin, X., Shao, H., Ngo, C., Kwon, C., Kim, S.S.: 'Support of uncompressed video streaming over 60 GHz wireless networks'. Proc. 5th IEEE Consumer Communications and Networking Conf., (CCNC 2008), January 2008, pp. 243–248
- 27 Singh, H., Niu, H., Qin, X., *et al.*: 'Supporting uncompressed HD video streaming without retransmissions over 60 GHz wireless networks'. IEEE Wireless Communications and Networking Conf., (WCNC 2008), 31 March–3 April 2008, pp. 1939–1944
- 28 Shao, H.-R., Ngo, C., Singh, H., *et al.*: 'Adaptive multi-beam transmission of uncompressed video over 60 GHz wireless systems', *Future Gener. Commun. Netw.*, (FGCN 2007), 2007, **1**, pp. 430–435
- 29 Hong, S.-E., Lee, W.Y.: 'Flexible unequal error protection scheme for uncompressed video transmission over 60 GHz multi-Gigabit wireless system'. Proc. 20th Int. Conf. on Computer Communications and Networks (ICCCN), 31 July–4 August 2011, pp. 1–6
- 30 Girod, B., Aaron, A., Rane, S., Rebollo-Monedero, D.: 'Distributed video coding', *Proc. IEEE*, 2005, **93**, pp. 71–83
- 31 Hanzo, L., Liew, T., Yeap, B., Tee, R.: 'Turbo coding, turbo equalisation and space-time coding' (John Wiley, New York, 2011)
- 32 Berrou, C., Glavieux, A., Thitimajshima, P.: 'Near shannon limit error-correcting coding and decoding: Turbo codes'. Proc. Int. Conf. on Communications, Geneva, Switzerland, May 1993, pp. 1064–1070
- 33 Bahl, L.R., Cocke, J., Jelinek, F., Raviv, J.: 'Optimal decoding of linear codes for minimising symbol error rate', *IEEE Trans. Inf. Theory*, 1974, **20**, pp. 284–287
- 34 Nasruminallah, Hanzo, L.: 'Near-capacity H.264 multimedia communications using iterative joint source-channel decoding', *IEEE Commun. Surv. Tutorials*, Second Quarter 2012, **14**, pp. 538–564

8 Appendix

Let us initially follow the procedure of the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [33] algorithm-based MAP decoder for deriving the APP $p(x_i|y_0^t)$. The APP $p(x_i|y_0^t)$ of the m -bit pixel x_i , $x_i \in X_m$ conditioned on the soft Markov process y_0, \dots, y_t may be formulated as

$$p(x_i|y_0^t) = \frac{p(x_i \wedge y_0^t)}{p(y_0^t)} \quad (13)$$

where $p(x_i \wedge y_0^t)$ indicates the probability that the soft Markov process y_0^t is received and the pixel value x_i is transmitted at time instant i . Furthermore, the joint probability $p(x_i \wedge y_0^t)$ may be further formulated as follows:

$$\begin{aligned} p(x_i \wedge y_0^t) &= p(x_i \wedge y_i \wedge y_0^{i-1} \wedge y_{i+1}^t) \\ &= p(y_{i+1}^t | x_i \wedge y_i \wedge y_0^{i-1}) p(y_i | y_0^{i-1} \wedge x_i) p(y_0^{i-1} \wedge x_i) \\ &= p(y_{i+1}^t | x_i) p(y_i | x_i) p(y_0^{i-1} \wedge x_i) \\ &= \beta_i(x_i) \chi_i(x_i) \alpha_i(x_i) \end{aligned} \quad (14)$$

In (14), the probability functions α , β and χ are defined as follows:

$$\begin{aligned} \alpha_i(x_i) &= p(y_0^{i-1} \wedge x_i) \\ \beta_i(x_i) &= p(y_{i+1}^t | x_i) \\ \chi_i(x_i) &= p(y_i | x_i) \end{aligned} \quad (15)$$

The probability function α , β and χ in (15) are illustrated graphically in the trellis of Fig. 3. Specifically, the function α indicates the probability that the soft pixels y_0^{j-1} are received and the pixel x_i is transmitted at time instant i , whereas the function β indicates the probability that the soft pixels y_{i+1}^j are received conditioned on the pixel x_i being transmitted at time instant i . In (15), the symbol-based channel information $\chi_i(x_i) = p(y_i/x_i)$ gives the a-priori probability of the pixel x_i , which may be calculated from the bit-based channel information $p[y_i(k)/x_i(k)]$ as follows:

$$\begin{aligned} \chi_i(x_i) &= p(y_i|x_i) \\ &= \prod_{k=0}^{m-1} p[y_i(k)|x_i(k)] \\ &= C_{\chi_i} \exp \sum_{k=0}^{m-1} \frac{x_i(k)}{2} L[y_i(k)|x_i(k)] \end{aligned} \quad (16)$$

where C_{χ_i} is the normalisation factor, which solely depends on the soft information y_i . Furthermore, similar to the forward recursion calculation in BCJR, the function $\alpha_i(x_i)$ in (15) may be formulated as

$$\begin{aligned} \alpha_i(x_i) &= p(y_0^{j-1} \wedge x_i) \\ &= \sum_{x_{i-1} \in X_m} p(y_{i-1} \wedge y_0^{j-2} \wedge x_i \wedge x_{i-1}) \\ &= \sum_{x_{i-1} \in X_m} p(y_{i-1}|x_{i-1}) p(x_i|x_{i-1}) p(y_0^{j-2} \wedge x_{i-1}) \\ &= \sum_{x_{i-1} \in X_m} \chi_{i-1}(x_{i-1}) p(x_i|x_{i-1}) \alpha_{i-1}(x_{i-1}) \end{aligned} \quad (17)$$

Similarly, the backward recursion calculation of the function

$\beta_i(x_i)$ in (15) may be formulated as follows:

$$\begin{aligned} \beta_i(x_i) &= p(y_{i+1}^j|x_i) \\ &= \sum_{x_{i+1} \in X_m} p(y_{i+2}^j \wedge y_{i+1} \wedge x_{i+1}|x_i) \\ &= \sum_{x_{i+1} \in X_m} p(y_{i+2}^j|x_{i+1}) p(y_{i+1}|x_{i+1}) p(x_{i+1}|x_i) \\ &= \sum_{x_{i+1} \in X_m} \beta_{i+1}(x_{i+1}) \chi_{i+1}(x_{i+1}) p(x_{i+1}|x_i) \end{aligned} \quad (18)$$

Based on the above rules, the functions α , β and χ of (14) may be calculated. On the other hand, the bit-based APP $p[x_i(k)|y_0^j]$ may be determined from the symbol-based APP $p(x_i|y_0^j)$ as [7]

$$p[x_i(k) = u|y_0^j] = \sum_{\substack{x_i(k)=u \\ x_i \in X_m}} p(x_i|y_0^j) \quad (19)$$

where u represents the possible binary values 0 or 1 of $x_i(k)$. Then, the bit-based APP of the bit $x_i(k)$ conditioned on the soft Markov process y_0^j may be formulated as

$$\begin{aligned} L[x_i(k)|y_0^j] &= \ln \frac{p[x_i(k) = 0|y_0^j]}{p[x_i(k) = 1|y_0^j]} \\ &= \ln \frac{\sum_{x_i(k)=0} p(x_i|y_0^j)}{\sum_{x_i(k)=1} p(x_i|y_0^j)} \\ &= \ln \frac{\sum_{x_i(k)=0} \beta_i(x_i) \chi_i(x_i) \alpha_i(x_i)}{\sum_{x_i(k)=1} \beta_i(x_i) \chi_i(x_i) \alpha_i(x_i)} \end{aligned} \quad (20)$$