www.ietdl.org

Published in IET Communications
Received on 6th August 2010
Revised on 5th January 2011
doi: 10.1049/iet-com.2010.0717

ISSN 1751-8628

Robust decoder-based error control strategy for recovery of H.264/AVC video content

R.A. Farrugia C.J. Debono

Department of Communications and Computer Engineering, University of Malta, Msida MSD 2080, Malta
E-mail: cjdebo@eng.um.edu.mt

Abstract: Real-time wireless conversational and broadcasting multimedia applications offer particular transmission challenges as reliable content delivery cannot be guaranteed. The undelivered and erroneous content causes significant degradation in quality of experience. The H.264/AVC standard includes several error resilient tools to mitigate this effect on video quality. However, the methods implemented by the standard are based on a packet-loss scenario, where corrupted slices are dropped and the lost information concealed. Partially damaged slices still contain valuable information that can be used to enhance the quality of the recovered video. This study presents a novel error recovery solution that relies on a joint source-channel decoder to recover only feasible slices. A major advantage of this decoder-based strategy is that it grants additional robustness while keeping the same transmission data rate. Simulation results show that the proposed approach manages to completely recover 30.79% of the corrupted slices. This provides frame-by-frame peak signal-to-noise ratio (PSNR) gains of up to 18.1 dB, a result which, to the knowledge of the authors, is superior to all other joint source-channel decoding methods found in literature. Furthermore, this error resilient strategy can be combined with other error resilient tools adopted by the standard to enhance their performance.

1 Introduction

Despite many advances in multimedia communication systems in terms of services, usage, bandwidth, and capacity, the quality of experience achieved by the current and the next generation networks is not sufficient to guarantee error-free delivery of real-time multimedia content over wireless channels [1]. This is particularly problematic in conversational and multicast/broadcast applications because of the delay constraints and the unavailability of a feedback channel, respectively [2].

The H.264/AVC video coding standard [3] includes several error resilient tools to alleviate the effect of transmission errors on the perceptual quality of the recovered video content. However, these methods assume a packet-loss scenario, where the receiver discards and conceals all the video information contained within a corrupted slice. This implies that the error resilient methods adopted by the standard operate at a lower bound since not all the information contained within a corrupted packet is un-utilisable [4]. Driven by the observation that most video applications prefer damaged packets to lost packets, novel transport layer protocols such as UDPLite [5] were developed. This enables the transport layer to flag damaged packets to the application layer while still forwarding them to the application for further processing.

Several extensions to the standard and various error resilient strategies have been proposed in literature to enhance the performance of the standard H.264/AVC codec for wireless transmission. Inspired by the previous video coding standards, the authors in [6] have proposed a set of syntax and semantic violation rules that can be used to detect transmission errors at macroblock (MB) level. However, this method only manages to detect 57% of the corrupted MBs resulting in a number of visually impaired regions that significantly degrade the quality of the reconstructed video sequences.

The inherent redundancies present within spatio-temporal neighbouring MBs can be used to detect these residual artefacts. Procedures based on dissimilarity metrics and heuristic thresholds were investigated in [7–13]. However, all these methods have limited applications in practice since the optimal thresholds vary from sequence to sequence. An iterative solution presented in [14] attained a substantial gain in quality at the expense of significantly increasing the complexity of the decoder, making it unsuitable for real-time mobile applications. Machine learning algorithms have been recently applied in [15–17] to detect visually distorted MBs at pixel level. Although this approach manages to detect most of the residual artefacts at image level, at a moderate increase in complexity, they do not manage to recover the original quality of the transmitted video content.

Data hiding techniques have been widely investigated as a means of error resilient coding [18–23]. However, the embedded information contributes to the reduction of the quality of the transmitted video content even when the transmission occurs over an error-free channel. Scalable video coding [24], multiple description coding [25, 26], distributed scheduling [27] and distributed resource management schemes [28] are alternative approaches that

can be used to provide error resilience. However, the increased robustness is achieved by increasing the data rate required to deliver the same quality criterion in the absence of transmission errors.

Error-control strategies offer a more promising alternative to protect the transmitted bitstreams. The authors in [29] have replaced the variable length code (VLC) tables of MPEG-4 with variable length error control (VLEC) codes. However, this solution reduces the compression efficiency of the encoder since VLEC codewords have a longer average codeword length. The redundancy in the compressed image and video was analysed in [30], where it was concluded that significant gain in performance can be achieved when additional video data properties are taken into consideration while decoding. Based on this observation the authors in [31, 32] have implemented a modified Viterbi decoding algorithm to recover feasible video sequences [33]. Sequential decoding methods were adopted for H.264/AVC encoded sequences in [31, 34] whose performance in terms of quality is similar to the results presented in [33]. However, sequential decoding algorithms do not always converge and introduce variable decoding delays, making them unsuitable in real-time applications [35].

This paper presents an error-control method which employs a list decoder strategy to recover the most-likelihood feasible bitstream. This technique adopts soft information and source constraints to minimise complexity and aid convergence. Since in H.264/AVC the VLC tables are switched according to the previous syntax elements, a contextual module was adopted whose function is to keep track of the next VLC table to be loaded for each bitstream stored in the list. The proposed approach is fundamentally different from the methods presented in [31, 33, 34] in the way the lists are managed. This is mainly attributed to the fact that the proposed approach always keeps the most probable correct partial bitstreams in the list and therefore it

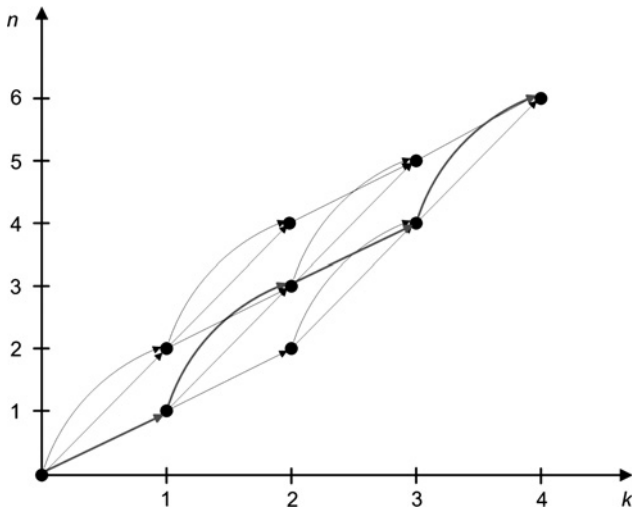

Fig. 2 *Trellis representation according to [30]*

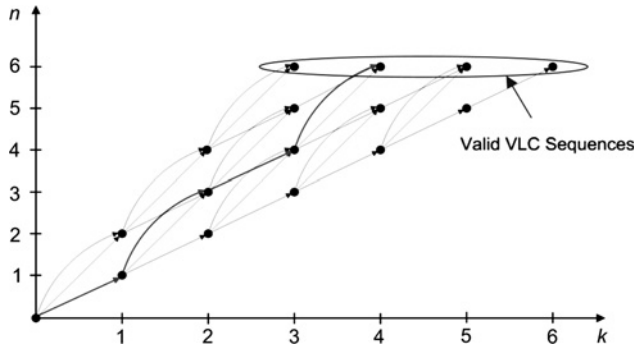

Fig. 3 *Trellis representation used in this work*

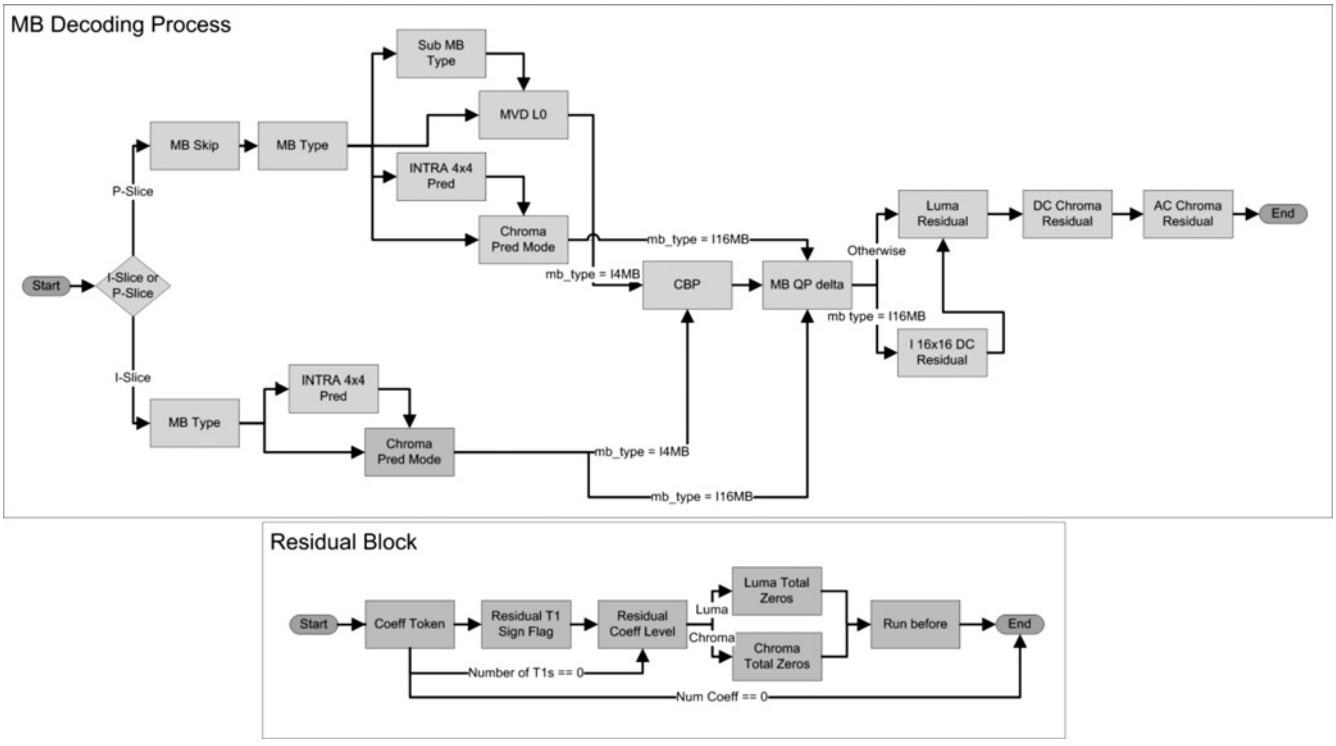

Fig. 1 *MB decoding process used by the H.264/AVC standard*

maximises the possibility of recovering the transmitted slice. In fact the proposed approach manages to completely recover, on average, 30.79% of the corrupted slices relative to the 6.28% recovered by the method proposed in [33]. This results in a significant gain in quality relative to the standard and other approaches considered in this paper. Furthermore, the proposed solution can be applied in conjunction with other error resilient tools offered by the standard to further enhance their performance. Results show that through this error recovery strategy the decoder achieves a significant gain in quality, with peak signal-to-noise ratios (PSNR) of up to 18.1 dB being observed on a frame-by-frame basis over the standard.

This paper is organised as follows. After a brief introduction to the notation and trellis representation of the VLC symbols, the basic components used to solve the VLC sequence estimation problems are presented in Section 2 followed by the implementation details of the proposed list-decoding VLC approach in Section 3. Simulation results are then given in the following section whereas final comments and conclusions are delivered in Section 5.

2 Error control of VLC symbols

2.1 Residual source redundancy

The aim of current video compression standards is to eliminate the redundancy in the video signal to maximise compression. However, practical schemes do not remove completely this redundancy because of complexity constraints, and thus residual source redundancy remains after compression. In [30] it was concluded that traditional error-control mechanisms can be designed to exploit this redundancy to enhance the error-control capabilities of the system without affecting the transmission bit rate.

Traditional channel coding techniques introduce structured redundant information which can be used by the decoder for error detection and correction. Instead, error-control schemes for image/video applications can exploit the residual source redundancies available after compression to ensure that only valid image blocks or slices are derived. For a bitstream to conform to a valid H.264/AVC slice or frame, the following set of constraints must be satisfied:

1. Create an empty list state $S_{0,0}$, containing an empty bitstream, set the metric $\Lambda(S_{0,0})$ equal to zero, and store it in list L_{k-1}. Set $k = 0$ and $i = 0$.
2. Initialise a second list F as an empty list and increment the symbol time k by one.
3. Each state $S_{i,k-1}$ in the list L_{k-1} contains a metric $\Lambda(S_{i,k-1})$ and a bitstream $y(S_{i,k-1})_0^{n_i}$ of length n_i bits. For each state in list L_{k-1}, execute the following procedure:
   i. For each VLC codeword c_j of length λ_j bits in the VLC table do the following:
      a. Derive the new partial bitstream $y(S_{p,k})_0^{n+\lambda_j} = \left[y(S_{i,k-1})_0^{n_i}, c_j \right]$ which corresponds to the concatenation between the partial bitstream stored in $S_{i,k-1}$ of length n_i bits and the codeword c_j. The resulting bitstream is stored in state $S_{p,k}$, where p is the state index in list L_k.
      b. Compute the metric $\Lambda(S_{p,k}) = \Lambda(S_{i,k-1}) + \left\| r_0^{n_i+\lambda_j}, y(S_{p,k})_0^{n_i+\lambda_j} \right\|$ where $\|\bullet\|$ denotes the Euclidean distance. Store the resulting metric in state $S_{p,k}$.
      c. Store the states that contain complete feasible bitstreams in list F.
      d. Store the states that contain incomplete feasible bitstreams in list L_k.
      e. Prune the states that represent non-feasible bitstreams.
   ii. Reset list L_{k-1} as an empty list.
   iii. Sort the states in list L_k in ascending order according to the metric, and move the first M states into list L_{k-1} for the next iteration. Prune all the remaining states accordingly.
4. If L_k is empty and $k \geq 1$ terminate the process. Otherwise, go to step 3.

Fig. 4 *List decoding algorithm*

doi: 10.1049/iet-com.2010.0717

Table 1 Source codebook

Symbol	Codeword
a	1
b	01
c	001

The length of the sequence of VLC codewords is equal to N bits.

1. The number of MBs decoded must be equal to the number of MBs, N_{MB}, in the slice. All these MBs must be completely decoded.
2. The decoded symbols and derived parameters are within the allowed ranges and obey the syntax rules specified by the H.264/AVC standard.

The bitstreams that conform to all these constraints are considered to be complete feasible H.264/AVC sequences. On the other hand, sequences of lengths strictly smaller than N bits and whose number of decoded MBs is smaller than N_{MB} are considered as incomplete feasible sequences. The remaining sequences are not feasible and can therefore be pruned by the error-control method.

2.2 Source constraints

Another source of redundancy present after compression is the source constraints available in any video coding standard. Two different entropy coding methods are supported by H.264/AVC, namely: context-adaptive VLC (CAVLC) and context-adaptive binary arithmetic coding. The baseline profile, which is generally adopted for real-time wireless applications, adopts the CAVLC entropy coding method [3, 31]. For this reason, CAVLC entropy coding is considered in this work. CAVLC encodes the quantised coefficients using VLC tables which are switched depending on the previous syntax elements and the VLC table adopted in the previous decoding step. This method allows higher coding efficiency when compared to other coding standards that adopt single VLC tables [36].

The H.264/AVC decoding process of an MB is illustrated in Fig. 1, where the decoding of each MB starts from the initial state and proceeds step by step until it reaches the termination state. The decoder keeps track of the processing order of the slice it is decoding and predicts the next VLC table to be considered in the following step. The residual block shown in Fig. 1 is adopted by the Luma Residual, I16 × 16 DC Residual, DC Chroma Residual and AC Chroma Residual to derive the residual coefficients. More information on these coefficients can be found in [37].

To apply error-control methods to H.264/AVC bitstreams, it is necessary that each state identifies which VLC table must be loaded next. The 'contextual module', whose function is to store all the relevant information for each state in the trellis representation, was thus developed. This information is then used at the decoding time to predict which VLC table needs to be loaded for the next processing stage.

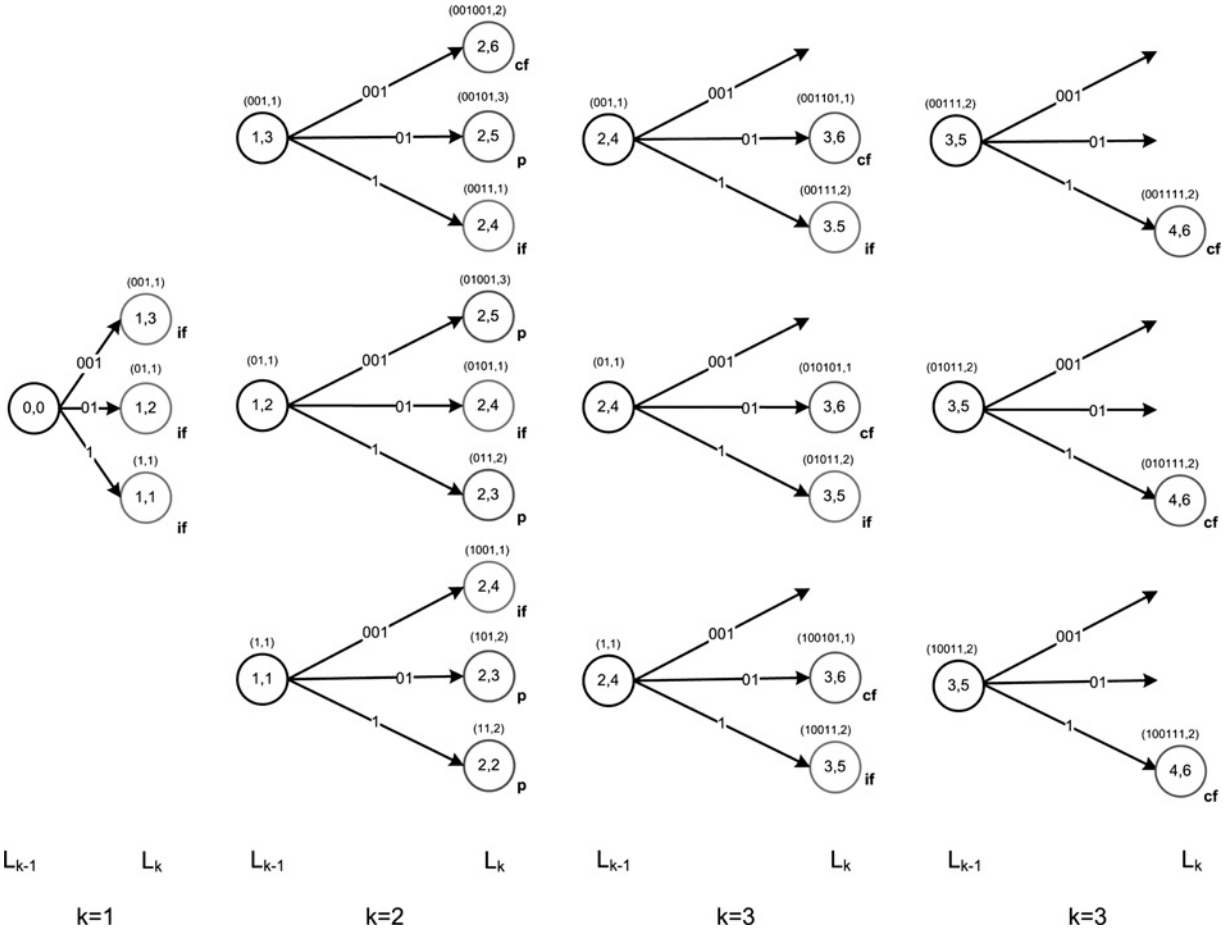

Fig. 5 *Illustration of the list decoding method proposed in this paper*

Table 2 Potentially valid codewords using the list decoder approach

Sequence	Hamming distance
cc	2
acb	1
bbb	1
cab	1
acaa	2
bbaa	2
caaa	2

Table 3 Potentially valid codewords using the Viterbi approach adopted in [26]

Sequence	Hamming distance
cc	2
abc	3
bbb	1
aca	3
bbaa	2
aaac	4
aabb	5
aaaab	5
aabaa	3
aaaaaa	4

2.3 Trellis representation of VLC symbols

In order to provide a recovery strategy for the original bitstreams, the VLC sequence has to be represented by a trellis representation. The error-control strategy presented in this paper is based on the trellis representation published in [38], which was modified to handle H.264/AVC encoded bitstreams. The concept of this trellis representation is introduced by a simple example as in [38]; Consider a VLC codebook with an alphabet size $Q = 3$ with codewords $C = (1, 01, 00)$. Further, consider that a sequence of $K = 4$ source symbols $\boldsymbol{u} = (0, 2, 0, 1)$ is to be transmitted. This sequence is mapped onto a sequence of variable length codewords $\boldsymbol{C} = (1, 00, 1, 01)$, where the length of the transmitted bitstream $N = 6$. The VLC sequences with $K = 4$ symbols and $N = 6$ bits can then be represented in a trellis diagram, as shown in Fig. 2.

However, most of the time, the number of transmitted symbols $\hat{K}$ is not known by the decoder. Thus, a modified trellis representation, illustrated in Fig. 3, is considered in this work. Here, the paths arriving at a node $S_{n,k}$ correspond to the sub-sequences of k VLC codewords of length n bits. Since the decoder has no knowledge on the number of symbols contained within the sequence, all the sequences of length N bits are considered to be candidate complete sequences. Given this trellis representation, VLC sequence estimation techniques that derive the most-likelihood sequence of VLC symbols based on the minimum distance decoding criterion can be applied.

3 List decoder strategy

The proposed list decoding method employs three lists to derive the most likelihood sequence of VLC codewords: L_{k-1}, which contains the incomplete feasible sequences of $k-1$ symbols; L_k, which stores the incomplete feasible sequences of k symbols; and F, stores the complete feasible sequences. On receiving a sequence of modulated symbols $\boldsymbol{r}$, the list-decoding strategy adopted is shown in Fig. 4, assuming that the VLC table to be loaded is known a priori.

When the algorithm terminates the list decoder process, it recovers the H.264/AVC bitstream with the smallest metric among all the sequences contained within the list of feasible sequences F. As an illustration of this algorithm, consider that we have the source codebook shown in Table 1 and that we want to transmit the sequence cab (bitstream 001101) over a noisy channel. Assume that the third bit is corrupted by a transmission error and thus the receiver receives the bitstream 000101 which clearly generates a syntax error. Furthermore, consider that the list decoder having size $M = 3$ adopted is shown at the decoder.

As shown in Fig. 5, the list decoder strategy starts with an empty state $S_{0,0}$. Using the source table, the list decoder generates three states, where each state represents a partial sequence of 1 symbol, which in this example stores the hamming distance to represent the reliability of each state [$\mathrm{HD}(S_{1,1}) = \mathrm{HD}(0, \quad 1) = 1$, $\mathrm{HD}(S_{1,2}) = \mathrm{HD}(00, \quad 01) = 1$, $\mathrm{HD}(S_{1,3}) = \mathrm{HD}(000, \quad 001) = 1$]. The list decoder chooses the three states with the smallest hamming distance and

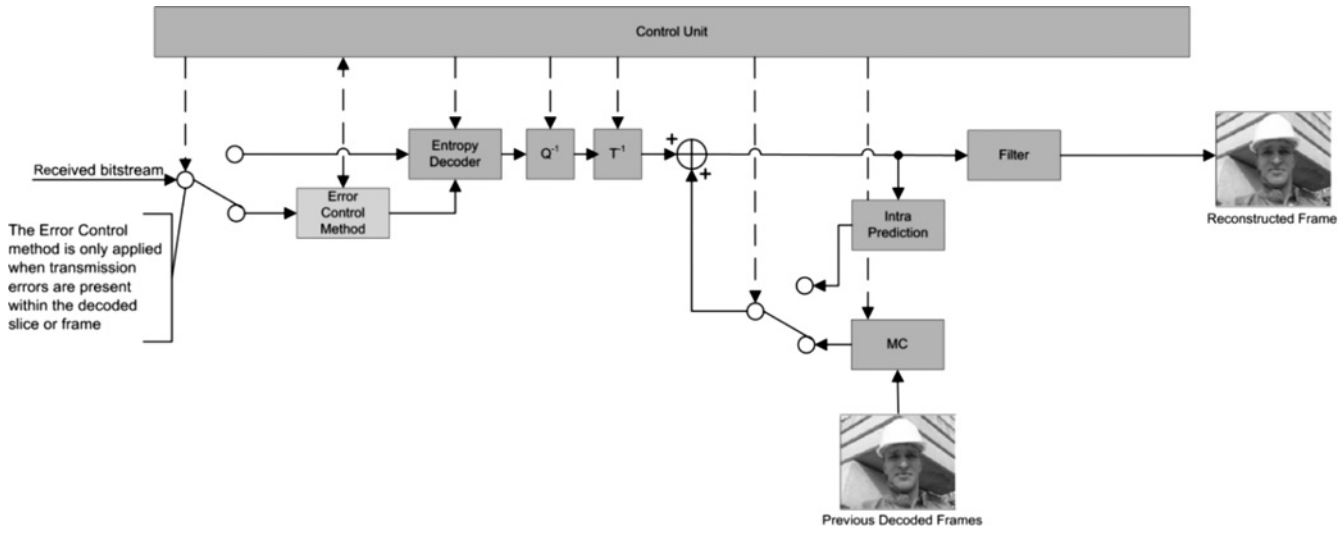

Fig. 6 *Modified H.264/AVC decoding strategy using error control as a post process*

doi: 10.1049/iet-com.2010.0717

Table 4 Reduced slice error rate (%) of the Foreman video sequence

BER	$M = 1$	$M = 5$	$M = 10$	$M = 100$
1.00E − 006	7.87	20.83	20.83	20.83
2.00E − 006	3.56	29.72	29.72	29.72
3.00E − 006	6.51	32.44	33.00	33.00
5.00E − 006	9.28	37.10	38.99	38.99
1.00E − 005	11.24	36.75	38.00	38.55
2.00E − 005	9.42	36.59	38.12	38.49
3.00E − 005	10.41	33.09	35.31	35.70
5.00E − 005	12.60	32.02	34.78	35.36
1.00E − 004	9.84	27.61	29.13	31.97
2.00E − 004	7.89	21.75	23.58	25.54
average	8.86	30.79	32.15	32.81

stores them in list L_1. In the following iteration each state in list L_1 generates three states, one for each symbol contained within the source codebook. Each generated state will inherit the partial bitstream and concatenates it with a codeword considered by the state. For example, state $S_{1,3}$, which is stored in list L_1, generates three symbols $S_{2,6}$, $S_{2,5}$ and $S_{2,4}$ and inherits the bitstream 001. Thus, the bitstream stored in $S_{2,6}$ is 001001 which has a hamming distance of 2. Similarly, state $S_{2,5}$ has a bitstream 00101 which has a hamming distance of 3 and $S_{2,4}$ has a bitstream 0011 which has a hamming distance of 1. As shown at instant $k = 2$ in Fig. 5, the states marked incomplete feasible (if) have the smallest hamming distance and are stored in list L_2 for the next iteration. The state marked complete feasible (cf) represents potentially feasible bitstreams and is stored in list F whereas the states marked as prune (p) are pruned from the list. This method proceeds until list L_k is empty at which point list F contains the most probable feasible bitstreams of length N bits.

As shown in Table 2, the sequence cab is present in the list F generated by the list decoder. On the other hand, if the Viterbi approach adopted in [32, 33] is performed, this sequence is lost at the expense of sequences having a higher hamming distance from the received bitstream (Table 3). This is mainly because the Viterbi approach

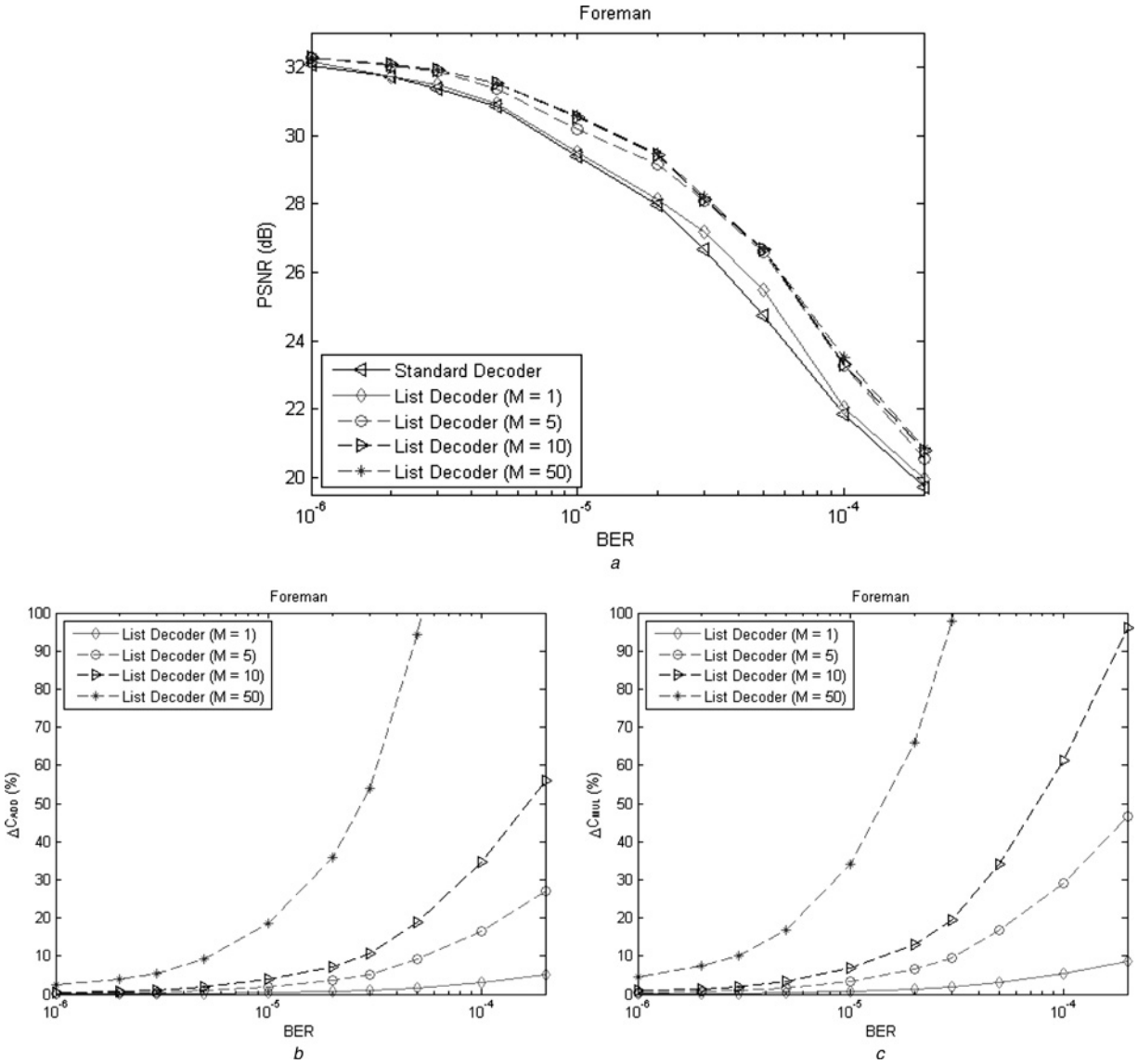

Fig. 7 *Effect on the 'Foreman' sequence by the value of M*

a Performance of the list decoder in terms of PSNR for different list size M
b Percentage increase in the number of additions computed ΔC_{Add}
c Percentage increase in the number of multiplications computed ΔC_{Mul}

selects one partial sequence of k symbols and n bits, thus prematurely pruning a number of potentially valid sequences. Furthermore, as will be shown in the following section, the list decoder method can be even more powerful when employing soft information and source constraints which increase the probability of recovering the original transmitted bitstream.

The proposed error-control module was integrated within the standard H.264/AVC decoder, as illustrated in Fig. 6. The corrupted slices are detected by the transport layer of the network protocol stack which flags corrupted slices in the Network Abstraction Layer Unit (NALU) header [4]. This flag is then used by the proposed solution to ensure that the error-control mechanism is only applied to corrupted slices. Thus, no additional delay is introduced in the system when the slices are not corrupted. The error-control module cannot recover all corrupted slices and therefore error concealment methods, which are adopted by the standard decoder, are used to conceal these unrecovered slices. Therefore the higher the error correction capabilities of the system, the lower the number of slices to be concealed, and thus the quality of the recovered video content is expected to improve.

The corrupted slices are processed by the error-control module that recovers the most-likelihood slice using the list-decoding strategy. The list decoder benefits from all the residual source redundancy and utilises the notion of survivor sequence. The error-control module further informs the control unit whether it has managed to recover the original slice. This can be done by re-computing the transport layer checksum, this time using the slice derived by the error-control mechanism. Uncorrupted and recovered slices are processed by the H.264/AVC decoder whereas the unrecovered slices are simply discarded and concealed.

4 Simulation results

The list decoder was integrated within the JM software model [39]. In order to test the system, the JM software was modified to allow the decoder to decode any partially damaged

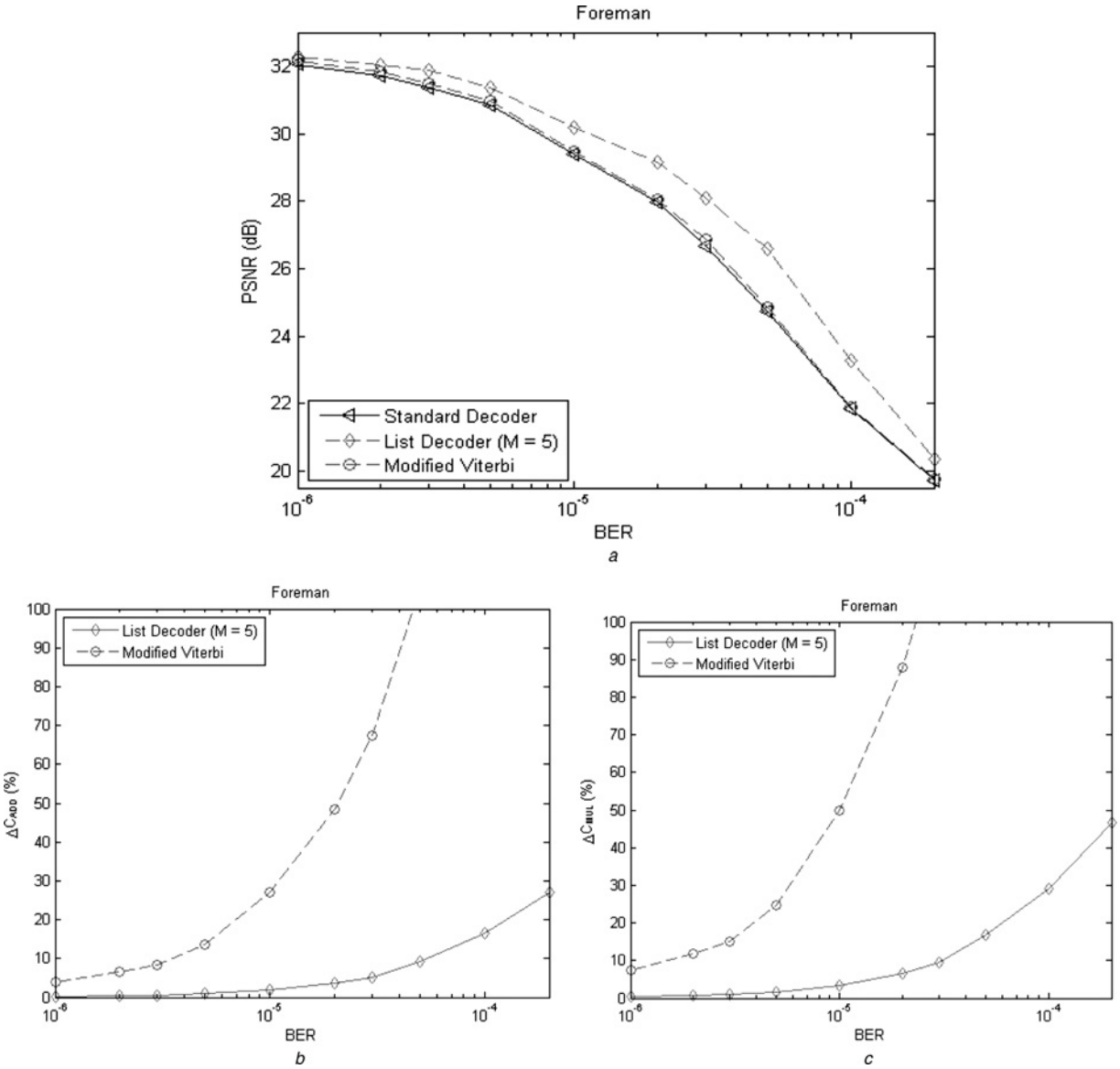

Fig. 8 *Modified Viterbi and list decoder enhance the error resilience capabilities of the standard*

a Performance of the list decoder and modified Viterbi approach
b Percentage increase in the number of additions computed ΔC_{Add}
c Percentage increase in the number of multiplications computed ΔC_{Mul}

doi: 10.1049/iet-com.2010.0717

bitstreams according to [6]. A group of pictures value of 1 was selected to obtain an encoding sequence of IPPP. The codec was configured to apply the baseline profile where the CAVLC was used as the entropy encoder, to ensure low delay. The raw video sequences used in this work were encoded at quarter common intermediate format (QCIF) resolution at 15 frames per second at a data rate of 64 kbps. The encoder only adopts slice structuring with a fixed number of 100 bytes in each slice. Each slice is encapsulated within real-time transport protocol (RTP)/user datagram protocol (UDP)/internet protocol (IP) packets using the single NALU packet mode. These packets were modulated using binary phase shift keying and transmitted over an additive white Gaussian noise channel. To ensure convergence, 34 different noise patterns were considered for each bit error rate (BER). Unless otherwise specified, no other error resilience tools were considered to be adopted by the standard H.264/AVC encoder.

The complexity of the list decoder proposed in this paper is controlled by the list size M. Table 4 and Fig. 7 show how the recovered 'Foreman' sequence is affected by the value of M. It is evident that the performance of the list decoder improves with increasing list size M. However, whereas significant gain in quality is achieved with $M > 1$, it does not continue to improve significantly after $M = 5$. Furthermore, as it can be seen in Figs. 7b and *c*, the complexity of the list decoder increases exponentially with increasing M and therefore it is important to keep the list size as small as possible. Thus, for the remaining simulations a list size of $M = 5$ is adopted giving an average error correction rate of 30.79%.

The performance of the list decoding strategy was compared to that of the Viterbi algorithm presented in [38]. Fig. 8 confirms that both the modified Viterbi and list decoder enhance the error resilience capabilities of the standard. However, for the Foreman sequence, the Viterbi decoder only manages to reduce the slice error rate by 6.28%, which corresponds to a marginal PSNR gain of around 0.1 dB. The list decoder method outperforms the Viterbi approach, where it achieves average PSNR gains of around 1.2 dB. The gain in performance is even more impressive when considering the complexity analysis (Figs. 8*b* and *c*), from which it is deduced that the list decoder achieves the substantial gain in quality by using 14 times less arithmetic operations when compared to the modified Viterbi algorithm.

In order to test the generalisation and flexibility of the list decoder, a set of video sequences were considered. This set included two standard video sequences 'Mother and Daughter' and 'Carphone', and two clips from the movies 'Lord of the Rings (Two Towers)' and 'Beautiful Mind'. The performance of the standard decoder, modified Viterbi decoder and list decoder are shown in Fig. 9.

These results confirm that both error-control mechanisms considered in this paper manage to improve the

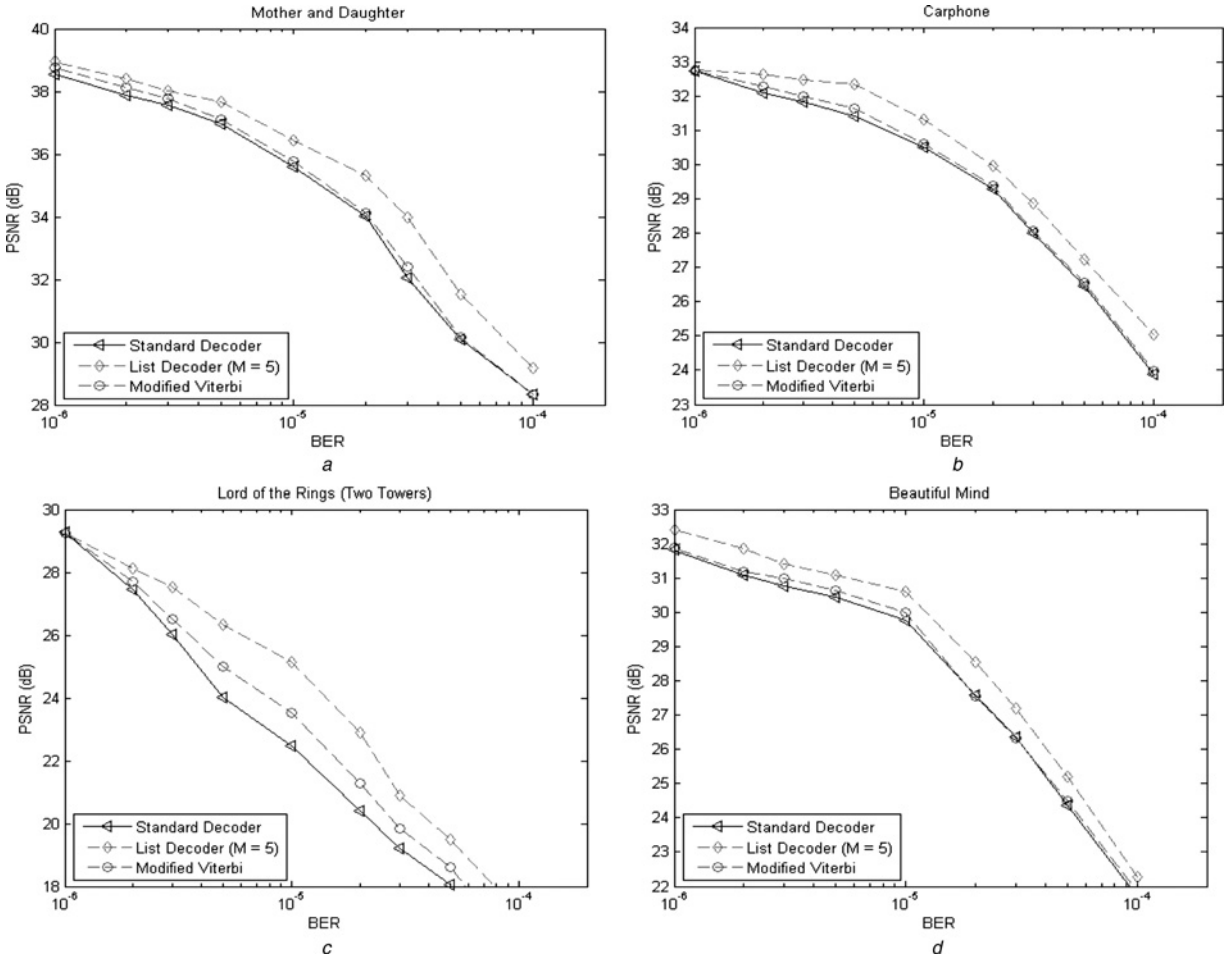

Fig. 9 *Performance of the error-control methods*

a Mother and Daughter
b Carphone
c Lord of the Rings (Two Towers)
d Beautiful Mind

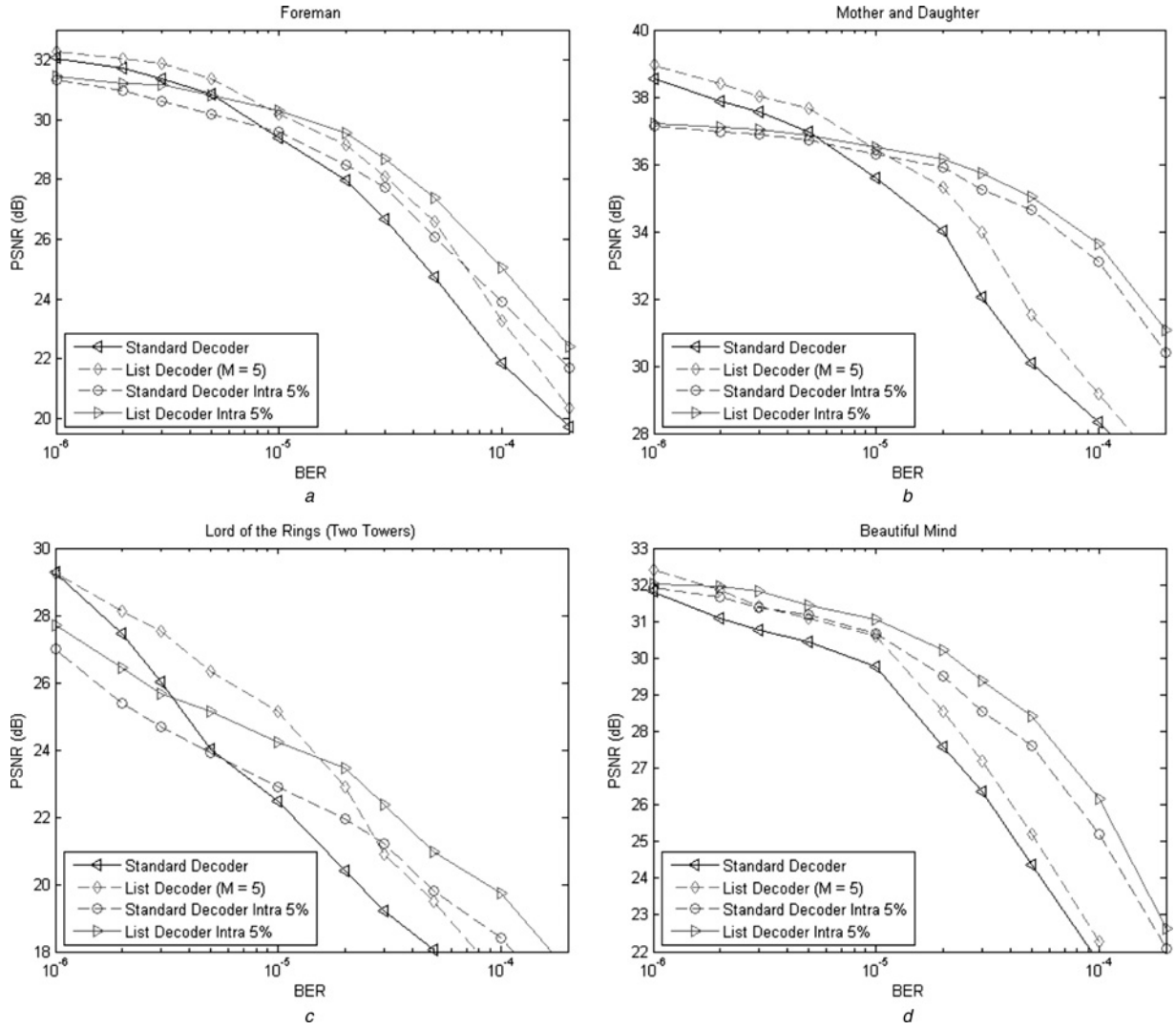

Fig. 10 *Performance of the error-control methods using Intra Refresh 5%*

a Foreman
b Mother and Daughter
c Lord of the Rings (Two Towers)
d Beautiful Mind

performance of the standard decoder. For this set of video sequences, the Viterbi decoder method provides a marginal gain of 0.1–0.2 dB in PSNR whereas the list decoder provides a gain of 1–2 dB. This confirms that the proposed strategy is the best solution since it outperforms the other methods in all the considered video sequences. The result is mainly attributed to the fact that the modified Viterbi decoder only selects one partial sequence of length n bits at each symbol time k and thus a number of sequences are pruned prematurely. On the other hand, the proposed method selects the M most reliable incomplete VLC sequences at each symbol time irrespective of their length.

Another major advantage of this method is that the list decoder can be applied in conjunction with other error resilient tools such as Intra Refresh and Flexible Macroblock Ordering (FMO). As it can be seen from Figs. 10 and 11, the performance of the list decoder with no other error resilient tools is most of the time comparable with these standard error resilient tools. However, opposed to these methods, this is achieved at no additional cost in bandwidth requirements. Furthermore, adopting the list decoder in conjunction with both Intra Refresh and dispersed FMO further improves the robustness of the system to transmission errors. Results obtained with the list decoder coupled with dispersed FMO show a significant gain, with average PSNR gains ranging between 1 and 8 dB.

The performance of the error-control methods relative to subjective quality was tested and the results are illustrated in Fig. 12. From these results it is clear that the list decoder method outperforms the other methods where the PSNR gain on a frame-by-frame basis can go up to 18.1 dB relative to the standard. Thus the list decoder manages to recover video sequences at an acceptable level of quality even when transmitted over noisy environments.

5 Comments and conclusion

This paper has presented the application of list decoding to provide resilient transmission of H.264/AVC video content. Instead of exploiting structured redundancy introduced by traditional channel coding methods, the list decoder algorithm adopted exploits the residual source redundancy which is left by the encoder after compression to provide error-control capabilities without affecting the transmission bit rate. The strength of the algorithm is compared to the standard video decoder and a modified Viterbi algorithm proposed in [33],

doi: 10.1049/iet-com.2010.0717

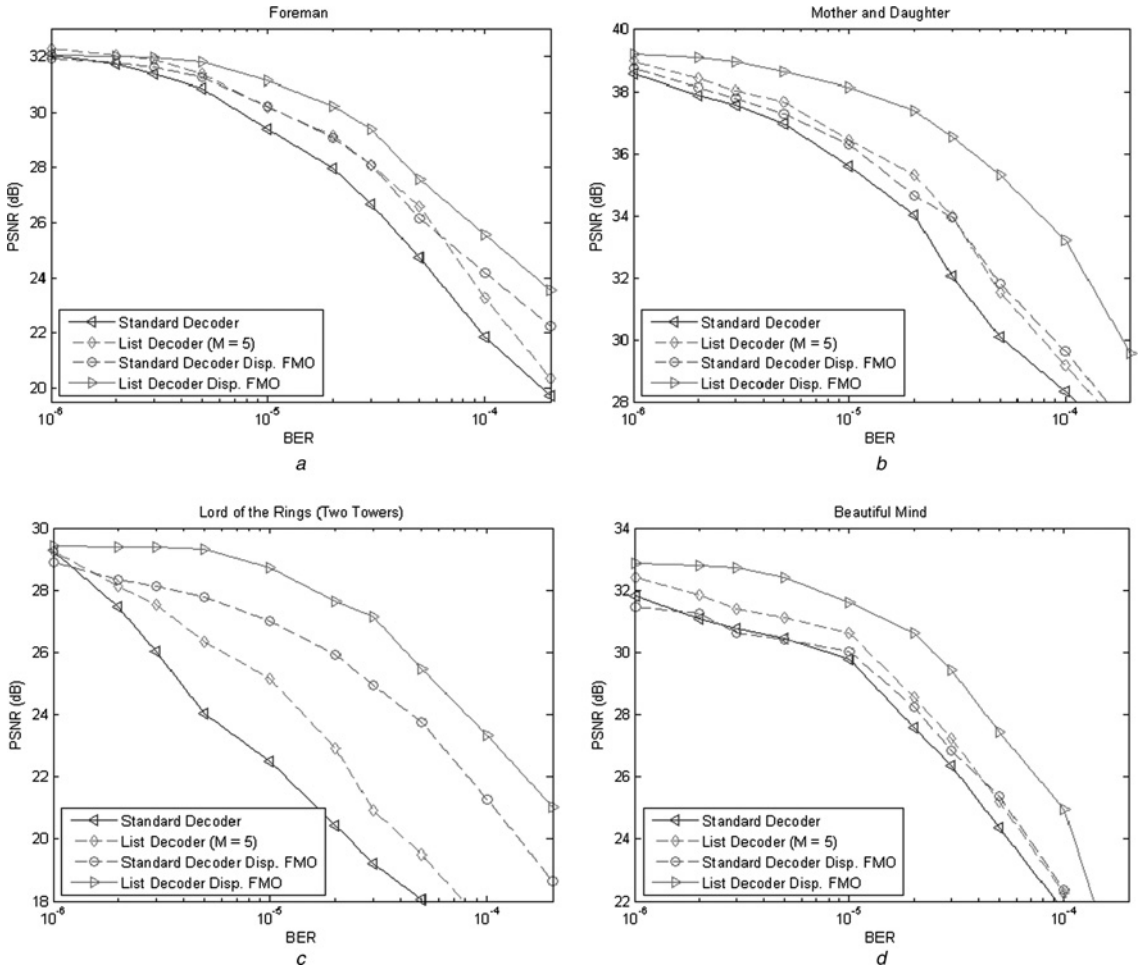

Fig. 11 *Performance of the error-control methods using dispersed FMO*

a Foreman
b Mother and Daughter
c Lord of the Rings (Two Towers)
d Beautiful Mind

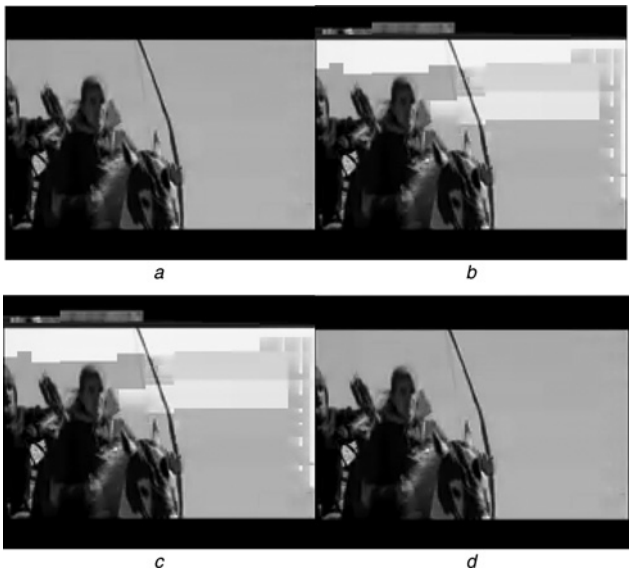

Fig. 12 *Frame 322 from the sequence Lord of the Rings (Two Towers) at 64 kbps*

a Reference sequence without errors
b Standard decoder
c Modified Viterbi decoder
d List decoder

where both the proposed list decoder and modified Viterbi approaches use the same information for error control.

Results have shown that the list decoder method outperforms both the standard decoding method and the modified Viterbi approach at the cost of a moderate increase in computational complexity of the decoder relative to the standard. However, the increase in complexity because of list decoding is still orders of magnitude smaller than the Viterbi approach. Furthermore, the list decoder can be implemented in conjunction with other error resilient tools adopted by the standard to further enhance their respective performance. The gain in perceptual quality achieved by the proposed method is substantial, where PSNR gains up to 18.1 dB on a frame-by-frame basis were registered.

6 References

1 Van der Schaar, M., Chou, P.P.: 'Multimedia over IP and wireless networks: compression, networking and systems' (Elsevier Inc., San Diego, CA, USA, 2007)
2 Sullivan, G.J., Wiegand, T.: 'Video compression – from concepts to the H.264/AVC standard', *Proc. IEEE*, 2005, **93**, (1), pp. 18–31
3 Advanced video coding for generic audiovisual services, ISO/IEC 14496-10 and ITU-T Rec. H.264, 2005
4 Stockhammer, T., Hannuksela, M.M., Wiegand, T.: 'H.264/AVC in wireless environments', *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**, (7), pp. 657–673

5 Larzon, L.-A., Degermark, M., Pink, S.: 'UDP lite for real time multimedia'. IEEE Proc. Int. Conf. on Communications, Vancouver, Canada, 1999

6 Superiori, L., Nemethova, O., Rupp, M.: 'Performance of a H.264/AVC Error detection algorithm based on syntax analysis'. Proc. Int. Conf. on Advances in Mobile Computing and Multimedia, Yogyakarta, Indonesia, December 2006

7 Chu, W.-J., Leou, J.-J.: 'Detection and concealment of transmission errors in H.261 images', *IEEE Trans. Circuits Syst. Video Technol.*, 1998, **8**, (1), pp. 74–83

8 Shyu, H.-C., Leou, J.-J.: 'Detection and concealment of transmission errors in MPEG-2 images – a genetic algorithm approach', *IEEE Trans. Circuits Syst. Video Technol.*, 1999, **9**, (6), pp. 937–948

9 Farrugia, R.A., Debono, C.J.: 'Enhancing the error detection capabilities of the standard video decoder using pixel domain dissimilarity metrics'. IEEE Proc. of Int. Conf. EUROCON, Warsaw, Poland, September 2007

10 Farrugia, R.A., Debono, C.J.: 'Enhancing the error detection capabilities of dct based codecs using compressed domain dissimilarity metrics'. IEEE Proc. Int. Conf. EUROCON, Warsaw, Poland, September 2007

11 Ye, S., Lin, X., Sun, Q.: 'Content based error detection and concealment for image transmission over wireless channel'. IEEE Proc. Int. Symp. on Circuits and Systems, Bangkok, Thailand, May 2003

12 Superiori, L., Nemethova, O., Rupp, M.: 'Detection of visual impairments in the pixel domain of corrupted H.264/AVC packets'. IEEE Int. Picture Coding Symp., Lisbon, Portugal, November 2007

13 Lehtoranta, O., Hämäläinen, T.D., Lappalainen, V.: 'Detecting corrupted intra macroblocks in H.263 video'. IEEE Workshop on Multimedia Signal Processing, Virgin Island, USA, December 2002

14 Khan, E., Lehmann, S., Gunji, H., Ghanbari, M.: 'Iterative error detection and correction of H.263 coded video for wireless networks', *IEEE Trans. Circuits Syst. Video Technol.*, 2004, **14**, (12), pp. 1294–1307

15 Farrugia, R.A., Debono, C.J.: 'Enhancing error resilience in wireless transmitted compressed video sequences through a probabilistic neural network core'. IEEE Proc. Int. Picture Coding Symp., Lisbon, Portugal, November 2007

16 Farrugia, R.A., Debono, C.J.: 'A robust error detection mechanism for H.264/AVC coded video sequences based on support vector machines', *IEEE Trans. Circuits Syst. Video Technol.*, 2008, **18**, (12), pp. 1766–1770

17 Farrugia, R.A., Debono, C.J.: 'Improved quality of experience of reconstructed h.264/avc encoded video sequences through robust pixel domain error detection'. IEEE Int. Workshop on Multimedia Signal Processing, Cairns, Australia, October 2008

18 Nemethova, O., Forte, G.C., Rupp, M.: 'Robust error detection for H.264/AVC using relation based fragile watermarking'. Int. Conf. on Systems, Signals and Image Processing, Budapest, Hungary, September 2006

19 Chen, M., He, Y., Lagendijk, R.L.: 'A fragile watermark error detection scheme for wireless video communications', *IEEE Trans. Multimed.*, 2005, **7**, (2), pp. 201–211

20 Park, W., Jeon, B.: 'Error detection and recovery by hiding information into video bitstream using fragile watermarking'. Proc. SPIE Visual Communication and Image Processing, January 2002, vol. 4671, pp. 1–10

21 Adsumilli, C.B., Farias, M.C.Q., Mitra, S.K., Carli, M.: 'A robust error concealment technique using data hiding for image and video transmission over Lossy channels', *IEEE Trans. Circuits Syst. Video Technol.*, 2005, **15**, (11), pp. 1394–1406

22 Lin, S.D., Meng, H.-C., Su, Y.-L.: 'A novel error resilience using reversible data embedding in H.264/AVC'. IEEE Int. Conf. Information, Communications and Signal Processing, Singapore, December 2007

23 Facciol, R., Farrugia, R.: 'Robust video transmission using reversible watermarking techniques'. IEEE Proc. Int. Symp. on Multimedia, Taichung, Taiwan, December 2010

24 Schwarz, H., Marpe, D., Wiegand, T.: 'Overview of the scalable H.264/MPEG4-AVC extension'. IEEE Int. Conf. Image Processing, Atlanta, USA, October 2006

25 Goyal, V.K.: 'Multiple description coding: compression meets the network', *IEEE Signal Process. Mag.*, 2001, **18**, pp. 74–93

26 Frossard, P., Carlos de Martin, J., Civanlar, R.: 'Media streaming with network diversity', *Proc. IEEE*, 2008, **96**, (1), pp. 39–53

27 Zhou, L., Wang, X., Tu, W., Mutean, G., Geller, B.: 'Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks', *IEEE J. Sel. Areas Commun.*, 2010, **28**, (3), pp. 409–419

28 Shiang, H.-P., Van der Schaar, M.: 'Distributed resource management in multi-hop cognitive radio networks for delay sensitive transmission', *IEEE Trans. Video Technol.*, 2009, **58**, (2), pp. 941–953

29 Buttigieg, V., Deguara, R.: 'Using variable length error-correcting codes in MPEG-4 video'. IEEE Int. Symp. on Information Theory, Adelaide, Australia, September 2005

30 Nguyen, H., Duhamel, P.: 'Estimation of redundancy in compressed image and video data for joint source-channel decoding'. IEEE Global Telecommunications Conf., San Francisco, USA, December 2003

31 Bergeron, C., Lamy-Bergot, C.: 'Soft-input decoding of variable-length codes applied to the H.264 standard'. IEEE Workshop on Multimedia Signal Processing, Siena, Italy, October 2004

32 Farrugia, R.A., Debono, C.J.: 'Robust transmission of H.264/AVC sequences using list decoding and source constraints'. IEEE Proc. Int. Conf. MELECON, Ajacio, France, May 2008

33 Nguyen, H., Duhamel, P., Brouet, J., Rouffet, F.: 'Optimal VLC sequence decoding exploiting additional video stream properties'. IEEE Proc. Int. Conf. on Acoustics, Speech and Signal Processing, Montreal, Canada, May 2004

34 Sabeva, G., Ben Jamaa, S., Kieffer, M., Duhamel, P.: 'Robust decoding of H.264 encoded video transmitted over wireless channels'. IEEE Workshop on Multimedia Signal Processing, Victoria, Canada, October 2006

35 Lin, S., Costello, D.J.: 'Error control coding: fundamentals and applications' (Englewood Cliffs, Prentice-Hall, 1983)

36 Wiegand, T., Sullivan, G.J., Bjøntegaard, G., Luthra, A.: 'Overview of the H.264/AVC video coding standard', *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**, (7), pp. 560–576

37 Bjøntegaard, G., Luthra, A.: 'Context-adaptive VLC (CAVLC) coding of coefficients'. Document JVT-C02rl, Fairfax, VA, May 2002

38 Bauer, R., Hagenauer, J.: 'Symbol-by-symbol MAP decoding of variable length cores'. Proc. ITG Conf. on Source and Channel Coding, Munich, Germany, January 2000

39 H.264/AVC Software Coordination: 'JM software', ver. 12.2 [Online], available at http://iphome.hhi.de/suehring/tml

doi: 10.1049/iet-com.2010.0717