A Hybrid Error Control and Artifact Detection Mechanism for Robust Decoding of H.264/AVC Video Sequences

Reuben A. Farrugia, *Member, IEEE,* and Carl J. Debono, *Senior Member, IEEE*

***Abstract*—This letter presents a hybrid error control and artifact detection (HECAD) mechanism which can be used to enhance the error resilient capabilities of the standard H.264/advanced video coding (AVC) codec. The proposed solution first exploits the residual source redundancy to recover the most likelihood H.264/AVC bitstream. If error recovery is unsuccessful, the residual corrupted slices are then passed through a pixel-level artifact detection mechanism to detect the visually impaired macroblocks to be concealed. The proposed HECAD algorithm achieves overall peak signal-to-noise ratio gains between 0.4 dB and 4.5 dB relative to the standard with no additional bandwidth requirement. The cost of this solution translates in a marginal increase in the complexity of the decoder. In addition, this method can be applied in conjunction with other error resilient strategies and scales well with different encoding configurations.**

***Index Terms*—Error correction coding, error resilient video transmission, H.264/advanced video coding (AVC), learning systems.**

I. Introduction

Current trends in wireless communications provide for fast and location independent access to multimedia services. Due to its high-compression efficiency, H.264/advanced video coding (AVC) [1] is expected to become the video coding standard of choice for future multimedia applications and services. However, real-time conversational and multicast/broadcast services still provide challenges in wireless communication systems, as reliable delivery of multimedia content cannot be assured [2].

The H.264/AVC coding standard specifies several error resilient strategies aimed at minimizing the effect of transmission errors on the perceptual quality of the reconstructed video sequences. However, most transport layer protocols drop corrupted segments and thus the receiver must conceal all the dropped or lost segments contained within the dropped segment. Most of the time this forces the standard decoder to operate at a lower bound, since not all the information contained within a corrupted packet is un-utilizable [3]. For this purpose, several transport layer protocols were recently developed in order to allow partially damaged payloads to be passed on to the application layer [4].

Several extensions to the standard and error resilient strategies have been proposed in literature. These include a set of syntax and semantic violation rules [5] and pixel-level artifact detection mechanisms based on heuristic thresholds [6], [7], both presenting limited robustness, iterative solutions such as [8], with their high-complexity, and data-hiding techniques [9], [10], which reduce the image quality even when transmitting over an error-free channel. Machine learning algorithms were recently introduced to detect the visually impaired macroblocks (MBs) to be concealed [11]. However, these do not manage to recover the quality of the original image.

Joint source-channel methods were adopted in [12]–[16] to better protect the transmitted bitstream. However, the method proposed in [12] reduces the compression efficiency of the codec. On the other hand, [13]–[16] have limited error correction capabilities. Further solutions include interactive encoder–decoder error control [17] mechanisms, scalable video coding [18], and multiple description coding [19]. However, the interactive approaches become useless with increasing round-trip delay, while the other two mechanisms demand additional bandwidth.

This letter presents a hybrid error control and artifact detection (HECAD) mechanism which applies list-decoding as a preprocess to recover the most likelihood feasible slices and a pixel-level artifact detection mechanism as a postprocess to detect the regions to be concealed. This method outperforms the standard decoder where peak signal-to-noise ratio (PSNR) gains between 0.4 dB and 4.5 dB were registered.

This letter is organized as follows. The HECAD architecture is described in the following sections followed by the description of the error control method and artifact detection mechanisms. The simulation results are then presented in Section V while the concluding remarks are made in Section VI.

II. HECAD Architecture

The H.264/AVC decoder was modified in order to allow the decoding of partially damaged slices according to [5]. As illustrated in Fig. 1, the error control module is positioned ahead of the normal decoding stage while the artifact detection

Manuscript received January 21, 2009; revised August 7, 2009 and December 22, 2009. First version published March 15, 2010; current version published May 5, 2010. This paper was recommended by Associate Editor L. Chen.

The authors are with the Department of Communications and Computer Engineering, University of Malta, Msida MSD 2080, Malta (e-mail: reuben.farrugia@um.edu.mt; c.debono@ieee.org).

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

Digital Object Identifier 10.1109/TCSVT.2010.2045808

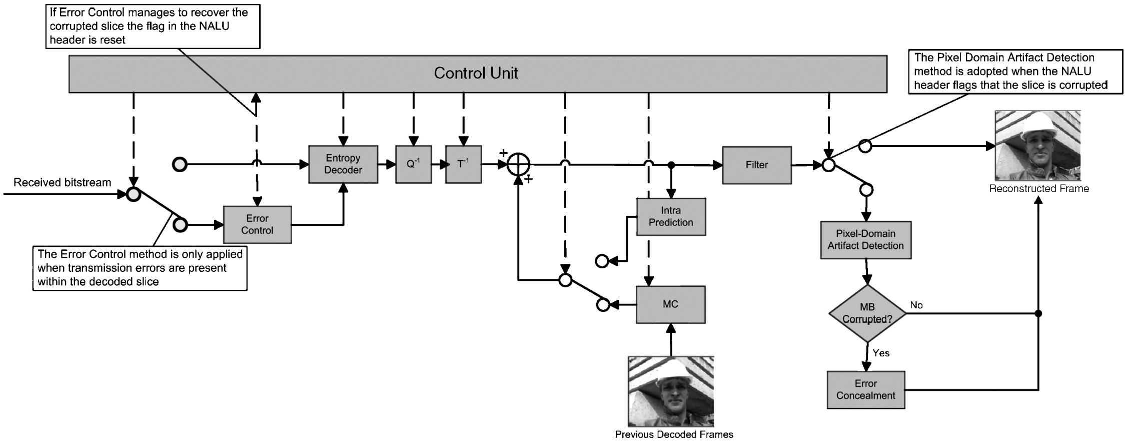

If Error Control manages to recover the corrupted slice the flag in the NALU header is reset
Control Unit
The Pixel Domain Artifact Detection method is adopted when the NALU header flags that the slice is corrupted
Received bitstream
Entropy Decoder
Q-1
T-1
+
+
Filter
Reconstructed Frame
Error Control
Intra Prediction
Pixel-Domain Artifact Detection
The Error Control method is only applied when transmission errors are present within the decoded slice
MC
MB Corrupted?
No
Yes
Error Concealment
Previous Decoded Frames

Fig. 1. HECAD method.

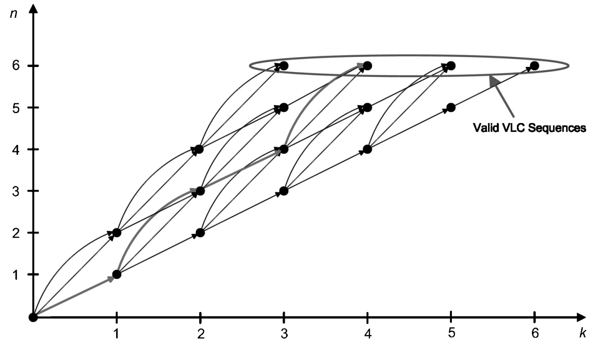


Fig. 2. Trellis representation used to estimate the H.264/AVC bitstream.

module is placed at the end of the process. Corrupted slices are detected using the error detection capabilities of the transport layer which flags them using the network abstraction layer unit (NALU) header [3]. This information is used to apply the HECAD mechanism only on these slices, ensuring that no additional delay is introduced when slices are uncorrupted.

The flagged slices are processed by the error control module which recovers the most likelihood feasible slice using the soft decision list decoding procedure, described in Section III. The module further informs the control unit whether it has succeeded in recovering the original slice. This is done by recomputing the checksum of the slice derived by the error control mechanism. The uncorrupted and recovered slices are passed directly to the H.264/AVC decoding process while the unrecovered ones are forwarded to the artifact detection mechanism which is described in Section IV, to detect those MBs which contain major visual distortion.

III. Error Control

The error control method adopts a modified version of the trellis representation published in [20], as shown in Fig. 2. In this trellis, the paths arriving at node $S_{n,k}$ correspond to the subsequences of k variable-length coding (VLC) codewords of n bits. Since the decoder has no knowledge of the number of VLC codewords contained within the VLC sequence, all the sequences of length N bits are considered to be candidate VLC sequences. Given this trellis representation,

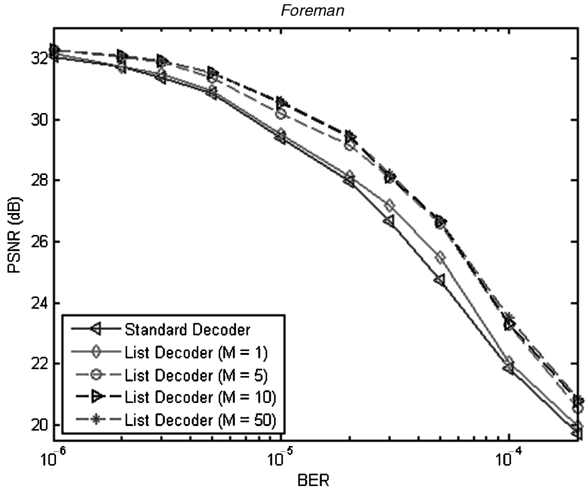


Fig. 3. Performance of the list decoder in terms of PSNR with M.

VLC sequence estimation techniques which derive the most likelihood sequence of VLC symbols based on the minimum distance decoding are adopted.

The error control schemes for image/video applications can exploit the residual source redundancies to ensure that only valid slices are considered to be feasible. For a bitstream to conform to valid H.264/AVC slices or frames, the following set of source constraints must be satisfied.

1) The length of the sequence of VLC codewords is equal to N bits, which is known by the receiver.
2) The number of MBs, N_{MB}, decoded must be equal to the number of MBs in the slice. All these MBs must be completely decoded.
3) The decoded symbols and derived parameters are within the ranges allowed by the H.264/AVC standard, and obey the syntax check rules provided in [5].

The bitstreams that conform to all these constraints are considered to be complete feasible sequences. On the other hand, sequences of lengths strictly smaller than N, whose number of decoded MBs is smaller than N_{MB} and which conform to the source constraints listed in (3) are considered as incomplete feasible sequences. The remaining sequences are not feasible and are thus pruned from the

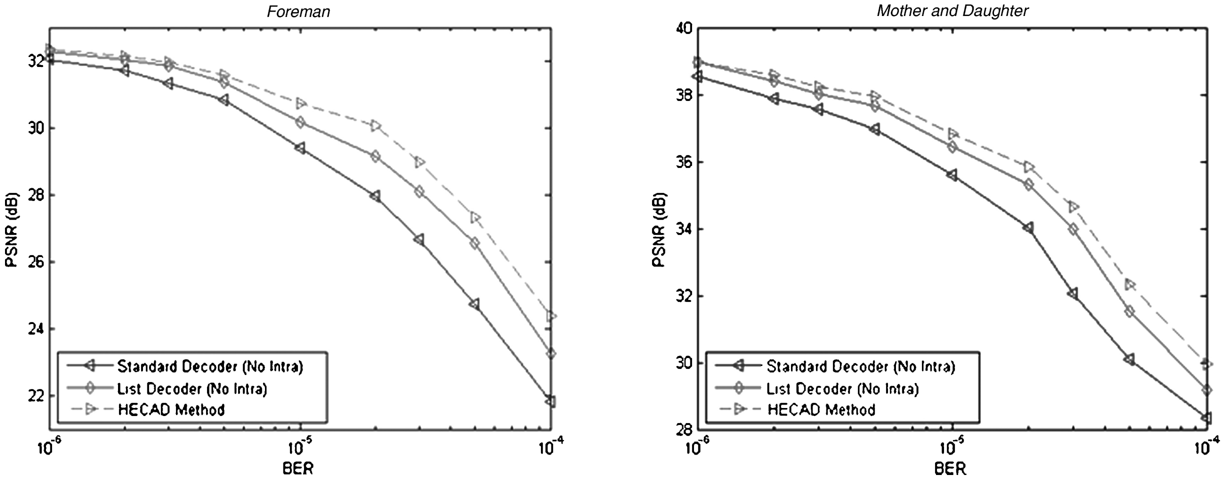


Fig. 4. Performance of the HECAD method (no intra refresh or FMO) for (a) *Foreman* and (b) *Mother and Daughter*.

trellis.

The context-adaptive VLC encoding method adopted by the baseline profile encodes the quantized coefficients using VLC tables which are switched depending on the previous syntax elements, and the VLC table adopted in the previous decoding step. To apply sequence estimation algorithms 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 decoding time to predict which VLC table must be loaded for the next process.

A novel sequence estimation algorithm was designed based on the list-decoding approach. This method employs three lists: L_{k-1} which contains incomplete feasible sequences of $k-1$ symbols, L_k which stores incomplete feasible sequences of k symbols, and F which stores complete feasible sequences. On receiving a corrupted slice the following list-decoding strategy is adopted.

1) Create an empty state $S_{0,0}$ containing an empty bitstream with the metric equal to zero and store it in list L_{k-1}.
2) A second list F is initialized as an empty list and the symbol time k is incremented by one.
3) Each state $S_{i,k-1}$ in list L_{k-1} contains a metric $\Lambda\left(S_{i,k-1}\right)$ and a bitstream $y\left(S_{i,k-1}\right)_0^{n_i}$of length n_i bits. For each state in list L_{k-1}, the following procedure is executed.
   a) For each VLC codeword c_j of length λ_{j} bits in the VLC table do the following.
      i) Derive the partial bitstream $y\left(S_{i,k}\right)_0^{n_i+\lambda_j}$which is made up of the concatenation of the bitstream stored in $S_{i,k-1}$ and the considered codeword c_j.
      ii) Compute the Euclidean distance between the partial bitstream $y\left(S_{i,k}\right)_0^{n_i+\lambda_j}$ and the first $n_i+\lambda_i$ modulated symbols received. This corresponds to the metric.
      iii) Bitstreams containing complete feasible bitstreams are stored in list F together with the corresponding metric.
      iv) Bitstreams containing incomplete feasible bitstreams are stored in list L_k together with the corresponding metric.
      v) States representing nonfeasible bitstreams are pruned.
   b) Reset list L_{k-1} as an empty list.
   c) 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. All the remaining states in L_{k-1} are pruned accordingly.
4) If L_k is empty and $k \geq 1$ terminate the process. Otherwise, increment k and go to step 3.

TABLE I
Reduced Slice Error Rate (%) of *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
Overall	**8.86**	**30.79**	**32.15**	**32.81**

Once the decoding algorithm is terminated, the list F will contain all the complete feasible VLC sequences. The optimal sequence of VLC codewords is then the complete feasible sequence with the smallest distance measure among all the sequences found in F.

IV. Artifact Detection

The residual corrupted slices generally contain bitstreams which do not cause syntax violations and are thus not detected by the syntax analysis procedure [5]. To detect the visually impaired MBs to be concealed, an artifact detection

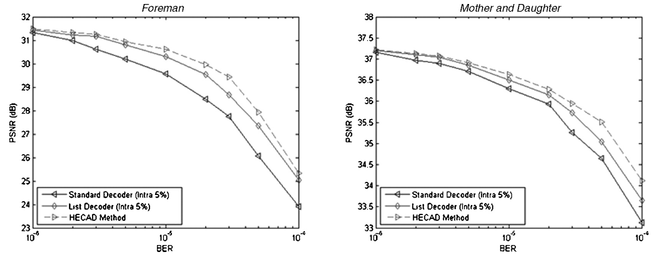
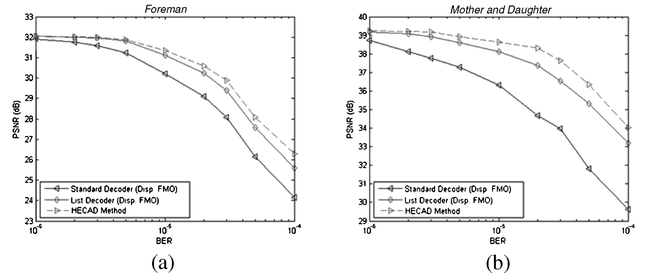


Fig. 5. Performance of the HECAD method (intra refresh 5%) for (a) *Foreman* and (b) *Mother and Daughter*.



(a) (b)

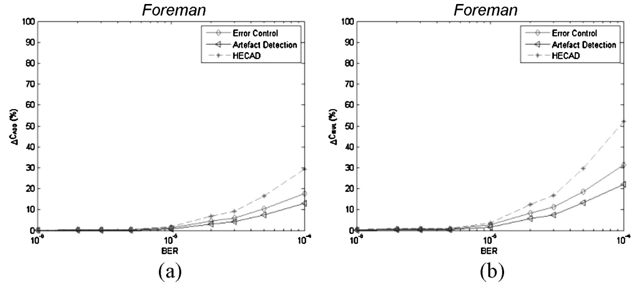
Fig. 6. Performance of the HECAD method (dispersed FMO) for (a) *Foreman* and (b) *Mother and Daughter*.

mechanism which exploits the inherent spatial redundancies is adopted.

This method uses a set of eight dissimilarity metrics to describe the reliability of the MB under test. The dissimilarity metrics employed are: 1) the average inter-sample difference across boundaries (*AIDB*), 2) the mean and 3) standard deviation of the internal *AIDB* per block (*IAIDBblock*), the 4) vertical and 5) horizontal internal *AIDB* (*IAIDB*), 6) the mean and 7) standard deviation of the average internal difference between subsequent blocks (*AIDSB*), and 8) texture consistency (*TC*). The feature vector derived from these metrics is fed to a support vector machine (SVM) classifier [21] which detects the visually impaired MBs and the undecodable ones to be concealed. The SVM was trained using a modified version of the sequential minimal optimization [22]. Additional information about the dissimilarity metrics and the implementation of the artifact detection method can be found in our previous work [11].

V. Simulation Results

The proposed methods were integrated within the joint model (JM) software version 12.2. The NALUs are encapsulated within RTP/UDP/IP packets, where the single NALU packetization mode is adopted. The JM software was further modified to allow the decoding of partially damaged bitstreams, such that they remain available for further processing. The raw video sequences used in this section were encoded at quarter common intermediate format (QCIF) resolution at 15 frames/s, with the format IPPP… and a data rate of 64 kb/s. The encoder adopts only slice structuring with a fixed number of 100 bytes in each slice. Each packet is modulated using binary phase shift keying and transmitted over an additive white Gaussian noise channel, where 34 different noise patterns for each bit error rate (BER) were considered. Unless otherwise specified, no other error resilience tools are considered to be adopted by the H.264/AVC encoder.



(a) (b)

Fig. 7. Complexity analysis of the HECAD method. (a) ΔC_{Add} and (b) ΔC_{Mul} for the *Foreman* sequence.

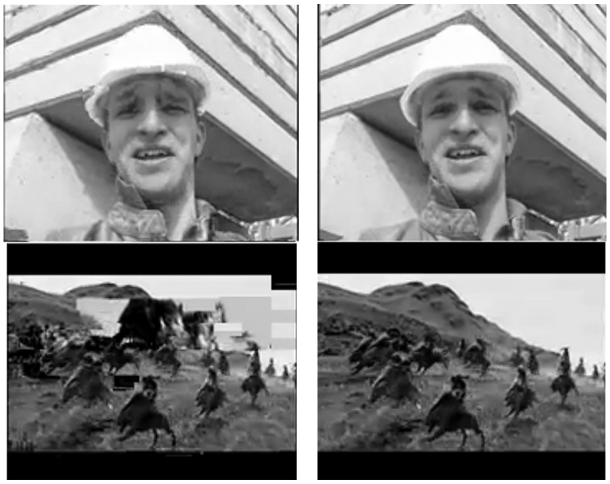


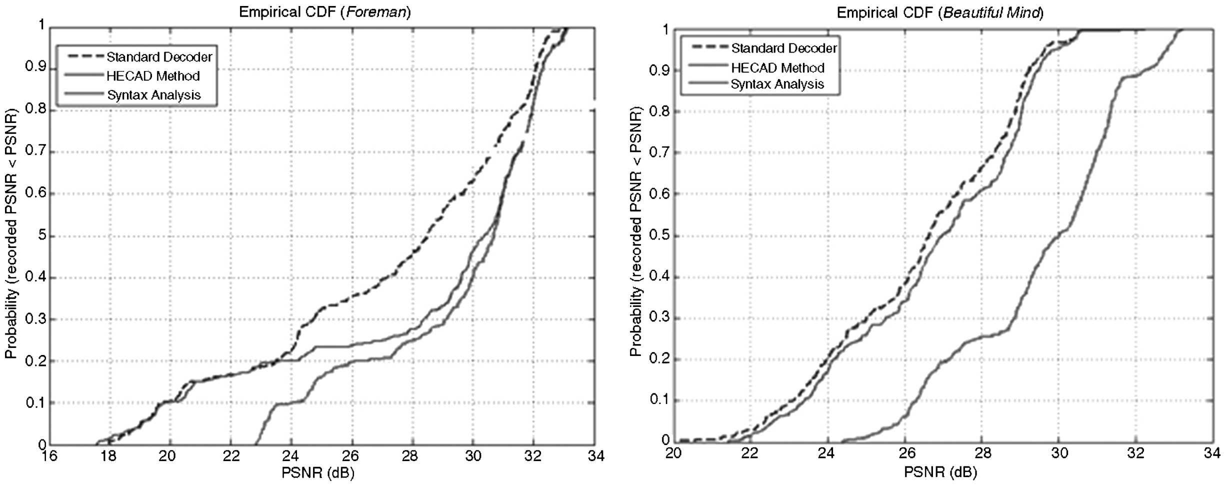
Fig. 8. Video sequences from (a) *Foreman* and (b) *Lord of the Rings (Two Towers)* using (left) Standard decoder and (right) HECAD algorithm.

TABLE II
Artifact-Detection Rate of Distorted MBs

Distortion Level (DL)	Recognition	Cross-Validation
Very Annoying (4)	100.00%	100.00%
Annoying (3)	100.00%	100.00%
Slightly Annoying (2)	94.81%	90.77%
Perceptible but not Annoying (1)	73.97%	78.18%
Overall	**94.60%**	**90.61%**

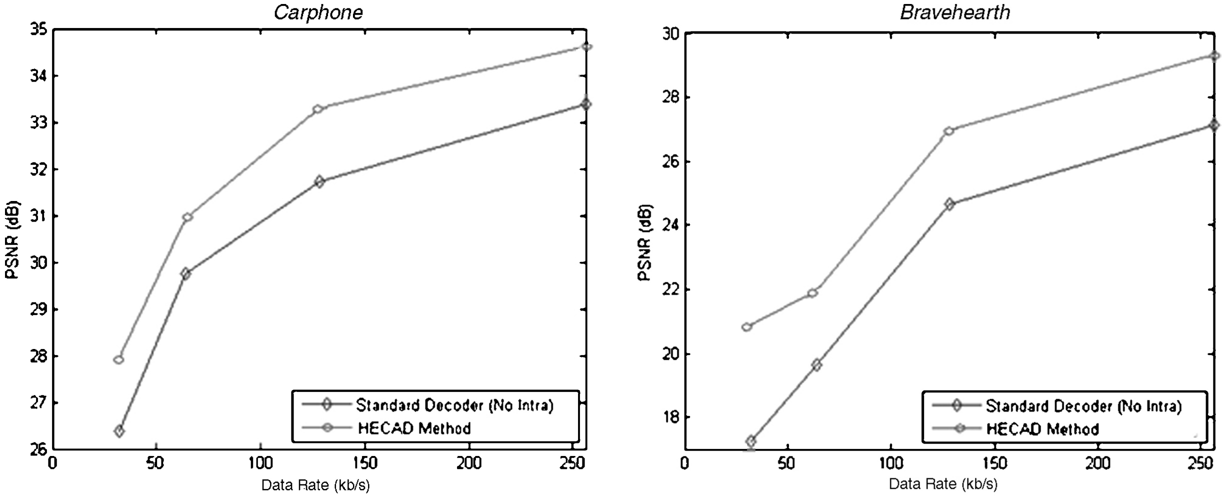
The error correction capability of the proposed method is dependent on the list size M which defines the number of sequences to be stored in list L_{k-1}. Fig. 3 and Table I show the results for the *Foreman* sequence. It is evident that the performance of the list decoder improves as M increases. An analysis of the result shows that while significant improvement in quality is achieved with $M > 1$, the increase in quality is not significant after $M > 5$. Moreover, the complexity will increase exponentially with increasing M, and therefore, M must be kept small to make this method useable in real-time applications. From this discussion, it was considered that $M = 5$ provides a fair compromise between robustness (30.79% of the corrupted slices are recovered) and complexity, and thus this value is kept for all the remaining simulation results.

The artifact detection module is designed to maximize the



(a) (b)

Fig. 9. Cumulative distribution of encoding PSNR for each frame at a bit rate of 64 kb/s and error rate 1.00E−005. (a) *Foreman*. (b) *Beautiful Mind*.



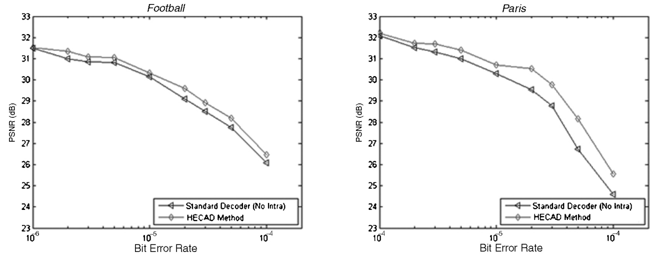
(a) (b)

Fig. 10. Performance of the HECAD method (no intra refresh or FMO) for (a) *Carphone* and (b) *Bravehearth* sequences at different data rates.

detection of highly distorted MBs which significantly distort the quality of the decoded frame while being more lenient with imperceptible artifacts. To optimize this module, a set of five video sequences (*Foreman*, *Carphone*, *Mobile*, *Coastguard*, and *News*) were used to derive the training and the testing data for experimentation purposes. A third set made up of another four video sequences (*Miss-America*, *Salesman*, *Akiyo*, and *Silent*) was used for cross-validation. The sequences from the three sets were encoded at QCIF resolution and errors were injected at different error rates. From the resulting distorted frames, a population of 3000 MBs was extracted at random and divided in three distinct groups. Each group consisting of 500 corrupted MBs and 500 uncorrupted MBs. The first group was used for training while the remaining two groups were used for recognition and cross-validation, respectively. More information on the procedures can be found in [11].

Table II summarizes the recognition and cross-validation performance provided by the SVM for different distortion levels. These results show that on average the SVM has managed to detect all the major visually distorted MBs (DL4 and DL3) which will be concealed. Most of the MBs which were not detected by the SVM provide minor or almost imperceptible visual distortion (DL2 and DL1) and thus the quality of the reconstructed video sequence will not be significantly affected. Moreover, the false-positives, that is uncorrupted MBs detected to be corrupted, are only 4.6%. Thus, the pixel-level artifact detection mechanism ensures that the most annoying artifacts are concealed.

The performance of the HECAD algorithm was first tested using two standard video sequences *Foreman* and *Mother and Daughter*. Figs. 4–6 illustrate the performance of the HECAD method relative to the modified list decoder, discussed in Section III, and the standard H.264/AVC decoder for different error rates and with intra refresh and flexible macroblock ordering (FMO). These results show that the HECAD architecture manages to boost the performance of the error control module, with PSNR gains of 0.1–3 dB relative to the list decoder being achieved. This corresponds to PSNR gains



(a) (b)

Fig. 11. Performance of the HECAD method (no intra refresh or FMO) for (a) *Football* and (b) *Paris* sequences at a different image resolution.

of 0.4–4.5 dB relative to the standard, constituting a considerable gain in quality. Furthermore, as shown in Fig. 7, the additional complexity, measured as the percentage increase in the number of additions (ΔC_{Add}) and multiplications (ΔC_{Mul}), introduced by this architecture is still manageable.

The gain in subjective quality is illustrated in Fig. 8. From these frames, it is clear that the HECAD method outperforms the standard. This is mainly attributed to the error control method which manages to recover a number of slices. To further boost the performance of the error control method, the artifact detection mechanism ensures that only the residual visually impaired MBs and invalid MBs are concealed.

In addition, the results for the cumulative distribution of the decoded PSNR for each frame are presented in Fig. 9. These distributions confirm that the HECAD method manages to increase the probability of having good quality video (on a frame-by-frame basis). Considering a PSNR of 25 dB to represent a satisfactory level of video quality, it can be concluded that the standard decoder achieves unsatisfactory quality of experience for 32.5% of *Foreman*, and for 30% of *Beautiful Mind*, while the syntax analysis [5] reduces this to 23.25% and 25%, respectively. On the other hand, the HECAD method achieves unsatisfactory quality for 16% of *Foreman*, and for 1% of *Beautiful Mind*.

The individual modules forming this algorithm were designed to be generalized, and therefore, the HECAD algorithm promises to scale well when applied to other applications which demand different encoding configurations.

As shown in Fig. 10, the HECAD algorithm outperforms the standard at different data rates, with PSNR gains in the region of 2 dB being observed, and thus can be applied in systems which employ higher or even lower data rates. Furthermore, the complexity increases marginally with increasing data rate. The performance of the HECAD was further tested on the *Football* and *Paris* sequences at common intermediate format resolution. As shown in Fig. 11, the HECAD architecture still outperforms the standard decoder in both cases.

VI. Comments and Conclusion

This letter has presented a solution which combines error control and artifact detection methods to improve the robustness of H.264/AVC. The error control module adopts a list decoding approach to recover 30.79% of the corrupted slices, while the artifact detection mechanism operates on the residual unrecovered slices to detect the visually impaired MBs to be concealed. This preconcealment mechanism manages to detect all the MBs which provide significant visual distortions, and reports an overall recognition rate of 94.6%. The HECAD architecture further boosts the performance of each individual component and thus provides a higher quality of experience. This performance improvement is achieved at the expense of an increase in the computational complexity of the decoder, which is still manageable and thus applicable for real-time low-bit rate applications. Furthermore, it was shown that the HECAD architecture is scalable and can cater for applications which demand different encoding configurations. Moreover, adopting advanced concealment techniques after the artifact detection mechanism further enhance the quality.

References

[1] *Advanced Video Coding for Generic Audiovisual Services*, ISO/IEC 14496-10 and ITU-T Rec. H.264, 2005.

[2] T. Stockhammer and M. M. Hannuksela, "H.264/AVC video for wireless transmission," *IEEE Wireless Commun.*, vol. 12, no. 4, pp. 6–13, Aug. 2005.

[3] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 657–673, Jul. 2003.

[4] M. Welzl, "Passing corrupted data across network layers: An overview of recent developments and issues," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 2, pp. 242–247, 2005.

[5] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," in *Proc. 4th Int. Conf. Adv. Mobile Comput. Multimedia*, Yogyakarta, Indonesia, 2006, pp. 1–10.

[6] R. A. Farrugia and C. J. Debono, "Enhancing the error detection capabilities of the standard video decoder using pixel domain dissimilarity metrics," in *Proc. IEEE Int. Conf. EUROCON*, Warsaw, Poland, Sep. 2007, pp. 1091–1095.

[7] S. Ye, X. Lin, and Q. Sun, "Content-based error detection and concealment for image transmission over wireless channels," in *Proc. IEEE Int. Symp. Circuits Syst.*, Bangkok, Thailand, May 2003, pp. 368–371.

[8] E. Khan, S. Lehmann, H. Gunji, and M. Ghanbari, "Iterative error detection and correction of H.263 coded video for wireless networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 12, pp. 1294–1307, Dec. 2004.

[9] M. Chen, Y. He, and R. L. Lagendijk, "A fragile watermark error detection scheme for wireless video communications," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 201–211, Apr. 2005.

[10] C. B. Adsumilli, M. C. Q. Farias, S. K. Mitra, and M. Carli, "A robust error concealment technique using data hiding for image and video transmission over lossy channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1394–1406, Nov. 2005.

[11] R. A. Farrugia and C. J. Debono, "A robust error detection mechanism for H.264/AVC coded video sequences based on support vector machines," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 12, pp. 1766–1770, Dec. 2008.

[12] V. Buttigieg and R. Deguara, "Using variable length error-correcting codes in MPEG-4 video," in *Proc. IEEE Int. Symp. Informat. Theory*, Adelaide, Australia, Sep. 2005, pp. 2379–2383.

[13] H. Nguyen and P. Duhamel, "Robust source decoding of variable-length encoded video data taking into account source constraints," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1077–1084, Jul. 2005.

[14] C. Bergeron and C. Lamy-Bergot, "Soft-input decoding of variable-length codes applied to the H.264 standard," in *Proc. IEEE Workshop Multimedia Signal Process.*, Siena, Italy, Oct. 2004, pp. 87–90.

[15] R. A. Farrugia and C. J. Debono, "Robust transmission of H.264/AVC sequences using list decoding and source constraints," in *Proc. 14th IEEE Mediterranean Electrotech. Conf. (MELECON)*, Ajaccio, France, May 2008, pp. 885–889.

[16] G. Sabeva, S. Ben Jamaa, M. Jieffer, and P. Duhamel, "Robust decoding of H.264 encoded video transmitted over wireless channels," in *Proc. IEEE Workshop Multimedia Signal Process.*, Victoria, Canada, Oct. 2006, pp. 9–13.

[17] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.

[18] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable H.264/MPEG4-AVC extension," in *Proc. IEEE Int. Conf. Image Process.*, Atlanta, GA, Oct. 2006, pp. 161–164.

[19] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005.

[20] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in *Proc. ITG Conf. Source Channel Coding*, Munich, Germany, Jan. 2000, pp. 111–116.

[21] N. Cristianini and J. D. Taylor, *Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[22] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Comp.*, vol. 13, no. 3, pp. 673–649, Mar. 2001.