Accepted in IEEE Symposium on Industrial Electronics & Applications (ISIEA 2014), 2014

Performance Evaluation of Maximum Likelihood Decoding Combined with Error Resilient Video Coding

Firouzeh Golaghazadeh[1], Stéphane Coulombe[2], Fabrice Labeau[3], François Caron[4]

Department of Software and IT Engineering[1]
École de technologie supérieure, Université du Québec
Montreal, Canada
firouzeh.golaghazadeh.1@ens.etsmtl.ca

Department of Software and IT Engineering[2]
École de technologie supérieure, Université du Québec
Montreal, Canada
stephane.coulombe@etsmtl.ca

Department of Electrical and Computer Engineering[3]
McGill University
Montreal, Canada
fabrice.labeau@mcgill.ca

Codec Engineering Group[4]
Vantrix Corporation
Montreal, Canada
francois.caron@vantrix.com

***Abstract*—In this work, we study the performance of the maximum likelihood decoding (MLD) approach in error-resilient video sequences to establish the performance improvement of this method compared to well-known error concealment approaches. In particular, we consider various interactions between error resilience coding and error concealment/correction. The error resilience methods under consideration include random intra macroblock updating and weighted error resilience. For error concealment/correction, we consider (i) the frame copy (FC), (ii) spatio-temporal boundary matching error concealments (STBMA), and (iii) MLD error correction. Our experimental results show that the best performance is achieved when the MLD interacts with weighted error resilience. Together, they yield, on average, about a 2 dB gain over using FC error concealment with weighted error resilience and a 1 dB gain over STBMA with identical error resilience. Furthermore, MLD with error resilience can be more than 10 dB better than FC without error resilience in certain cases.**

***Keywords*—Video transmission; video error resilience; video error concealment; video error correction; maximum likelihood decoding**

I. INTRODUCTION

Over the past few years, digital video communications have attracted considerable attention in a wide variety of application environments, such as mobile video streaming, video conferencing, telepresence video conferencing, etc. Restrictions on data storage, processing, and communication speed make video compression a mandatory step in processing video streams more efficiently, However, the high coding efficiency of current video compression standards (e.g., H.264, HEVC) render compressed video streams extremely vulnerable to transmission errors. Single bit errors in variable-length-codes (VLC) can desynchronize a decoder and an encoder, and lead to visual artifacts that may propagate over several video frames [1]. Transmission errors in intra and inter predicted frames may result in annoying visual artifacts

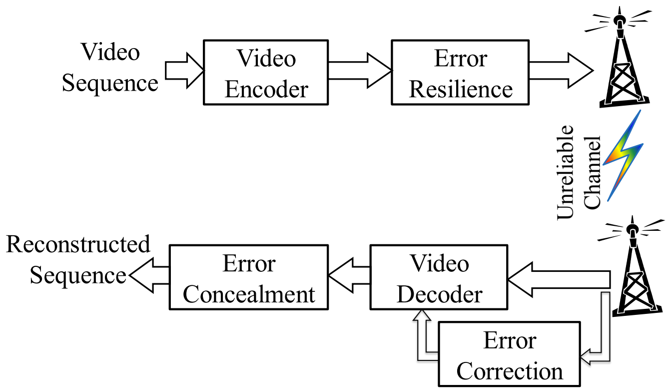


Fig. 1. General schematic of a video communication system

that can propagate spatially and/or temporally in reconstructed frames. Various error control mechanisms have been proposed in order to prevent transmission errors or enhance the perceived visual quality in the presence of transmission errors [2]. Fig. 1 illustrates a general schematic of a video transmission system with three error control mechanisms: (i) error resilience as a pre-processing error control at the encoder side, (ii) error correction, and (iii) error concealment (EC) at the decoder side. Error-resilient schemes inject redundancies during source coding to make the bitstreams more robust against transmission errors and restrict the end-to-end distortion caused by packet loss. The goal is to achieve maximum error tolerance at the expense of decreased coding efficiency. A wide range of studies has been conducted in this category of error control [3]. Among them, forced intra macroblock (MB) refresh is the most common approach that can cope with error propagation quite efficiently [4]. However, forcing the use of intra coding instead of selecting the optimal mode reduces the coding efficiency. As such, this has been the concern in most error resilient rate distortion optimization (ER-RDO) schemes [5-7]. Further, some methods exist that address ER-RDO based on motion estimation to mitigate the error propagation effect induced by the motion compensation process [8,9]. The downside is that all these approaches require knowledge – or estimation – of the channel's packet loss rate during the distortion-modeling phase. Inaccurate channel estimation can decrease their distortion-modeling performance [10]. In [11,12], a weighted distortion approach in RDO, dissociated from channel characteristics estimation and robust to changing channel conditions, is proposed. Two weighting factors in the RDO framework, one for motion estimation and one for mode decision, identify *sensitive* MBs, which are discarded from the prediction process. *Sensitive* MBs are defined as those having the greatest impact on error propagation; these MBs are intra-coded rather than inter-coded.

At the decoding side, EC attempts to reconstruct lost pixels by utilizing the inherent spatiotemporal correlations among adjacent pixels. There are three distinct classifications of EC: (i) spatial EC [13-15], (ii) temporal EC [16-18], and (iii) spatiotemporal EC [19,20]. Spatial EC schemes reconstruct lost pixels based on the smoothness property of the available spatially-adjacent pixels. Temporal approaches attempt to recover the motion information of the lost blocks by exploiting temporal correlation between neighboring frames. Spatio-temporal EC approaches exploit both temporal and spatial correlation features to reconstruct the missing areas. A state-of-the-art embodiment of this technique is proposed in [20], which is based on a spatial and temporal boundary-matching algorithm. In this method, lost motion vectors are recovered by a modified boundary matching algorithm that uses both spatial and temporal smoothness properties of the neighboring MBs. Each recovered motion vector points to a best candidate MB for the lost area. Partial differential equations are then used to smooth any discontinuity arising from the replaced MB.

Video EC works according to the assumption that all transmission errors result in packet loss. In practice, network congestion results in packet loss while attenuation, fading, and the likes result in corrupted packets. These partially damaged packets may however contain valuable information, which is useful for improving the visual quality of the reconstructed video [21-23]. The goal of video error correction is to use these corrupted packets to recover the originally sent values rather than assuming that the whole packet is lost. These approaches attempt to correct errors by modifying the received bits into a most likely sequence of bits. Many publications on topics such as joint source-channel decoding [24-27] and list decoding [28-32] have been proposed to correct transmission errors. In general, these use the channel information or the source semantics verifier to choose the most likely packet that is compatible with the video decoder. However, all these approaches suffer from a general drawback of having a fairly large solution space for candidate packets, as well as a computationally complex decoding process. These issues restrict the use of the approaches in low-delay applications. Recently, a less complex approach than list decoding has been proposed in [33-35], in which a maximum likelihood decoding (MLD) method is applied at the syntax element (SE) level instead of the whole packet or slice level. The solution space is therefore limited to a set of valid codewords for each specific SE.

In this work, we show that decoder-side error control mechanisms do not eliminate the need for error resilience, since their performance rapidly decreases when too many MBs are corrupted. This fact is highlighted by employing three different approaches during the encoding process: (i) non-error resilience, (ii) random intra MB updating and (iii) weighted error resilience [12]. Furthermore, we will show that the performance of error robustness approaches can be improved considerably if a proper error control mechanism at the decoder-side is employed. We present the interaction of the above-mentioned error resilience approaches with three different decoder-side error control mechanisms. Frame copy (default EC in JM) [36], state-of-the-art EC [20], and MLD error correction [35] are used as the decoder-side error control approaches in this work. The performance of each interaction is evaluated for different channel conditions. To the best of our knowledge, this is the first time that the performance evaluation of such interactions has been reported. Our experimental results show that the best visual quality performance is achieved when MLD interacts with weighed error resilience coded sequences.

The rest of the paper is organized as follows. Section II and III describe the weighted error resilience and MLD approaches, respectively. Section IV presents the proposed systems, combining error resilience with EC/correction approaches. Section V describes the experimental setup and presents the experimental results to verify the performance of the different interactions. Concluding remarks are drawn in Section VI.

II. WEIGHTED ERROR RESILIENCE

During the encoding process, the encoder has a variety of coding options to choose from, such as: motion vector, quantization level, block or partition size, prediction mode, reference frame, etc. The role of the encoder is to minimize the distortion D, given a maximum rate R. This constrained minimization problem is commonly solved using Lagrangian optimization. The H.264 test model [36] simplified the problem of coding options determination by using two Lagrangian optimization steps: motion estimation and mode decision. During motion estimation, motion vectors are optimized by using (1):

$$J_{me} = D_{\mathrm{SAD}} + \lambda_{me} \cdot R_{me} \tag{1}$$

where λ_{me} is the Lagrange multiplier for motion estimation and R_{me} denotes the number of bits required to code the motion vectors. The sum of absolute differences (D_{SAD}) is used as the distortion measure in the H.264 reference software. After the optimal motion vectors are determined, the best coding mode for each MB is selected by minimizing (2):

$$J_{md} = D_{SSD} + \lambda_{md} \cdot R_{md} \tag{2}$$

where λ_{md} is the Lagrange multiplier for mode decision. The sum of squared differences (D_{SSD}) is used as the distortion measure, and R_{md} denotes the number of bits required to code the partition using that mode. Since predictive coding is primarily responsible for error propagation, an appropriate weighting factor has been added to the distortion measurement in [12] to take into consideration the error propagation effect in the RDO framework. Thus, the distortion measurements in (1) and (2) are weighted as follows:

$$J_{me} = w_{me} \, .D_{SAD} + \lambda_{me} \, . \, R_{me} \tag{3}$$

$$J_{md} = w_{md} \, . \, D_{SSD} + \lambda_{md} \, . \, R_{md} \tag{4}$$

where w_{me} and w_{md} are the weighting factors for motion estimation and mode decision, respectively. These weighting factors reduce the usage of MBs having a high impact on error propagation for prediction. In other words, motion vectors are assigned to MBs having less of an impact on error propagation.

III. Maximum Likelihood Error Correction Decoding

By keeping corrupted packets at the decoder side, error correction allows the recovery of at least some of the MBs that are actually intact (those located before the first error in the packet) or that contain few errors. MLD [35] attempts to estimate the likeliest syntactically valid slice based on the received erroneous slice. The likeliest transmitted slice (S^*) can be obtained from (5):

$$\begin{aligned} S^* &= \underset{S_T \in H}{\operatorname{argmax}} \left\{ \mathrm{P}(S_T \,|\, S_R) \right\} \\ &= \underset{S_T \in H}{\operatorname{argmax}} \left\{ \frac{\mathrm{P}(S_R \,|\, S_T) \times \mathrm{P}(S_T)}{\mathrm{P}(S_R)} \right\} = \underset{S_T \in H}{\operatorname{argmax}} \left\{ \mathrm{P}(S_R \,|\, S_T) \times \mathrm{P}(S_T) \right\} \end{aligned} \tag{5}$$

where S_R is the received corrupted slice and H is the set of all hypothetically transmitted slices from which S_T is selected. $P(S_T|S_R)$ is the probability that S_T was transmitted given that S_R was received. $P(S_R|S_T)$ represents the probability that S_R was received, given that S_T was transmitted. $P(S_T)$ represents the probability that S_T was transmitted. This slice-level error correction scheme suffers from a large solution space, as well as othere list decoding approaches. To address this problem, the correction process was transferred from the slice-level to the SE level by using (6):

$$s_i^* = \underset{s_{i,T} \in C_i}{\operatorname{argmax}} \left\{ \mathrm{P}(s_{i,R} \,|\, s_{i,T}) \times \mathrm{P}(s_{i,T} \,|\, \Psi_i, s_1^*, s_2^*, ..., s_{i-1}^*) \right\} \tag{6}$$

where s_i^* is chosen from the codebook C_i containing all the valid codewords for the i[th] SE. The obtained s_i^* is the likeliest SE to the hypothetically transmitted SE ($s_{i,R}$). Ψ_i can be interpreted as the set of variables that hold the context needed to decode the i[th] SE. This reduces the solution space from 2^n to a set of valid codewords for each SE, where n is the number of bits in each slice. To evaluate $P(s_{i,R}|s_{i,T})$ the first probability in (6), two means were employed. One was to consider this probability as independent Bernoulli trials since the decoder can assume that each bit has the same probability of being in error. The Hamming distance, the number of flipped bits, between the hypothetically transmitted SE ($s_{i,T}$) and the received SE ($s_{i,R}$) represents the number of successes [33]. The second method involves evaluating the $P(s_{i,R}|s_{i,T})$ likelihood function using the soft-output information provided by the channel decoder [34,35].

The second probability in (6), $\mathrm{P}(s_{i,T} \,|\, \Psi_i, s_1^*, s_2^*, ..., s_{i-1}^*)$, is measured by modeling the probability distribution of SEs. The modeled SEs in [35] are the first five SEs of the slice header (i.e., *first_mb_in_slice*, *slice_type*, *pic_parameter_set_id*, frame*_num,* and *pic_order_cnt_lsb*) as well as the SEs employed for prediction (i.e., *mb_type mb_skip_run, sub_mb_type*, *intra_chroma_pred_mode*, *intra4x4_pred_mode*, *mvd_l0*, and *coded_block_pattern*). Unfortunately, models for residual coefficients are currently lacking, which limits the performance of the method at high bit rates, where residual information becomes more prominent.

IV. Error Resilience and Error Correction Interactions

We found that the body of work dedicated to studying the performance of error resilience and EC, although very thorough, lacks studies comparing their combined performance. It seems obvious that a good EC/correction method combined with error resilience will improve the perceived visual quality of the reconstructed video over the sole use of error resilience. What is yet to be quantified is how much more effective such a combination would perform under various channel conditions. To our knowledge, this paper is the first to study and report the performances of such interactions.

To encode video sequences, we employ the standard JM encoder [36] with three different encoding methods: (i) non-error resilience, (ii) random intra MB updating, and (iii) weighted error resilience [12]. To handle corrupted sequences, we use both post-processing approaches: frame copy and state-of-the-art EC [20], along with MLD error correction [35], at the decoder side. In this work, the performance of the above-mentioned decoder-side error control mechanisms in *individual* and *combined* systems is evaluated. In particular, *individual* systems evaluate the performance of the decoder-side error control approaches in non-error-resilient video sequences. In the *combined* systems, the performance is evaluated when random intra MB updating or weighted error resilient video coding are used. Our results show transmission error effects on low resolution video sequences. Results for state-of-the-art error correction technique [35] are only reported for high resolution sequences.

V. Experimental Results

We carry out the experiments using H.264's Baseline profile on the first 100 frames of the QCIF (176x144) sequences *Football, Crew, Harbour, Soccer* and *City* [37]. These sequences are coded in *IPPP…* (a single intra frame) at a frame rate of 15 Hz. Slices contain a single row of MBs and are packed into RTP packets. The sequences are coded with three different coding approaches: one uses the JM 18.5 software [36] without any error resilience approach, the second uses the JM software with 20% *random-intra MB-*

updating, and the last one uses a low computationally complex form of weighted error resilience approach [12]. We denote these three encoding approaches as "S", "RIU" and "W".

As an experiment, random errors occur between the 21th frame and the 80th frame. The first 20 frames are kept intact to allow the MLD approach to gather video statistics. The transmission simulations are repeated 100 times for each sequence at each different channel condition.

Three different approaches are then used to handle the corrupted sequences: (i) EC by frame copy (FC) [36], (ii) state-of-the-art EC (STBMA) [20], and (iii) error correction using soft-output MLD [35]. The MLD approach stops when the Hamming distance between the likeliest SE and the received bits is larger than 1; the MBs that cannot be corrected by the MLD approach are concealed using STBMA.

For performance evaluation, we calculate the average peak signal-to-noise ratio (PSNR) from frames 21 to 80 to compare the visual quality of the reconstructed corrupted sequence. Two categories of experiments are carried out. In the first category, sequences with different bit rates pass through a channel with a fixed bit error rate (BER). In the second category different channel BERs are considered for the fixed bit rate sequences. Fig. 2 and Fig. 3 summarize the results for fixed BER and fixed bit rates, respectively.

Fig. 2 shows that the PSNR performance of the reconstructed sequence decreases rapidly when there is no error robustness. No matter how they handle corrupted sequences, random intra MB updating and weighted error resilience approach can considerably increase the PSNR in comparison with the non-error resilience approach. More improvement can be achieved when a proper decoder-side error control approach is employed. When weighted error resilience is used, we observe that, on average, MLD improves PSNR by about 2 dB over FC for both *Football* and *City* sequences. In addition, the average PSNR improvement of MLD over STBMA is 2 dB for the *Football* sequence and 1 dB for the *City* sequence, and over 2 dB at higher bit rates. As it can also be seen in Fig. 2 the interaction between MLD and weighted error resilience has the best performance compared with all the other eight interactions. We observe that MLD with weighted error resilience is nearly 6 dB better than without error resilience for *Football* and 10 dB better for *City*.

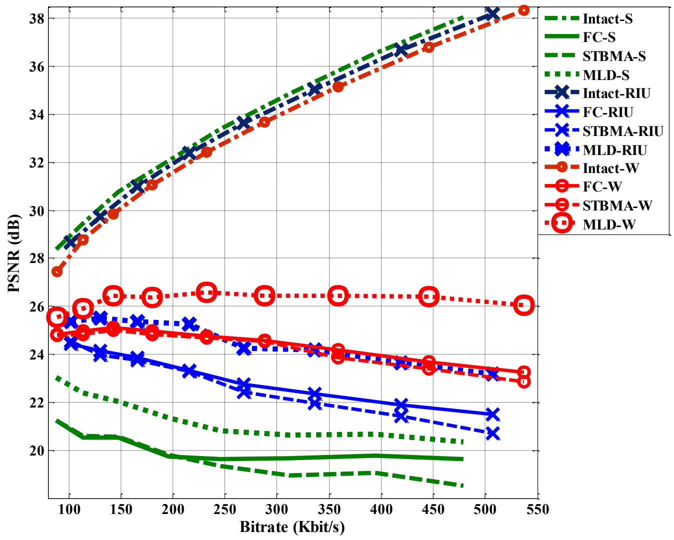
For the second category of experiments, different channel signal-to-noise ratios (SNRs) ranging from 5.5 dB to 7 dB are applied on fixed bit rate sequences. The SNR variation range corresponds to BERs in the interval [10^{-5},10^{-4}]. The packet loss rate (PLR) can be calculated from the BER using (7).

$$PLR = 1-(1-BER)^N \tag{7}$$

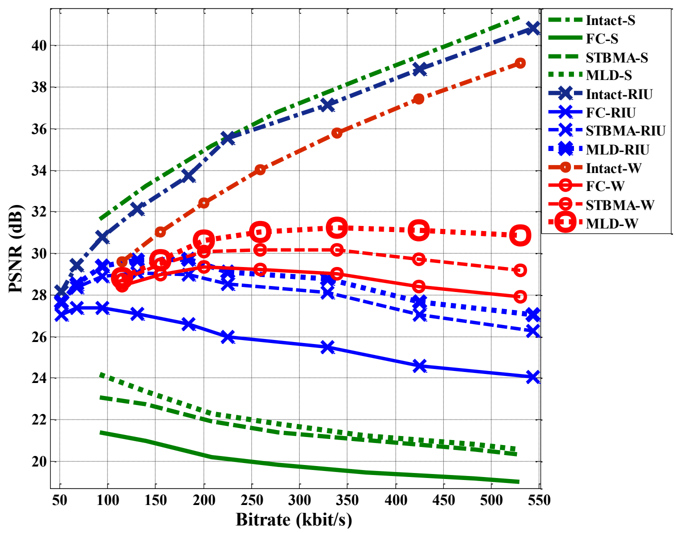
where N is the packet length. In our simulations, N was set as the average slice length of P frames. Fig. 3 shows that for different PLR experiments, MLD still maintains its average PSNR improvement of 2 dB over FC and STBMA when PLR exceeds 10%. For a numerical comparison, the average PSNR values for a fixed bit rate (around 350kbps) at three different BERs (low, moderate and high) are given in Table I. On average, MLDs with a weighted error resilience show 2.06 dB in gains over FC, and 1.73 dB in gains over STBMA, with the same weighted error resilience approach when the BER is 5×10^{-5}.

Fig. 4 shows the frame-by-frame PSNR performance of the first 80 frames of *Football* and *City* sequences. It is measured with a fixed quantization parameter and fixed BER. The figure presents the average PSNR of each frame for the different interactions. It shows the superiority of MLD with weighted error resilience in comparison with the other interactions.

From the results of Fig. 2, Fig. 3, Fig. 4 and Table I, it can be inferred that for fast motion video sequences, e.g., *Football*, STBMA has the same performance as FC due to low temporal correlations in such sequences. It is noteworthy that the SEs related to the residuals were not modeled in the implemented MLD approach [35], which to some extent, degrades the performance of the MLD at high bit rates.

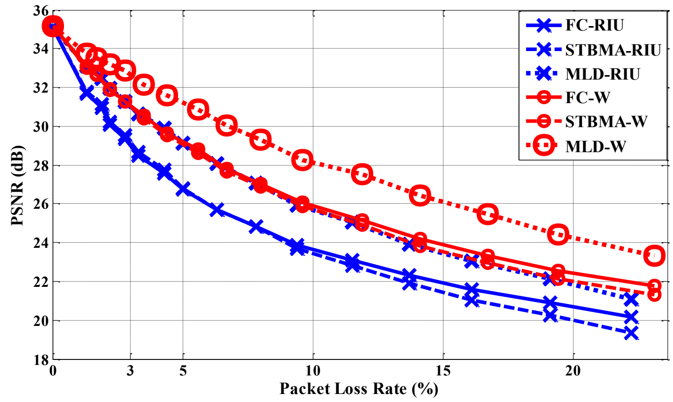


(a) Football

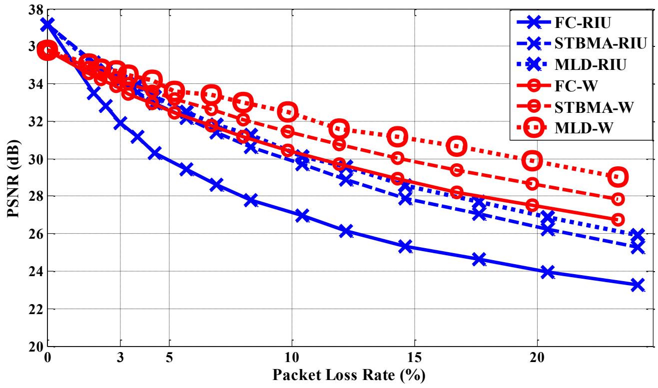


(b) City

Fig. 2. Rate distortion curves for *Football* and *City* sequences in a channel with SNR=6 dB (BER=5×10^{-5}). Intact represents an intact sequence without any error.

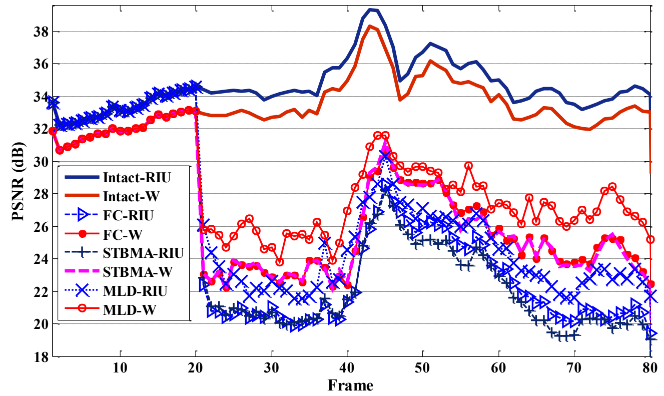


(a) Football

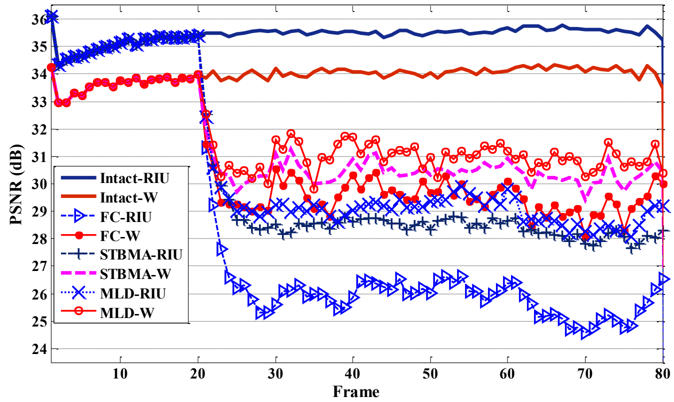


(b) City

Fig. 3. Average PSNR versus packet loss rate (PLR) for *Football* and *City* sequences at 336kbps and 329kbps, respectively by using 20% *random-intra MB-updating*, and 359kbps and 339kbps by using weighted error resilience coding. The PLR range corresponds to BER in the interval [10^{-5} , 10^{-4}].



(a) Football



(b) City

Fig. 4. PSNR over frame numbers for *Football* and *City* sequences at 336kbps and 225kbps, respectively by using 20% *random-intra MB-updating,* and 359kbps and 339kbps by using weighted error resilience coding. The channel's SNR was 6 dB (BER=5×10^{-5}) in both cases.

TABLE I. COMPARISON OF THE AVERAGE PSNR OBSERVED WITH DIFFERENT METHODS AS A FUNCTION OF THE BIT ERROR RATE. (THE DIFFERENCE BETWEEN EACH METHOD WITH FC APPROACH APPEARS IN PARENTHESES)

Sequence Name	Encoding Type	Bit Error Rate								
		10^{-5}			5×10^{-5}			10^{-4}		
		FC	*STBMA*	*MLD*	*FC*	*STBMA*	*MLD*	*FC*	*STBMA*	*MLD*
City	**RIU** (329kbps)	**32.8**	**34.74** (1.94)	**34.59** (1.79)	**26.97**	**29.71** (2.74)	**30.13** (3.16)	**23.97**	**26.23** (2.26)	**26.89** (2.92)
	W (339kbps)	**34.21**	**34.6** (0.39)	**34.76** (0.55)	**30.42**	**31.44** (1.02)	**32.51** (2.09)	**27.5**	**28.66** (1.16)	**29.89** (2.39)
Football	**RIU** (336kbps)	**30.07**	**30.23** (0.16)	**31.97** (1.9)	**23.09**	**22.83** (-0.26)	**25.08** (1.99)	**20.19**	**19.37** (-0.82)	**21.09** (0.90)
	W (359kbps)	**31.61**	**31.55** (-0.06)	**33.03** (1.42)	**25.14**	**24.9** (-0.24)	**27.52** (2.38)	**21.77**	**21.29** (-0.48)	**23.33** (1.56)
Harbour	**RIU** (350kbps)	**31.23**	**31.37** (0.14)	**31.59** (0.36)	**27.39**	**27.85** (0.46)	**28.15** (0.76)	**24.41**	**24.89** (0.48)	**25.41** (1)
	W (350kbps)	**31.09**	**31.15** (0.06)	**31.11** (0.02)	**27.53**	**27.62** (0.09)	**27.87** (0.34)	**24.74**	**24.73** (-0.01)	**25.28** (0.54)
Crew	**RIU** (329kbps)	**33.72**	**34.21** (0.49)	**34.96** (1.24)	**28.01**	**28.67** (0.66)	**29.94** (1.93)	**24.81**	**25.27** (0.46)	**26.32** (1.51)
	W (379kbps)	**35.12**	**35.25** (0.13)	**36.06** (0.94)	**30.21**	**30.46** (0.25)	**32.17** (1.96)	**26.93**	**27.17** (0.24)	**28.66** (1.73)
Soccer	**RIU** (304kbps)	**31.13**	**32.45** (1.32)	**33.82** (2.69)	**22.81**	**24.34** (1.53)	**26.08** (3.27)	**19.47**	**20.53** (1.06)	**21.99** (2.52)
	W (351kbps)	**34.27**	**34.52** (0.25)	**36.07** (1.8)	**26.4**	**26.92** (0.52)	**29.94** (3.54)	**22.07**	**22.56** (0.49)	**25.04** (2.97)

VI. CONCLUSIONS

In this paper, we have studied the interaction of non-error resilience, random intra MB updating, and weighted error resilience approaches, with three different error-control mechanisms at the decoder side, namely, FC, STBMA and MLD methods. Experimental results show that, under various testing conditions, MLD is a better choice for handling corrupted packets when sequences are coded with weighted error resilience in comparison with other interactions. On average, the interaction of weighted error resilience and MLD, brings about 2 dB gains over FC and 1 dB over STBMA, with the same weighted error resilience. Furthermore, MLD with weighted error resilience can be over 10 dB superior to non-error resilience interactions with FC. It is worth mentioning that since the SEs related to residuals are not modeled in MLD, we expect this combination to perform even better when the SEs are modeled in future research studies.

ACKNOWLEDGMENT

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant.

REFERENCES

[1] W.T. Tan and S. Bo, "Temporal propagation analysis for small errors in a single-frame in H.264 video," 15th IEEE Int. Conf. on Image Processing, San Diego, USA, pp. 2864–2867, 2008.

[2] Y. Wang, J. Ostermann, and Y. Zhang, "Video processing and communications," Upper Saddle River: Prentice Hall, 2002.

[3] Y. Wang, S. Wenger, J. Wen, and A.K. Katsaggelos, "Error resilient video coding techniques," IEEE Signal Processing Magazine, vol. 17, no. 4, pp. 61–82, 2000.

[4] J.Y. Liao, and J. Villasenor, "Adaptive intra block update for robust transmission of H.263," IEEE Trans. Circuits and Systems for Video Technology, vol. 10, no. 1, pp. 30–35, 2000.

[5] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," IEEE J. Selected Areas in Communications, vol. 18, no. 6, pp. 952–965, 2000.

[6] R. Zhang, S.L. Regunathan, and R. Kenneth, "Video coding with optimal inter/intra-mode switching for packet loss resilience," IEEE J. Selected Areas in Communications, vol. 18, no. 6, pp. 966–976, 2000.

[7] T. Stockhammer, D. Kontopodis, and T. Wiegand, "Rate-distortion optimization for JVT/H.26L video coding in packet loss environment," Int. Packet Video Workshop, Pittsburg, PA, 2002.

[8] H. Yang and K. Rose, "Rate-distortion optimized motion estimation for error resilient video coding," IEEE Int. Conf. Acoustics, Speech, and Signal processing, vol. 2, pp. 173–178, 2005.

[9] S. Wan and E. Izquierdo, "Rate-distortion optimized motion-compensated prediction for packet loss resilient video coding," IEEE Trans. on Image Processing, vol. 16, no. 5, pp. 1327–1338, 2007.

[10] O. Harmanci and A. M. Tekalp, "A stochastic framework for rate-distortion optimized video coding over error-prone networks," IEEE Trans. on Image Processing, vol. 16, no. 3, pp. 684–697, 2007.

[11] S. Nyamweno, R. Satyan, S. Solak, and F. Labeau, "Weighted distortion for robust video coding," IEEE 42nd Asilomar Conf. on Signals, Systems and Computers, pp. 1277–1281, 2008.

[12] S. Nyamweno, R. Satyan, and F. Labeau, "Error resilient video coding via weighted distortion," IEEE Int. Conf. on Multimedia and Expo, pp. 734–737, 2009.

[13] H. Sun, and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," IEEE Trans. on Image Processing, vol. 4, no. 4, p. 470–477, 1995.

[14] S.V. Chapaneri, and J.J. Rodriguez, "Low complexity error concealment scheme for intra-frames in H.264/AVC," 16th IEEE Int. Conf. on Image Processing (ICIP), pp. 925–928, 2009.

[15] P. Salama, N.B. Shroff, and E.J. Delp, "Error concealment in encoded video streams," Signal recovery techniques for image and video compression and transmission, Springer, pp. 199–233, 1998.

[16] W.M. Lam, A.R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 5, p. 417–420, 1993.

[17] H. Sun, J.W. Zdepski, W. Kwok and D. Raychaudhuri, "Error concealment algorithms for robust decoding of MPEG compressed video," Signal processing: Image Communication, vol. 10, no. 4, pp. 249–268, 1997.

[18] M.J. Chen, L.G. Chen, and R.M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 7, no. 3, pp. 560–563, 1997.

[19] L. Atzori, F.G. De Natale, and C. Perra, "A Spatio-Temporal concealment technique using boundary matching algorithm and mesh-based warping (BMA-MBW)," IEEE Trans. on Multimedia, vol. 3, no. 3, pp. 326–338, 2001.

[20] Y. Chen, Y. Hu, O.S. Au, H. Li, and C.W. Chen, "Video error concealment using spatio-temporal boundary matching and partial differential equation," IEEE Trans. on Multimedia, vol.10, no. 1, pp. 2–15, 2008.

[21] L. Larzon, M. Degermark, M. Pink, and S. Degermark, "UDP lite for real time multimedia applications," IEEE Int. Conf. of Communications Technical Report In Proceedings of the QoS mini-conference, 1999.

[22] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," Int. Conf. on Advances in Mobile Computing and Multimedia, Yogyakarta, Indonesia, pp. 49–58, 2006.

[23] L. Trudeau, S. Coulombe, and S. Pigeon, "Pixel domain referenceless visual degradation detection and error concealment for mobile video," IEEE 18th Int. Conf. on Image Processing, pp. 2229–2232, 2011.

[24] Y. Wang and S. Yu, "Joint source-channel decoding for H.264 coded video stream," IEEE Trans. on Consumer Electronics, vol. 51, no. 4, pp. 1273–1276, 2005.

[25] H. Nguyen and P. Duhamel, "Iterative joint source-channel decoding of variable length encoded sequences exploiting source semantics," IEEE Int. Conf. on Image Processing, vol. 5, pp. 3221–3224, 2004.

[26] C. Weidmann, P. Kadlec, O. Nemethova, and A.A. Moghrabi, "Combined sequential decoding and error concealment of H.264 Video," IEEE 6th Workshop on Multimedia Signal, pp. 299–302, October 2004.

[27] C. Bergeron and C. Lamy-Bergot, "Soft-input decoding of variable-length codes applied to the H.264 standard," IEEE 6th Workshop on Multimedia Signal Processing, pp. 87–90, 2004.

[28] G. Sabeva, S. Ben Jamaa, M. Kieffer, and P. Duhamel, "Robust decoding of H.264 encoded video transmitted over wireless channels," IEEE 8th Workshop on Multimedia Signal Processing, pp. 9–13, 2006.

[29] D. Levine, W. E. Lynch, and T. Le-Ngoc, "Iterative joint source-channel Decoding of H.264 compressed video," IEEE Int. Symposium on Circuits and Systems, pp. 1517–1520, 2007.

[30] N. Nguyen, W.E. Lynch, and T. Le-Ngoc, "Iterative joint source-channel decoding for H.264 video transmission using virtual checking method at source decoder," IEEE 23rd Canadian Conf. on Electrical and Computer Engineering, pp. 1–4, 2010.

[31] R.A. Farrugia and C.J. Debono, "A Hybrid Error Control and Artifact Detection Mechanism for Robust Decoding of H.264/AVC Video Sequences," IEEE Trans. on Circuits and Systems for Video Technology, vol. 20, no. 5, pp. 756–762, 2010.

[32] R.A. Farrugia and C.J. Debono, "Robust decoder-based error control strategy for recovery of H.264/AVC video content," IET Communications, vol. 5, no. 11, pp. 1928–1938, 2011.

[33] F. Caron and S. Coulombe, "A maximum likelihood approach to video error correction applied to H.264 decoding," IEEE 6th Int. Conf. on Next Generation Mobile Applications, Services, and Technologies, pp. 1–6, 2012.

[34] F. Caron and S. Coulombe, "A maximum likelihood approach to correct transmission errors for joint source-channel decoding of H.264 coded video," IEEE 20th Int. Conf. on Image Proc., pp. 1870–1874, 2013.

[35] F. Caron, and S. Coulombe, "Video error correction using soft-output maximum likelihood decoding applied to H.264 baseline profile," to appear in IEEE Trans. on Circuits and Systems for Video Technology.

[36] Joint Video Team. H.264/AVC JM reference software. http://iphome.hhi.de/suehring/tml/, November 2011.

[37] https://media.xiph.org/video/derf/, Last accessed July 15th 2014.