EZBC video streaming with channel coding and error concealment*

Ivan V. Bajić and John W. Woods
Center for Next Generation Video and ECSE Department
Rensselaer Polytechnic Institute, Troy, NY 12180-3590
ivanb@cipr.rpi.edu, woods@ecse.rpi.edu

ABSTRACT

In this text we present a system for streaming video content encoded using the motion-compensated Embedded Zero Block Coder (EZBC). The system incorporates unequal loss protection in the form of multiple description FEC (MD-FEC) coding, which provides adequate protection for the embedded video bitstream when the loss process is not very bursty. The adverse effects of burst losses are reduced using a novel motion-compensated error concealmet method.

Keywords: video streaming, error concealment, multiple descriptions, EZBC

1. INTRODUCTION

Internet video streaming is becoming an increasingly important way of delivering information throughout the world. There are currently more than 600 TV stations in over 100 countries[1] providing some form of streaming video content over the Internet. However, the quality of the delivered video is typically far below that provided by the conventional TV broadcast systems. Apart from the bandwidth limitations faced by many users, a fundamental problem lies in the fact that the delivery medium is not matched to the video streaming requirements: the Internet was designed for a reliable data transfer with potentially unbounded delay, while video streaming has strict delay requirements, but may tolerate some loss.

The Internet is a shared best-effort network, which makes it is extremely difficult to predict when a particular link will become congested. If routers could change the routing tables fast enough, congested links might be avoided on-the-fly. However, routing tables are relatively stable and change over time intervals on the order of tens of seconds, making congestion events unavoidable. During congestion, a packet stream typically loses several packets in sequence, i.e. suffers a burst loss. Video streams are particularly sensitive to burst losses. They typically have high bit rate, which means that during a congestion event many packets from a video stream might be lost. On the other hand, due to their real-time requirements, it may not be feasible to retransmit the lost packets. Hence, one must find alternative ways to deal with burst losses in video streaming.

Reducing burst losses in video streaming was recently addressed in refs.[2] and.[3] In both cases the authors propose the use of multiple servers to deal with the problem of burst losses, by exploiting path diversity. In this work we propose a much simpler and cost-efficient way, based on error concealment, to improve the video quality in the case of burst losses. It was demonstrated in ref.[4] that FEC and error concealment can be successfully combined in a concatenated manner to improve the performance of a video communication system.

This paper presents a robust video streaming system based on Embedded Zero-Block Coding (EZBC).[5] The system incorporates unequal loss protection (ULP) and motion compensated (MC) error concealment. The system block diagram is shown in Figure 1. The ULP is provided through multiple description FEC (MD-FEC)[6,7] coding. An EZBC video bitstream composed of motion vector bits and subband/wavelet coefficient bits is passed through the MD-FEC module prior to transmission. The MD-FEC module assigns appropriate erasure error protection to the bitstream, taking into account its operational rate-distortion characteristics, desired number of packets, maximum allowed transmission rate, and other channel parameters. Video streaming systems based on MD-FEC work well when the channel conditions vary slowly. However, rapid changes of

* This work was supported in part by the ARO grant DAAD 19-00-1-0559.

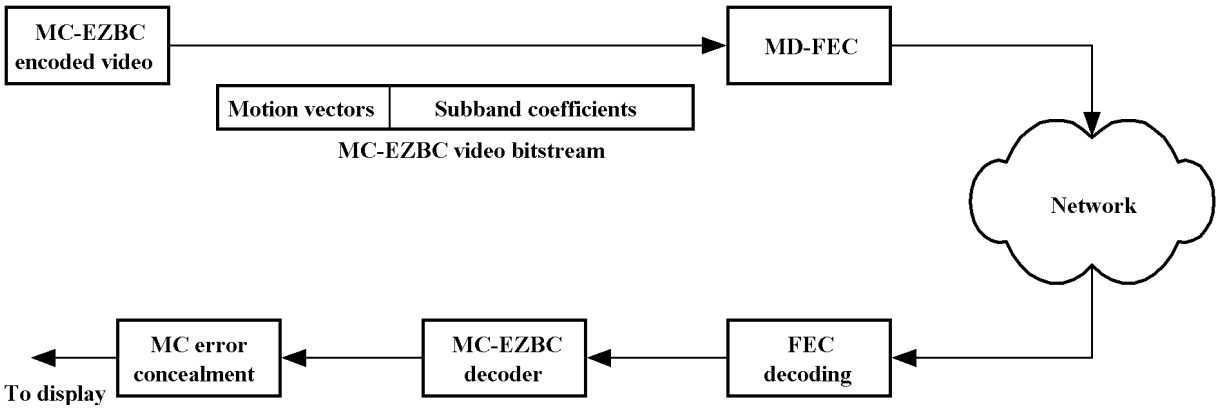

Figure 1: Illustration of a video streaming system based on the EZBC video coder.

channel parameters (e.g. in the case of burst losses) can cause significant degradation of video quality. Error concealment is used to improve received video quality in such situations. Observe that motion vectors are placed at the start of the bitstream so they receive most protection. Hence, motion vectors are the portion of the bitstream which is most likely to be received. This fact is utilized by the proposed MC error concealment method described later in the text.

The remainder of the paper describes the components of the proposed video streaming system. In Section 2 we provide a brief description of the EZBC video coder. In Section 3 we review the MD-FEC coding strategy and propose an alternative objective function to be used in FEC assignment. In Section 4 we examine the effects of burst losses on received video. In Section 5 we describe the proposed motion-compensated error concealment strategy to deal with burst losses. Results and conclusions are presented in Section 6.

2. MOTION-COMPENSATED EZBC VIDEO CODING

Embedded Zero-Block Coder (EZBC)[5] is a state-of-the-art motion-compensated subband/wavelet video coder. It produces embedded bitstreams supporting a full range of scalabilities (SNR, spatial, and temporal). Its block diagram is shown in Figure 2. Input video is subject to motion estimation and the resulting motion vectors are used for motion-compensated temporal analysis. The output of the MC temporal analysis block is the set of temporal low and high frequency subbands. They are illustrated in Figure 3, which shows a typical Group-Of-Pictures (GOP) structure of this coder. The top level represents the video at full frame rate. Neighboring frames are decomposed using a motion-compensated Haar filter bank to produce the temporal low frequency bands (solid lines) and temporal high frequency bands (dashed lines) at the next lower level. Motion vectors are shown as arrows. Low temporal frequency bands are effectively the MC averages of two neighboring frames at full frame rate, and they occur at half the frame rate. The process is repeated until we obtain the MC average of all 16 frames in the GOP, which is at the bottom of the temporal pyramid. Video data in this case has five temporal scalability layers, labeled (1) through (5) in the figure. The video at 1/16 of the full frame rate can be reconstructed from layer (1), at 1/8 of the full frame rate from layers (1) and (2), and so on. Temporal subbands are then subject to spatial subband/wavelet analysis and encoded using the 3-D version of the EZBC coding algorithm, details of which are given in ref.[5] The bitstreams produced by the motion vector (MV) encoder and EZBC are stored in the buffer for scaling according to the user requirements.

3. UNEQUAL LOSS PROTECTION

In this section we review the MD-FEC coding paradigm[6,7] as a way of providing unequal loss protection for embedded bitstreams. We also propose an alternative objective function to be used in optimizing the FEC assignment. Our notation is similar to that of ref.[7] Given a total transmission rate limit $R_{\max}$ and the desired number of descriptions (packets) N, the bitstream is divided into N sections $[R_{k-1}+1, R_k], k = 0, 1, ..., N$, with

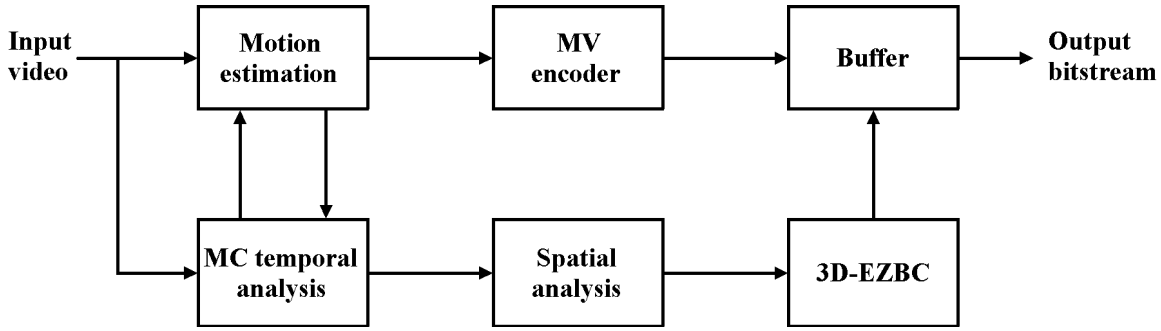

Figure 2: Block diagram of the motion-compensated EZBC video coder.

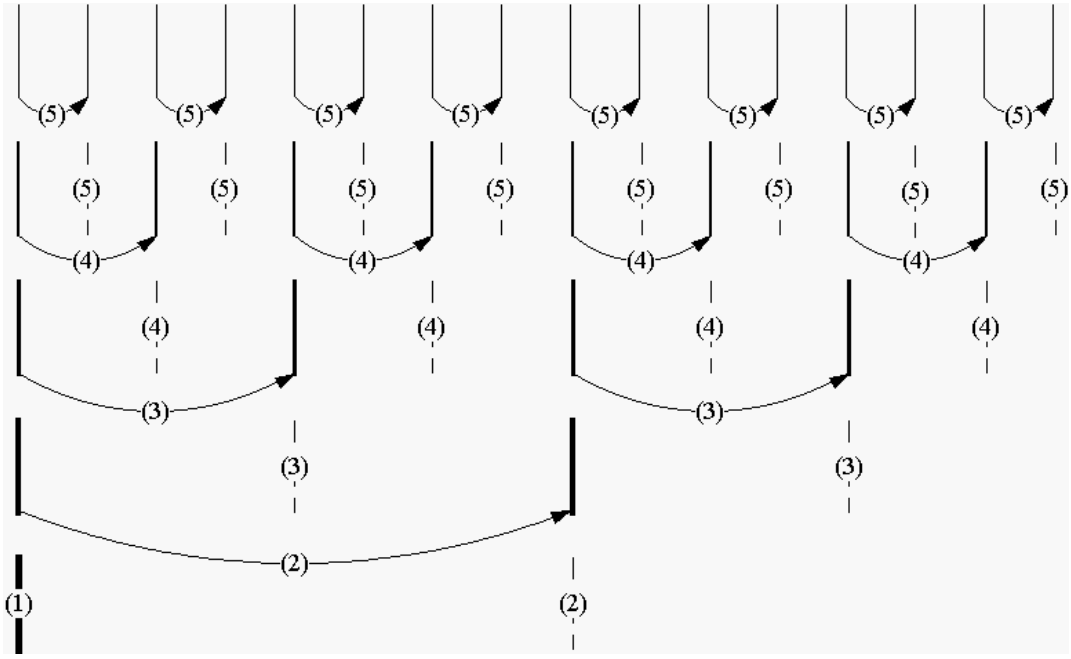

Figure 3: A typical GOP of 16 frames with 5 layers of temporal scalability.

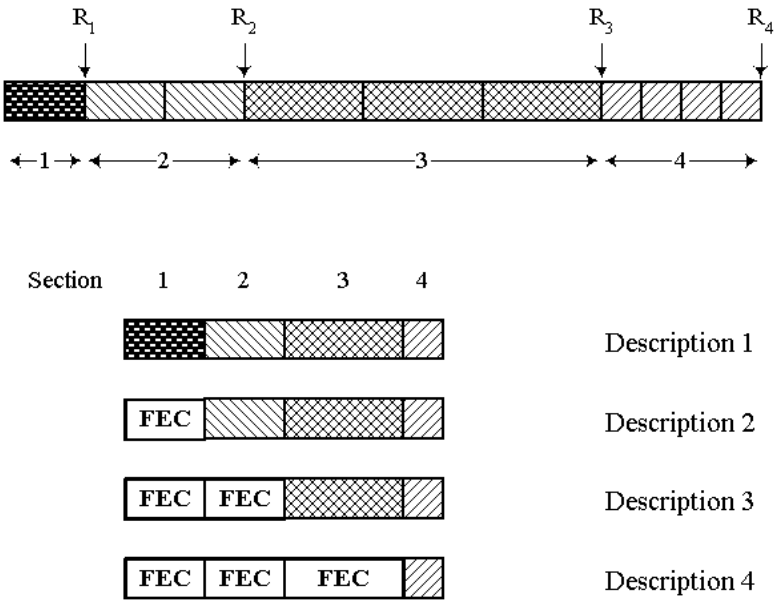

Figure 4: Illustration of MD-FEC coding of embedded bitstreams for $N = 4$ descriptions.

$R_0 = 0$ bits. The k-th section is further split into k subsections of equal size, and protected by a systematic (N, k) Reed-Solomon code. An illustration is given in Figure 4 for $N = 4$ packets. The "FEC" in the figure stands for the parity symbols of RS codewords. If n packets are received, decoding is guaranteed up to R_n.

Let $D(R)$ be the operational distortion-rate function for a given coder and a specified source signal, and q_j be the probability that j packets are received. To find the optimal FEC assignment, the authors in ref.[7] proposed solving the following problem: find $\mathbf{R} = (R_1, R_2, ..., R_N)$ which minimizes

$$f(\mathbf{R}) = \sum_{j=0}^{N} q_j D(R_j), \tag{1}$$

subject to $0 \leq R_1 \leq R_2 \leq ... \leq R_N$ and total rate $R_{total} \leq R_{\max}$. They called the function f from (1) the "expected distortion." This would indeed be the expected distortion at the receiver if we always decoded only up to the guaranteed point in the bitstream, e.g. up to R_j for j received packets. However, we can often do better than that. For example, referring to Figure 4, if packets 1 and 2 are received, and packets 3 and 4 are lost, we can decode not only up to R_2, but up to $R_2 + 2(R_3 - R_2)/3$, since this is the actual point where the variable length code (VLC) breaks. In general, if j packets are received and m is the lowest index among the lost packets' indices, then the bitstream is decodable up to $R_j + (R_{j+1} - R_j)(m-1)/(j+1)$. Let $\pi_{j,m}$ be the probability of the event that j packets are received and m is the lowest index among the lost packets' indices. The expected distortion at the receiver is

$$E[D(\mathbf{R})] = \sum_{j=0}^{N-1} \sum_{m=1}^{j+1} \pi_{j,m} D\left(R_j + (R_{j+1} - R_j)\frac{m-1}{j+1}\right) + q_N D(R_N). \tag{2}$$

If, given the number of received packets, any combination of received packets is equally likely, then the above expression can be rewritten as follows. Assume j is the number of received packets, hence $N - j$ is the number of lost packets. The total number of ways to choose $N - j$ lost packets out of a total of N packets is $\binom{N}{N-j}$. If m is the lowest index among the lost packets' indices, then the remaining $N - j - 1$ packets must have indices

Loss probability p	0	0.0625	0.125	0.1875	0.25	0.3125	0.375	0.4375	0.5
Avg. PSNR for (1)	36.8	35.7	34.4	33.5	32.8	31.9	31.6	31.2	29.8
Avg. PSNR for (2)	36.8	35.7	34.4	33.7	33.1	32.5	31.9	31.4	30.3
Gain	0.0	0.0	0.0	+0.2	+0.3	+0.6	+0.3	+0.2	+0.5

Table 1: Comparison of average PSNR in dB for two different objective functions

in $\{m+1, m+2, ..., N\}$. Since this set has $N-m$ elements, the number of ways in which m can be the lowest index is $\binom{N-m}{N-j-1}$. Hence, in this case we have $\pi_{j,m} = q_j\binom{N-m}{N-j-1}/\binom{N}{N-j}$, so

$$E[D(\mathbf{R})] = \sum_{j=0}^{N-1} q_j \sum_{m=1}^{j+1} \frac{\binom{N-m}{N-j-1}}{\binom{N}{N-j}} D\left(R_j + (R_{j+1} - R_j)\frac{m-1}{j+1}\right) + q_N D(R_N). \tag{3}$$

If our goal is to have the minimum expected distortion at the receiver, we should use (2) as the objective function to be minimized in FEC assignment. Now let $\mathbf{R}^{(1)}$ be the vector of optimal rates obtained by minimizing (1) and $\mathbf{R}^{(2)}$ be the vector of optimal rates obtained by minimizing (2). Since the minimum of $E[D(\mathbf{R})]$ is achieved at $\mathbf{R}^{(2)}$, we obviously have $E\left[D\left(\mathbf{R}^{(2)}\right)\right] \leq E\left[D\left(\mathbf{R}^{(1)}\right)\right]$. In other words, the solution obtained by minimizing (2) will, on average, offer at least as good a performance as the solution obtained by minimizing (1). The question is whether the performance gain (if any) obtained in this way justifies the use of a more complex objective function. The following example shows that extra complexity may be justified.

We performed a set of experiments on a 512×512 greyscale *Lena* image, with $R_{\max} = 0.43$ bpp and $N = 16$ packets. The image coder used in this example is 2-D EZBC.[8] Transmission over the independent loss channel was assumed and loss probability p was varied in the range $0 \leq p \leq 0.5$. For this channel $q_j = \binom{N}{j} p^{N-j}(1-p)^j$. It was assumed that the exact loss probability is known at the source. The FEC assignment was made in one case using the objective function from (1) and in another case using (2) which, for the independent loss channel, reduces to (3). The PSNR averaged over all possible combinations of lost packets was measured. The results are shown in Table 1. The first row of the table shows the loss probability, while the second and third rows show the average PSNR obtained by using (1) and (2), respectively, as the objective function, and the fourth row shows the PSNR gain. The results indicate that some gain may be obtained by using (2) instead of (1) as the objective function in FEC assignment. In this example, there is no gain for low loss probability, but at higher values of loss probability the gain reaches 0.6 dB. The next section examines the effects of burst losses on the received video.

4. EFFECTS OF BURST LOSSES ON FEC-PROTECTED VIDEO BITSTREAMS

The experiments reported in this section are based on the *Mobile Calendar* sequence (96 frames, SIF resolution, 30 fps). The sequence was encoded with a GOP size of 16 frames. The MD-FEC parameters were $N = 64$ packets, $R_{\max} = 1$ Mbps, and an expected random loss of 10%. The FEC was assigned for each GOP separately, assuming independent packet losses. Two simulations of network transmission were carried out using a Gilbert model, which was found to represent the packet loss model reasonably well.[9] The model is specified in terms of the loss probability P_B and average burst length L_B. In[9] the authors found the typical values of P_B to be between 0 and 0.6, while for L_B they are between 2 and 20. In our first simulation we used $(P_B, L_B) = (0.1, 2)$ and in the second one we used $(P_B, L_B) = (0.1, 5)$. The actual average packet loss was 9.4% in both cases, slightly less than the assumed loss of 10%. These two cases are aimed at illustrating realistic scenarios where the channel model may be mismatched to the actual channel. In the first case the bursts are short and the actual channel resembles the independent loss channel assumed in the FEC assignment. In the second case the bursts are longer and the mismatch is more significant.

Average PSNR was 26.4 dB in the first case, and 24.0 dB in the second case. The PSNR results for the two cases are shown in Figure 5. The top part shows the frame-by-frame PSNR of decoded video. The bottom part shows the average loss per GOP. The horizontal line in the bottom part indicates the expected loss of 10%. As the results show, the PSNR performance is more consistent in the first case. The losses were sufficiently

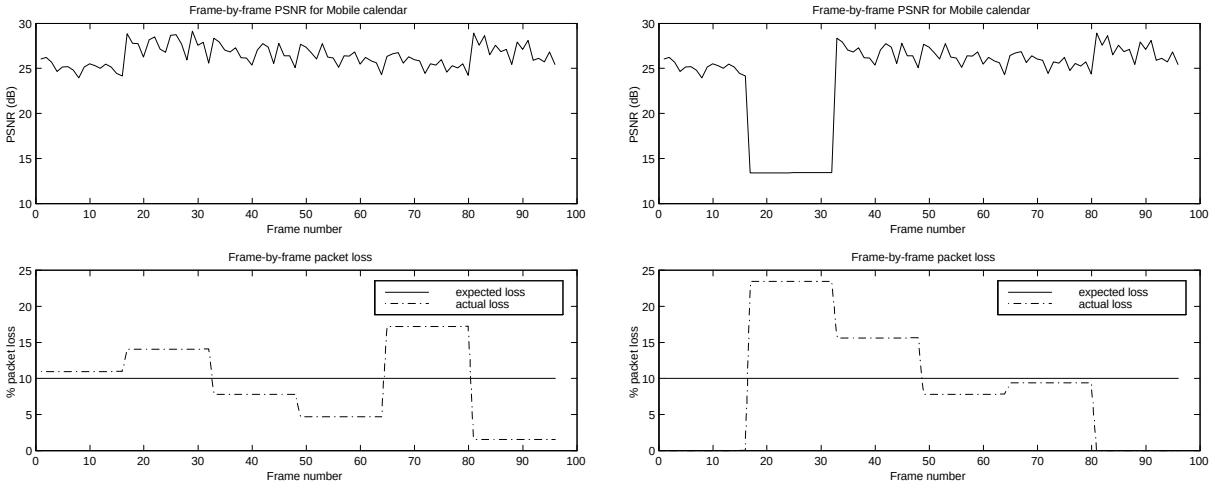

Figure 5: Left: average loss 9.4 %, average burst length = 2. Right: average loss 9.4 %, average burst length = 5.

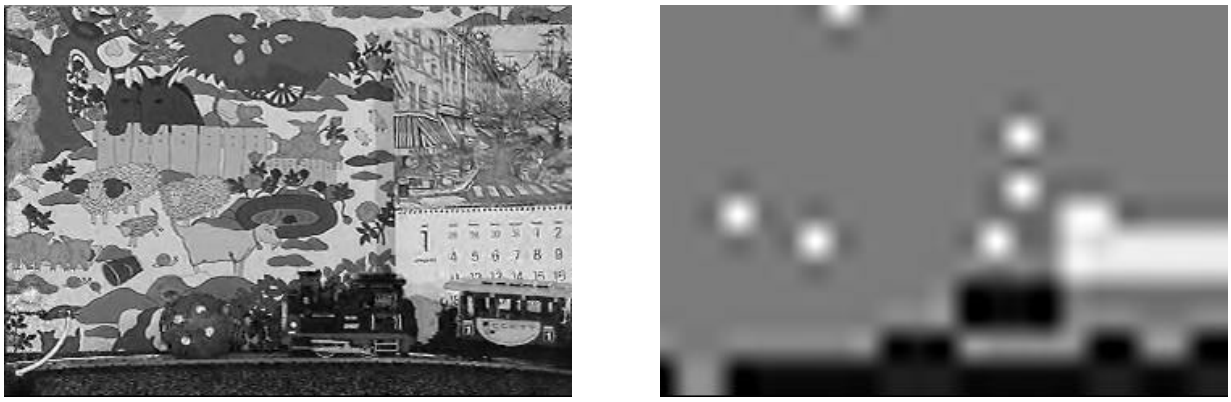

Figure 6. Illustration of visual quality degradation in case of burst losses. Left: frame 16, PSNR = 24.1 dB. Right: frame 17, PSNR = 13.4 dB.

spread out and did not deviate much from the assumed value, so FEC was powerful enough to reconstruct the missing data packets in all GOPs. In the second case, however, the second GOP (frames $17-32$) suffered a high loss of nearly 25%, causing a drop in PSNR of more than 10 dB. Similar sudden drops in quality for an MD-FEC scheme were observed in ref.[10] An illustration of visual quality degradation caused by burst losses is given in Figure 6, where we show frame 16 (first GOP in Figure 5 right) and frame 17 (second GOP in Figure 5 right) which suffered high loss. In frame 17, the decoder is only able to decode the motion vectors and the first few bitplanes. Black and white blurry spots in the figure indicate the locations of the coefficients with largest magnitudes. In the following section we propose a motion-compensated error concealment method which can significantly improve video quality in the case of such burst losses.

5. MOTION-COMPENSATED ERROR CONCEALMENT

In this section we propose a simple MC error concealment strategy which is appropriate for our MD-FEC framework using EZBC. The error concealment method proposed here takes advantage of ULP and the properties of the motion-compensated temporal filter bank to estimate the frames which cannot be recovered by FEC.

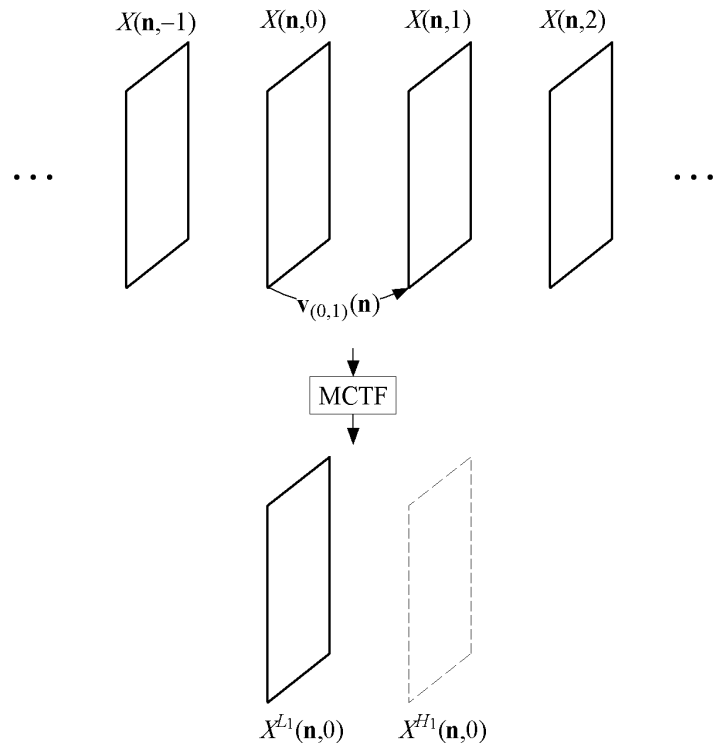

Figure 7: One level of motion-compensated temporal filtering.

Let $X(\mathbf{n}, t)$ denote the pixel value at spatial location $\mathbf{n} = (n_1, n_2)$ and time t in the input video signal. Consider one level of motion-compensated Haar temporal filtering illustrated in Figure 7. Let $t \in \{0, 1\}$ be the time indices of the frames at the input to the filter bank. The MC temporal filtering produces a low temporal frequency band $X^{L_1}(\mathbf{n}, 0)$, and a high temporal frequency band $X^{H_1}(\mathbf{n}, 0)$:

$$\begin{aligned} X^{L_1}(\mathbf{n},0) &= \frac{1}{\sqrt{2}}[X(\mathbf{n},0) + X(\mathbf{n}-\mathbf{v}_{(0,1)}(\mathbf{n}),1)], \\ X^{H_1}(\mathbf{n},0) &= \frac{1}{\sqrt{2}}[-X(\mathbf{n}+\mathbf{v}_{(0,1)}(\mathbf{n}),0) + X(\mathbf{n},1)], \end{aligned} \tag{4}$$

where $\mathbf{v}_{(0,1)}(\mathbf{n})$ is the estimate of the motion vector field between frames $X(\mathbf{n}, 0)$ and $X(\mathbf{n}, 1)$. If the motion vector estimates were perfectly accurate, we would have $X(\mathbf{n}, 0) = X(\mathbf{n} - \mathbf{v}_{(0,1)}(\mathbf{n}), 1)$ for all $\mathbf{n}$. In practice this is not the case, but with a reasonably accurate motion field we have $X(\mathbf{n}, 0) \approx X(\mathbf{n} - \mathbf{v}_{(0,1)}(\mathbf{n}), 1)$. Hence, from the first equation in (4) we have

$$X^{L_1}(\mathbf{n},0) \approx \frac{2}{\sqrt{2}} X(\mathbf{n},0) = \sqrt{2} X(\mathbf{n},0). \tag{5}$$

This relationship is also valid for k levels of MC temporal filtering, with the scaling constant in (5) modified appropriately:

$$X^{L_k}(\mathbf{n},0) \approx 2^{k/2} X(\mathbf{n},0). \tag{6}$$

Hence, the low temporal frequency band is approximately equal to the scaled version of the first frame in the GOP.

These observations form the basis of our motion compensated error concealment. When a GOP is affected by high loss, we find the estimate $\widehat{X}(\mathbf{n}, 0)$ of the first frame of that GOP using the frames of the previous GOP.

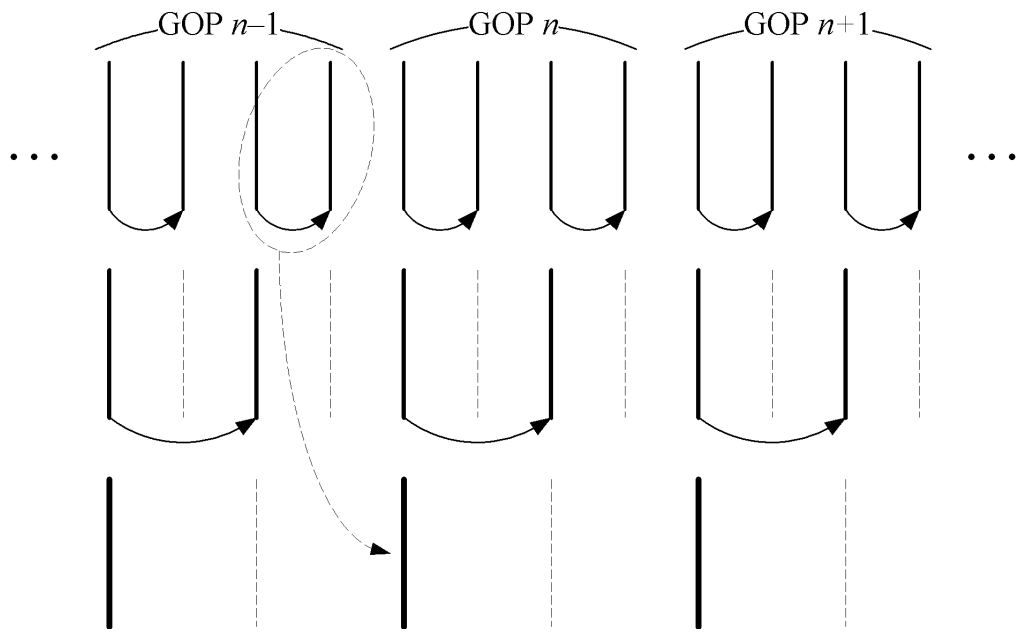

Figure 8: Illustration of MC error concealmen for GOP n.

Then we set $\widehat{X^{L_k}}(\mathbf{n}, 0) = 2^{k/2}\widehat{X}(\mathbf{n}, 0)$ and apply MC temporal synthesis using the available motion vectors to recover the frames of the corrupted GOP. This is illustrated in Figure 8. The figure shows $k = 2$ levels of MC temporal filtering, corresponding to GOP size of 4 frames, with GOP n assumed to be corrupted.

We tested two methods for obtaining $\widehat{X^{L_k}}(\mathbf{n}, 0)$:

1. "Replacement" - in this method the low temporal frequency band of the corrupted GOP is replaced by the scaled version of the last frame of the previous GOP:

$$\widehat{X^{L_k}}(\mathbf{n}, 0) = 2^{k/2}X(\mathbf{n}, -1).$$

2. "Prediction" - in this method the low temporal frequency band of the corrupted GOP is predicted from the scaled version of the last frame of the previous GOP using the last motion vector field from the previous GOP:

$$\widehat{X^{L_k}}(\mathbf{n}, 0) = 2^{k/2}X(\mathbf{n} + \mathbf{v}_{(-2,-1)}(\mathbf{n}), -1).$$

Results are given in the following section.

6. RESULTS AND CONCLUSIONS

The two versions ("replacement" and "prediction") of the MC error concealment algorithm were tested on the *Mobile Calendar* sequence from Section 4, and the *Foreman* sequence (96 frames, CIF resolution, 30 fps). The *Foreman* sequence was encoded using the same parameters as *Mobile Calendar*: $R_{\max} = 1$ Mbps, $N = 64$ packets and expected random loss of 10%. Its transmission was simulated over a bursty channel where a large burst occurred in the third GOP (frames $33-48$). Average PSNR results are shown in Table 2. Average PSNR is computed for the whole sequence (96 frames in both cases), as well as the corrupted segments (frames $17-32$ for *Mobile Calendar* and frames $33-48$ for *Foreman*). The frame-by-frame PSNR is shown in Figures 9 and 10. Visual improvement brought by the error concealment is illustrated in Figure 11 with a few frames from

Sequence	No concealment	"Replacement"	"Prediction"
Mobile Calendar (whole sequence)	24.0	24.8 (+0.8)	25.0 (+1.0)
Mobile Calendar (corrupted segment)	13.4	17.9 (+4.5)	19.5 (+6.1)
Foreman (whole sequece)	33.0	34.8 (+1.8)	35.1 (+2.1)
Foreman (corrupted segment)	14.7	25.5 (+10.8)	27.4 (+12.7)

Table 2: PSNR results in dB for MC error concealment

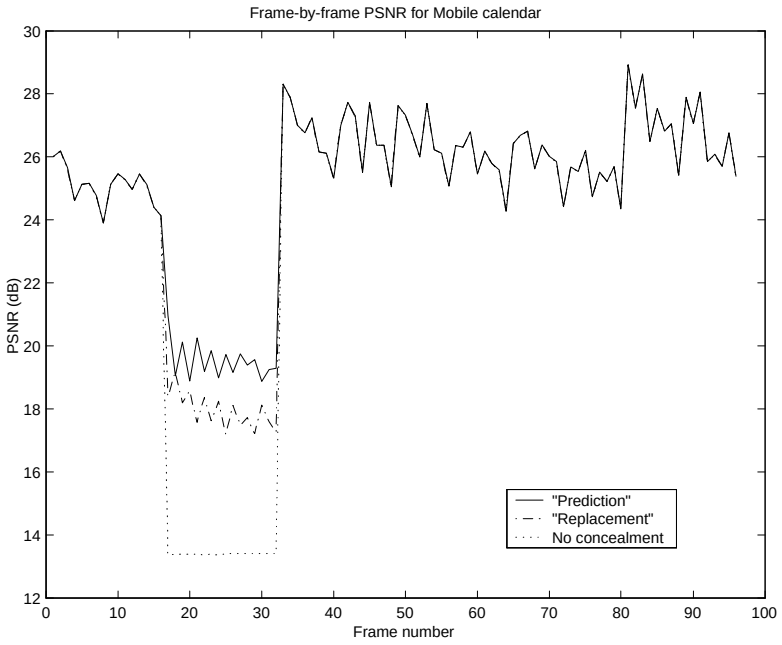

Figure 9: PSNR results for *Mobile Calendar*.

the *Mobile Calendar* sequence. The frame number is indicated next to each frame. Sample video clips may be found online.[11]

As indicated by the results, the gains achieved by error concealment are significant, both visually and in terms of PSNR. A major issue in the proposed MC error concealment is the spatial shift of moving objects between the current GOP which is being concealed and the previous GOP. This shift is being ignored in the "replacement" version of the algorithm. The "prediction" version partially solves this problem by assuming that the motion field between the previous GOP and the current GOP is the same as the motion field between the last two frames of the previous GOP. For this reason the "prediction" version of the algorithm outperforms the "replacement" version by about $1.5 - 2$ dB. But even with the "prediction" version the objects may be offset from the correct positions. The largest errors occur in the vicinity of moving edges. Hence, on a sequence with many moving edges, such as *Mobile Calendar*, the proposed concealment algorithm is expected to show worse PSNR performance than on a sequence with fewer moving edges, such as *Foreman*. This is confirmed by the results. Overall, it seems that the frames produced by error concealment visually look much better than the PSNR would suggest. Future work could include the development of a more accurate prediction of the motion vector field between the current and previous GOPs. Alternatively, the motion field between the GOPs may be estimated at the encoder and sent as an overhead to help error concealment at the receiver.

In summary, MC error concealment provides a simple and efficient way for combating the effects of burst losses in packet video communications. The results indicate that it can significantly improve both the PSNR and the visual quality of corrupted frames with a very small computational overhead.

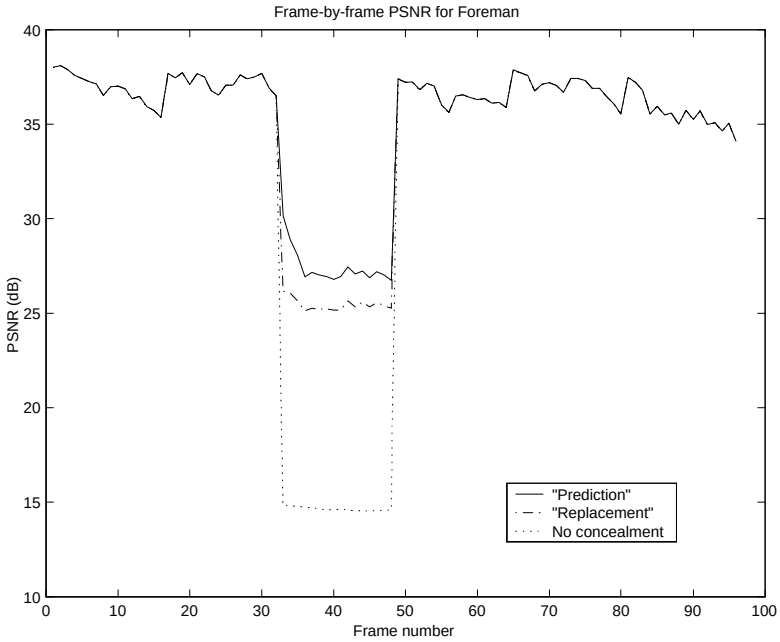

Figure 10: PSNR results for *Foreman*.

REFERENCES

1. World Wide Internet TV, http://wwitv.com
2. A. Majumdar, R. Puri, and K. Ramchandran, "Distributed multimedia transmission from multiple servers," *Proc. IEEE ICIP'02*, vol III, pp. 177-180, Rochester, NY, September 2002.
3. T. Nguyen and A. Zakhor, "Protocols for distributed video streaming," *Proc. IEEE ICIP'02*, vol III, pp. 185-188, Rochester, NY, September 2002.
4. I. V. Bajić and J. W. Woods, "Concatenated multiple description coding of frame rate scalable video," *Proc. IEEE ICIP'02*, vol. II, pp. 193-196, Rochester, NY, September 2002.
5. S.-T. Hsiang and J. W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank," *Signal Processing: Image Commun.*, vol. 16, pp. 705-724, May 2001.
6. A. E. Mohr, E. A. Riskin and R. E. Ladner, "Graceful degradation over packet erasure channels through forward error correction," *Proc. IEEE DCC'99*, pp. 92-101, Snowbird, UT, March 1999.
7. R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," *Proc. 33rd Asilomar Conf. on Signals, Systems and Computers*, Pacific Groove, CA, October 1999.
8. S.-T. Hsiang and J. W. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," *Proc. IEEE ISCAS'00*, vol. 3, pp. 662-665, Geneva, Switzerland, May 2000.
9. U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Commun.*, vol. 15, no. 1-2, pp. 77-94, September 1999.
10. R. Puri, K.-W. Lee, K. Ramchandran, and V. Bharghavan, "An integrated source transcoding and congestion control paradigm for video streaming in the Internet," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 18-32, March 2001.
11. http://www.cipr.rpi.edu/~ivanb/mcec_videos.htm

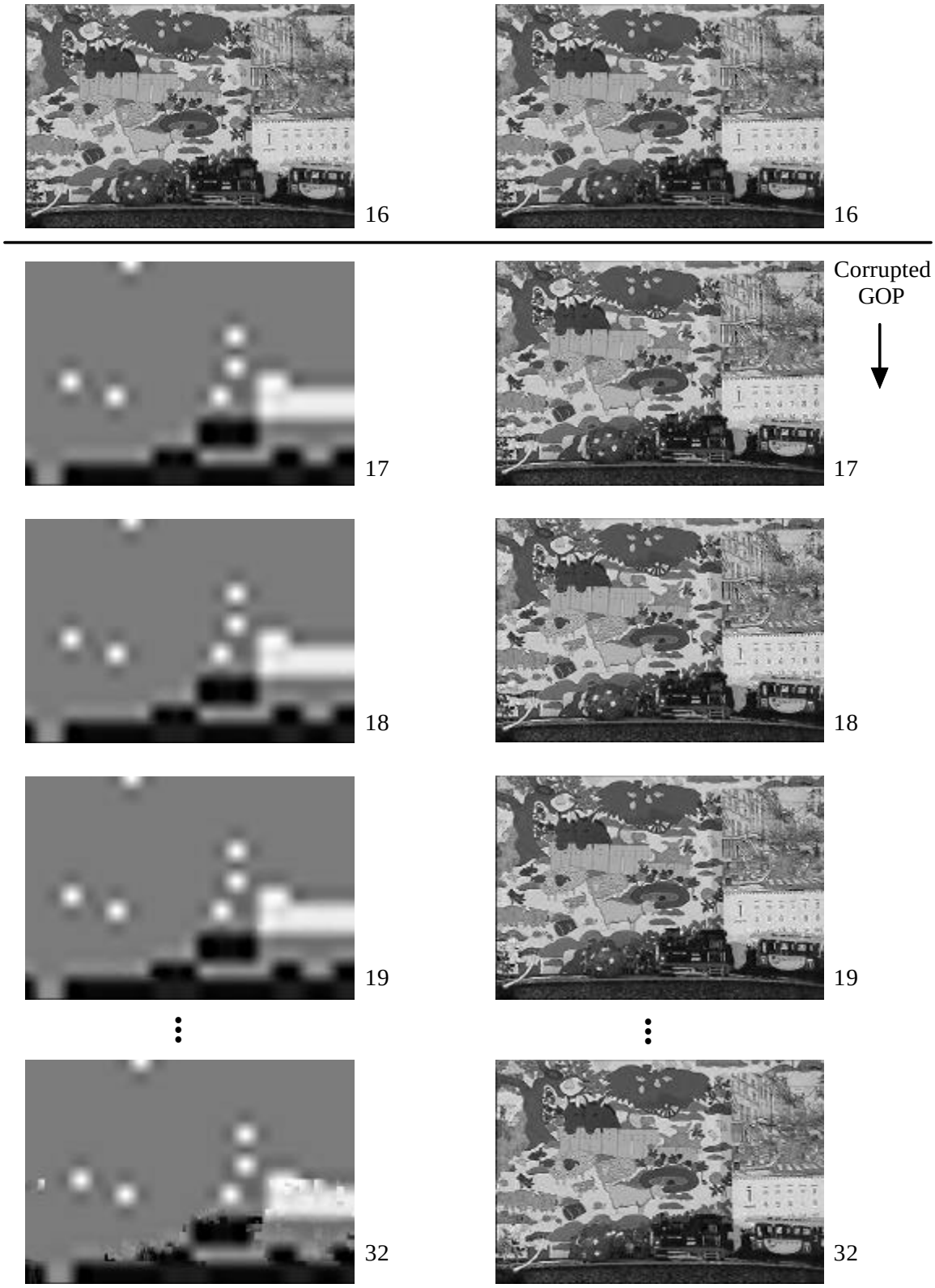

Figure 11. Visual comparison of decoded video without error concealment (left) and with error concealment by "prediction" (right).