A ROBUST ERROR RESILIENT APPROACH FOR MPEG VIDEO TRANSMISSION OVER INTERNET

Peng Yin^a, Min Wu^b and Bede Liu^a

^aDepartment of Electrical Engineering

Princeton University, Princeton, NJ 08544, U.S.A

^bDepartment of Electrical and Computer Engineering

University of Maryland and College Park, MD 20742, U.S.A

ABSTRACT

In network delivery of compressed video, packets may be lost if the channel is unreliable. Such losses tend to occur in burst. In this paper, we develop an error resilient video encoding approach to help error concealment at the decoder. We introduce a new block shuffling scheme to isolate erroneous blocks caused by packet losses. And we apply data hiding to add additional protection for motion vectors. The incorporation of these scheme adds little complexity to the standard encoder. Experimental results suggest that our approach can achieve a reasonable quality for packet loss up to 30% over a wide range of video materials.

Keywords: Error Resilient Coding, Block Shuffling, Data Hiding, Error Concealment

1. INTRODUCTION

Due to the congestion in the transmission of video over the Internet, packet may be lost and the loss tends to occur in burst.¹ This makes the use of error resilience and error concealment techniques necessary. For a channel with bursty errors such as the Internet, forward error concealment may not be appropriate.² Most error concealment approaches using postprocessing at the decoder recover the corrupted areas based on the correlation of the neighboring data. Such approaches are not effective against packet loss which often results in damaging a large contiguous region. The error resilient tools proposed in MPEG-4,³ such as resynchronization, data partitioning, reversible VLC coding and intra-refresh methods, are designed to combat random bit errors, so they are not very effective to packet loss, because packet loss can be viewed as highly bursty error with known position.

In this paper, we propose a robust error resilient approach to improve the performance of error concealment at the decoder for the video stream transport over the unreliable channels such as the Internet. We discuss our approach in terms of MPEG coded video, for which each DCT block is 8×8 and a macroblock is 16×16 .

Most of the error concealment methods for MPEG video fall into two categories: spatial and temporal error concealment. Spatial approaches has higher complexity, but produce better visual quality. So it is often applied to *I* frame while temporal concealment is applied to *P* and *B* frames. Both approaches desire the damaged blocks to be isolated. In this paper, we shall introduce a new block-shuffling scheme, so that a packet loss only causes errors of isolated blocks. Another concern is that in temporal error concealment, motion vector recovery is not easy for high activity videos. We shall propose a scheme to embed interframe-wise parity bits of motion vectors in the successive intraframe(or interframes) to help the exact recovery of motion vectors. The incorporation of these two features add little overhead.

We shall use MPEG-1 for our discussion. The extension to other block-based motion compensated video coding schemes is straightforward. We shall only consider packet loss, because the IP layer of Internet will discard any packet with random bit errors. And each packet contains one row of macroblocks with resynchronization mark. A block shuffling scheme is presented in Section 2. The use of data hiding to protect motion vector is discussed in Section 3. The overall scheme is presented in Section 4, followed by experimental results and conclusion.

This work is partly supported by a NJ State R&D Excellence Grant. Contact info: {pengyin,liu}@ee.princeton.edu, minwu@eng.umd.edu

Figure 1. Use 8, 4, or 2 correctly received neighboring blocks to recover the lost block: White blocks are correctly received blocks, dark block is lost block.

2. BLOCK SHUFFLING

2.1. Proposed Scheme

When using error concealment at decoder, it is desirable that the damaged block is surrounded by correctly received blocks, as illustrated in Figure 1. To increase the likelihood that the lost block is surrounded by correctly received neighboring blocks, we propose to shuffle the blocks before entropy coding and shuffle them back after entropy decoding. Ideally, the shuffling of the blocks should cause the damaged blocks to be separated as far as possible. In doing so, the error concealment can be more effective. In addition, the visual artifacts are less noticeable as the errors are not localized. Another side advantage of block shuffling is that concentrated errors are more likely to propagate to the following dependent frames by motion compensation, so by shuffling blocks, we can spread errors thus reducing the effect of error propagation. However, other issues such as delay and effect on compression must be taken into account. A well designed shuffling pattern should keep a good balance among these factors.

The “random block shuffling”⁴ requires that the erroneous blocks should be less than 4% to be effective to de-localize errors, making it unsuitable for Internet application. The “even-odd block interleaving” or “two-way block interleaving(2BI)” approach⁵ is shown in Figure 2(a). All blocks are labelled as “1” or “2”. All “1” blocks in each row are encoded in order, followed by the “1” blocks in the next row, and so forth. After all the “1” blocks are encoded, then all “2” blocks are encoded in the same way. Each “1” block is surrounded by 4 “2” blocks, and each “2” block is surrounded by 4 “1” blocks. So this shuffle can handle lost blocks up to 50%. Lost blocks, however, are localized.

The new shuffling pattern proposed in this paper disperses the errors over a wide area of the image, thus lowering the probability that the connected important blocks are lost at the same time. We shall call this shuffling pattern “Error Spreading Shuffling(*ESS*)”, which can more effectively spread errors and minimize the overhead in compression. The basic idea is that each transmitted packet will include blocks running through all the columns and rows of blocks, and that every column and row of blocks will have the same probability to be lost. One such pattern is shown in Figure 2(b). This pattern can be generated as follows. Suppose the image has N rows and M columns of blocks/macroblocks, and assume $M > N$. We label the (i, i) blocks as “1” starting with the top left corner. After the N “1” blocks, we label as “2” all the $(i, i + N)$ blocks, starting from the $(1, N + 1)$ block. When the labelling reaches the right extreme of the image, it wraps around to the left side. When the label “ k ” reaches the bottom row, say the (N, j) block, a new label “ $k + 1$ ” begins with the $(1, j + 1)$ block. If the $(1, j + 1)$ block has already labelled, then “ $k + 1$ ” is assigned to the $(1, j')$ block, where j' is the median of $j + 1$ and the column index of its greater nearest labelled neighbor. If it also has been labelled, then assign “ $k + 1$ ” to the $(1, j'')$, where j'' is the median of $j + 2$ and the column index of its greater nearest labelled neighbor. Continue with this manner until all the blocks have been labelled. This kind of shuffling can isolate error blocks under bursty packet loss.

We have noticed that fixed shuffling pattern can cause artifacts in fixed pattern, which can be annoyed for long video streams. In practice, mixing shuffling pattern of different direction can be employed to reduce the artifacts.

2.2. Source and Channel Coding

Block Shuffling in source coding can be analogue to bit (block) interleaving in channel coding, which aims to break bursty error into random bit errors. The difference lies at decoder: source decoder makes use of the remaining natural

1 2 1 2 1 2 1 2	1 8 3 5 2 7 4 6
2 1 2 1 2 1 2 1	6 1 8 3 5 2 7 4
1 2 1 2 1 2 1 2	4 6 1 8 3 5 2 7
2 1 2 1 2 1 2 1	7 4 6 1 8 3 5 2

(a) *2BI* (b) *ESS*

Figure 2. Several shuffling patterns

redundancy to recover random block error, while channel decoder uses explicit Forward Error Correction (FEC) code to recover the random bit error. Because the structure of MPEG video usually leads bit error to block error, and there is always some redundancy left in the coded video, source block shuffling can be more effective than channel interleaving in dealing with burst error. In wireless video transmission, while both random bit error and bursty error are severe, those two approaches can be combined.

3. DATA HIDING TO PROTECT MOTION VECTORS

Motion vectors are essential for temporal error concealment, thus many error concealment methods have been discussed to recover the corrupted motion vectors,² but they almost all fail if the corrupted region has little correlation with its neighbors. Forward error correction can restore the exact motions, but to be used against burst errors, it requires high redundant correction code,⁶ which may in turn incur more loss of packets. In this paper, we propose to use data hiding to help recover motion vectors by embedding explicit error correction bits.

Data hiding has been widely used for the purpose of ownership verification and tampering detection. But in general, data hiding provides a way to convey side information that can also be used for many other purposes.⁹ In this work, we use data hiding to help recover motion vectors. The alternative way to convey side information is to attach it separately such as in user data field or header field in the packet. These two approaches can be analogue to introducing explicit redundancy in the source coding or the channel coding. The error correction capability is the same, but the advantage of data hiding is its capability to convey the information in a computationally efficient manner while preserving the standard compliant appearance.⁹ In addition, data hiding can easily keep the total bit rates, while for attaching the side information in the user field (if available), we would need to transcode video to a lower rate. Such transcoding is not a trivial task because sophisticated rate control may have to be involved for a good tradeoff between bit rate and visual quality. Though data hiding sacrifices a little objective quality of video to convey side information, the perceptual quality can be maintained by applying proper imperceptible data hiding schemes. Many algorithms have been proposed in the literature for data hiding with different tradeoff between robustness and capacity. For our application, we prefer to use imperceptible data hiding scheme with higher capacity and the scheme should be able to survive video compression.

In our approach, the error correction capability is achieved via parity bits. Song, et al. proposed to insert parity bits across Group of Macroblocks (GOB) in a picture,¹⁰ which has the limitation that lost motion vectors are restricted to one GOB in a frame, making it not suitable for Internet transmission, where continuous GOBs may be lost in one frame. We propose to generate frame-wise parity bits for motion vectors across a pre-set group of interframes and embed them to the successive intraframe or interframes, depending on coding structure and delay requirement. This scheme can recover lost motion vectors in burst error environment. The use of data hiding keeps the rate unchanged. We shall call it *PMVDE* (Protect Motion Vector by Data Embedding). Since only P frames in the interframes will be used as anchor frame and are sensitive to error propagation, we will only protect motion vectors in those frames.

The generation of the parity bits is illustrated in Figure 3. In MPEG, the motion vectors are differentially Huffman coded. We arrange the coded bits of the motion vectors of each P frame row by row. If a macroblock does not have motion vector, we assign a “0” to it. One bit overhead is needed to indicate whether a macroblock

Figure 3. Protect motion vectors using frame-wise parity bits

is intra- or inter-coded. We then generated the modulo-2 sum of the rows of coded motion vectors, as indicated in Figure 3. Because the uneven lengths of rows, the shorter rows are padded with bit ‘0’. This scheme uses one parity bit to correct one bit error along each bit plane. The position of error bits can be detected by packet number. The generated parity bits are then embedded to the successive frame. The advantage of adopting the interframe-wise parity bits is their ability to protect against consecutive errors in the same frame using fewer parity bits.

The parity bits are embedded in the quantized DCT coefficients. In order to produce less noticeable artifacts, the DC and low AC coefficients are unchanged and a human visual model is used to select which coefficients are used for embedding. A shuffling is applied before the actual embedding to equalize the uneven embedding capacity of the blocks.⁷ If the embedding capacity of the video is low, we can choose to selectively embed the motion vectors or only embed the sum or average of motion vectors along one slice. And we may use more than one frame to embed one set of parity bits.

4. ERROR CONCEALMENT

Figure 4 shows the main processing flow of our approach for generating an error resilient MPEG stream from an original MPEG stream. At the encoder, the MPEG stream is entropy de-coded, and the blocks are shuffled according to Section 2. If de-entropied frame is I picture (or the first frame of a pre-set group of pictures), the frame-wise parity bits are embedded as described in Section 3. After entropy coding, the newly generated MPEG stream is packetized and sent.

The bit stream received by the decoder is first entropy decoded. The blocks are then shuffled back to their original order. If the de-entropied stream is I frame (or the first frame of a pre-set group of pictures) and there are errors in previous P frames, the embedded data is extracted. De-quantization and inverse DCT are performed as a standard decoder. Error concealment is performed after decoding the video. Notice that from packet numbers of received packets, we know exactly which packet is lost.

For I frame, Multi-Directional Interpolation(MDI) is performed.⁸ For P frames, of all slices in the same position in each P frame in a GOP , if only one slice is lost, the motion vectors of that slice can be recovered exactly assuming the corresponding slice in the following I frame (or P frame) is received correctly. Otherwise, Median Filter(MF) can be used to interpolate lost motion vectors from neighboring motion vectors. Furthermore, an overlapped motion compensation such as the one suggested in H.263 can be employed to improve the performance of temporal error concealment.

5. SIMULATION

We implemented the proposed approach in software. To simulate packet loss behavior in Internet, a simple 2 state Markov Model⁶ is used. The channel is characterized by average packet loss probability (P_l) and average burst length(L_b). The video is packetized in variable length. Each packet consists one row of macroblocks with a

Figure 4. Main processing flow: (a) encoder; (b) decoder

resynchronization marker. We have tested a wide range of videos with various frame size, encoding pattern and bit rates to show the performance of our method.

5.1. Channel Model

Internet can be fairly well modelled by a two-state Markov chain.⁶ In this model, the channel is associated with one of two states: a “good” state, 0, and a “bad” state, 1. In state 0, packets are received correctly whereas in state 1 packets are lost. The model is fully described by the transition matrix P . Let $s_n = 0$ if the channel is good during the n th time unit, and $s_n = 1$ otherwise. Then s_n is a binary Markov process with transition matrix

$$\mathbf{P} = \begin{pmatrix} P[s_n = 0|s_{n-1} = 0] & P[s_n = 1|s_{n-1} = 0] \\ P[s_n = 0|s_{n-1} = 1] & P[s_n = 1|s_{n-1} = 1] \end{pmatrix} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \quad (1)$$

assuming the Markov chain is stationary. The average loss probability

$$P_l = \frac{p_{01}}{p_{10} + p_{01}} \quad (2)$$

and the average burst length

$$L_b = \frac{1}{p_{10}}. \quad (3)$$

will be used to characterize the channel condition.

5.2. Experimental Result

We shall only show the experiment results for “football”, which is known by its high motion activity. The sequence is coded with MPEG-1 encoder of size QSIF(176×112) at 10 frames/s. The frame *GOP* is *IPPPP* and we have skipped $2B$ frames in-between (B frames do not cause error propagation). Fixed quantizer is used to test the effect of our approach on coding efficiency. The quantization scaling factor for *I*-frame is 8, for *P*-frame is 10. The number of an interframe group is 4. The motion vector is within the range of ± 15 pixels and its precision is half pixel. Intra-block is restricted to *P* frame to save the overhead. The resulting bit rate is 256 Kbps. Three simulations are carried out and the results are compared. Each simulation consists of 3 parts: shuffling scheme, spatial error concealment methods and temporal error concealment methods. We list the components of each simulation in Table 1 and the detailed description to the acronyms in the simulations in Table 2.

First, it is observed that the bit rate increases for Sim2 over Sim1 by 0.35% and for Sim3 by 1.02%. It is thus clear that our algorithm decreases the compression ratio negligibly.

Second, we show in Figure 5 the average PSNR as a function of P_l from simulations using 5 different channel conditions. We simulate a random loss case and a burst case where average burst length is 2, which is typical for the

Table 1. Simulation Schemes

	shuffling scheme	spatial error concealment methods	temporal error concealment methods
Sim1	NONE	<i>COPYING</i>	<i>COPYING</i>
Sim2	<i>2BI</i>	<i>MDI</i>	<i>MF</i>
Sim3	ESS	<i>MDI</i>	<i>PMVDE + MF</i>

Table 2. Description of Acronyms Used in Simulation

acronym	full name / description	related section
<i>2BI</i>	<u>two-way block interleaving</u>	2.1
<i>ESS</i>	<u>error spreading shuffling</u>	2.1
<i>COPYING</i>	replace the damaged block by the block in the previous frame with same spatial position	
<i>MDI</i>	<u>multi-directional interpolation</u>	4
<i>MF</i>	<u>median filter</u>	4
<i>PMVDE</i>	<u>protect motion vector by data embedding</u>	3

Internet transmission.¹ The results show that Sim3 has a PSNR gain up to 3.81 dB over Sim1 and 1.98 dB over Sim2 in random loss case, and 3.37 dB and 1.63 dB in burst case, respectively. Visual difference can be observed from Figure 6. Figure 6(a) shows the pattern of lost blocks when the 3rd packet in an *I* frame is lost. Figure 6(b) shows the reconstructed *I* frame. Figure 6(c) shows the PSNR of the reconstructed *I* frame and its 4 dependent *P* frames under the assumption that no error occurs in those *P* frames. We can see that Sim3 is much better than the other 2 methods. Spreading errors in *I* frame not only improves the error concealment quality, but also reduces the effect of error propagation to other dependent frames.

We notice that although the proposed *PMVDE* may worsen the image objective quality for *I* frame and produce drift errors for depended *P* frames, the visual degradation is hardly noticeable by taking advantage of human perceptual redundancy. (If the required quality is high, the drift errors can be eliminated by subtracting them in *P* frames.) More powerful error correction code can be applied if the successive *I* frame has enough embedding capacity.

6. CONCLUSION

In this paper, we propose an approach to improve the performance of error concealment at decoder for video transmission over Internet. The additional complexity of the approach is small and the coding efficiency is maintained. Experiments show that the approach can achieve a reasonable quality for packet loss up to 25-30%.

The proposed approach seeks ways to add redundancy in source coding. Its alternative is to add explicit redundancy in channel coding. However, our approach takes use of the structure and syntax of the video, and it is useful in combating packet losses. Further comparison and combination of these two approaches is a direction of future work.

(a) random loss

(b) $L_b = 2$

Figure 5. Average PSNR of five channel realizations using 3 different methods (Sim1, Sim2, Sim3)

(a) Pattern showing lost blocks

(b)Reconstructed I frame

(c)PSNR of Concealed anchor I frame and following P frames

Figure 6. Comparison of various methods. For (a)and (b): upper left - Original frame; upper right - Sim1; lower left - Sim2; lower right - Sim3

REFERENCES

1. Jill M. Boyce and Robert D. Gaglianella, "Packet loss effects on MPEG video sent over the public Internet," *ACM Multimedia'98*, 1998
2. Yao Wang and Qinfan Zhu, "Error control and concealment for video Communication: an overview", *Proceedings of IEEE*, vol. 86, NO. 5, May 1998.
3. ISO/IEC 14496-2:1999, "Information Technology - Coding of Audio/Visual Objects, Part 2: Visual.
4. King Ip Chan, Jianhua Lu and Justin C.-I Chuang, "Block shuffling and adaptive interleaving for still image transmission over rayleigh fading channels," *ITVT*, vol. 48, NO. 3, May 1999.
5. Qin-Fan Zhu, Yao Wang and Leonard Shaw, "Coding and cell-Loss recovery in DCT-based packet video," *CSVT*, vol. 3, NO. 3, June 1993.
6. U. Horn, K. Stuhlmuller, M. Link and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Image Communication*, Sept 1999.
7. Min Wu and Bede Liu, "Digital watermarking using shuffling," *ICIP'99*, vol. 1, pp.291-295, 1999.
8. Wenjun Zeng and Bede Liu, "Geometric-structure-based error concealment with novel applications in block-based low bit rate coding," *IEEE Transaction on CSVT*, June 1999.
9. Min Wu, "Multimedia Data Hiding," , *PH.D Thesis*, Princeton University, June, 2001.
10. Jie Song and K.J.R.Liu, "A data embedding scheme for H.263 compatible video coding," *ISCAS'99*, vol. 4, 1999.