

AN ERROR CONCEALMENT SCHEME FOR ENTIRE FRAME LOSSES FOR H.264/AVC

*Saurav K. Bandyopadhyay**

State University of New York at Buffalo
Dept. of Electrical Engineering
Buffalo, NY 14260, USA.

Zhenyu Wu, Purvin Pandit, Jill M. Boyce

Thomson - Corporate Research
2 Independence Way
Princeton, NJ 08540, USA.

ABSTRACT

In this paper, an error concealment scheme is proposed to conceal an entirely lost frame in a compressed video bitstream due to errors introduced during transmission. The proposed scheme targets low bit rate video transmission applications using H.264/AVC. The motion field of the lost frame is first reconstructed by copying the co-located motion vectors and reference indices from the last decoded reference frame. After the motion field estimation of the missing frame, motion compensation is performed to reconstruct the frame. This technique reuses existing modules of the video decoder and it does not incur extra complexity compared to decoding a normal frame. It has also been adopted as a non-normative decoder option to the JM reference software at the JVT meeting in Poznan, Poland in July 2005 [1] and has been incorporated into the SA4 video ad hoc group's toolkit at the 3GPP meeting at Paris [2] in September 2005. Simulation results will show its improved performance over other simple error concealment schemes such as "frame copy," both subjectively and objectively, without significant complexity overhead.

1. INTRODUCTION

Recently, there has been significant interest in the transmission of video content over error prone wireless networks. Video content delivered through error prone communication channels is subject to errors introduced during transmission. Transmitting compressed video bitstreams over such error-prone channels can severely affect the video content quality at the receiver. A number of techniques are available at different stages of the communication system to address this problem [3]. Among the available techniques, decoder error concealment does not require an increase in bit rate or transmission delay. It can also be applied to almost any application. Hence, it becomes a natural choice for limited bandwidth low bitrate applications.

In [4], a temporal error concealment scheme was proposed for missing macroblocks for H.264/AVC. A technique was proposed to conceal macroblocks based on the recovered motion vectors using surrounding available macroblocks. A refinement to the error concealment scheme in H.264/AVC was presented in [5]. With this scheme, a smaller block size is used to recover the MV of the missing macroblocks. However, the algorithms deal with situations where only a portion of the video frame is missing. This assumption becomes unrealistic for low bit-rate video transmission applications such as 3G wireless systems. In this type of

application, each coded video frame has a fairly small size and is usually encapsulated into a channel packet transmitted through the network. With the prevailing UDP/IP or RTP/UDP/IP protocol stacks, transmission error comes in the form of packet losses. A lost packet hence results in an entirely missing frame at the video decoder. Error concealment becomes more difficult in this case, because there are no spatial neighbors available that can be utilized to conceal the lost frame. In [6], [7], [8] several approaches were presented to address this problem.

In this paper, we propose an error concealment scheme for entire frame loss based on H.264/AVC. A mechanism is proposed to conceal reference as well as non-reference frame losses. The proposed technique is also able to conceal IDR frame losses. Moreover, to implement this algorithm only limited functionalities were added thus leading to low implementation overhead. The proposed technique has been adopted as a non-normative decoder option to the JM reference software at the JVT meeting in Poznan, Poland and has been incorporated into the SA4 video ad hoc group's toolkit at the 3GPP meeting at Paris in September 2005.

The rest of the paper is organized as follows: in Section 2 we give a brief overview of the error concealment schemes in H.264/AVC. The proposed algorithm is developed in Section 3. In Section 4 we present experimental results and conclusions are drawn in Section 5.

2. ERROR CONCEALMENT IN H.264/AVC

H.264/AVC is the latest video coding standard of ITU-T VCEG and ISO/IEC MPEG [9]. H.264/AVC is organized into two conceptual layers: the video coding layer (VCL) and the network abstraction layer (NAL). The VCL deals with the efficient encoding of the video data while the NAL defines the interface between the encoded video data and the transport media. The primary goals of H.264/AVC are improved coding efficiency and improved network adaptation. Error concealment is a non-normative feature in the H.264/AVC. It is an efficient way of providing error resiliency to the decoder without consuming any extra bitrate or incurring delay. An error resilient decoder should be capable of detecting and concealing the transmission errors.

Based on the information used, error concealment algorithms in H.264/AVC can be generally divided into two categories: spatial and temporal concealment. Temporal error concealment utilizes the temporal neighbors to conceal the errors in the current frame. H.264/AVC decoder implementation in JM 9.8 was unable to conceal completely lost frames. The spatial error concealment uses spatial redundancy to recover the damaged regions within the same frame.

*The work presented in this paper was carried out at Thomson Inc.

To increase the robustness of the bitstream to transmission errors at the cost of increased overhead, H.264/AVC may subdivide the picture into slices. The decoder is designed such that bit erroneous slices are discarded prior to passing the slices to the decoder. Again, the received data is buffered in such a way that the correct decoding order can be recovered. The “slice header” contains information about the picture it belongs to, the placement of the macroblocks encoded. An encoder can divide the picture into a number of slices. This increases the robustness of the bitstream to transmission errors. Every slice carries side information to allow its independent decoding. This requires replication of syntax elements like *frame_num* in the slice header. JM 9.8 was able to conceal slice losses. However, all the algorithms generally deal with situations in which only a portion of a received video frame is missing, e.g. some isolated or a set of consecutive MB’s.

An H.264/AVC decoder should also be able to conceal entire frame loss. The JM 9.8 was able to detect loss of reference frames by checking the coherence of the *frame_num* associated with each frame and carried in the slice header. The parameter *frame_num* was incremented by 1 for each reference frame sent by the encoder. Therefore, it is possible to define a procedure that checks this field for every slice and call a concealment routine when an unexpected gap is incurred. The implementation in JM 9.8 was not able to conceal any such loss and the decoder was unable to decode the bitstream. For non reference B and P pictures, the parameter *frame_num* is incremented at the most recent reference frame that preceded the disposable frame. Hence, JM 9.8 was unable to detect such losses. When transmitting low resolution sequences at low bit-rate viz. 3GPP networks it is often common that an entire picture fits the packet size. Hence, the decoder was unable to handle any loss pattern in such applications. This motivated us to propose the techniques for concealing reference and non-reference frame loss in H.264/AVC. The proposed techniques are implemented in versions later than JM 9.8. The techniques are elaborated in Section 3.

3. PROPOSED ALGORITHM

The classical concealment algorithm fails in case of entire frame loss due to unavailability of spatial neighbors. Video frames can be divided into two types in the H.264/AVC bitstreams, reference frames and non-reference frames. In the H.264/AVC decoder implementation of JM 9.8 software, the loss of non-reference frames cannot be detected. The decoder simply decodes the next available frame in the bitstream and skips the lost one. Hence, the output video sequence has fewer frames and this can cause display jitter and affects the viewing experience. In case of reference frames, the decoder stops without further decoding. In this section, we propose techniques for concealing such losses. Depending on whether a frame is a reference or a non-reference frame, different mechanisms are used to detect a frame loss. Upon the detection of a frame loss, a user-specified frame loss error concealment method is called to conceal the lost frame. The rest of the section is organized as follows: in Section 3.1 we present techniques for frame loss detection. The concealment algorithms are presented in Section 3.2.

3.1. Frame loss detection mechanism

1. Reference frame loss detection

The detection of a reference frame loss is based on the variable *frame_num*. Each encoded reference frame is assigned a value of *frame_num* and it is stored in its slice header. *frame_num* of an IDR frame is 0, and is incremented by 1 for each coded reference frame. When a reference slice is to be decoded, its *frame_num* is parsed from the slice header and compared with the one of the previous frame. If the current slice is not an IDR frame, and a gap more than one is found between the two *frame_num*, a reference frame loss is declared.

2. Detection of an IDR frame loss

Normally for frames within the same GOP, the *frame_num* of the frame currently being decoded is greater than the one of the previous frame. However, when an IDR frame is lost, the current *frame_num* is generally smaller than the previous one. So when a decrease in *frame_num* happens and the current *frame_num* does not equal to 0, an IDR frame loss is declared.

3. Non-reference frame loss detection mechanism

The loss of a non-reference frame does not affect the normal decoding process of the decoder. In order to detect a non-reference frame loss during the normal decoding process, the property of the decoder internal variable “Picture Order Count” (POC) can be utilized. The decoder assigns a POC value to each coded frame, starting from 0 for an IDR frame in a GOP. POC is assigned for the purpose of source decoding, such as weighted prediction for B slices, etc. Define a POC gap between a pair of POC values as

$$\text{POC Gap} = \text{POC}_{\text{Frame}_1} - \text{POC}_{\text{Frame}_2} \quad (1)$$

Typically, the POC gap for two temporally consecutive frames remains the same within a GOP. This property can be used to detect a non-reference frame loss. More specifically, with a prior knowledge of the POC gap value of the GOP being decoded, the decoder can check whether the POC gap between two temporally consecutive frames equals to the value. Because of the possible out-of-order coding for non-reference frames, such a check is conveniently performed whenever a frame is ready to be output. Since the possible lost reference frames have been detected and concealed based on *frame_num* prior to this stage, an unexpected POC gap thus indicates that a non-reference frame loss occurs.

3.2. Frame loss error concealment algorithm

1. “frame copy” algorithm

With the algorithm, each pixel value of the concealed frame is copied from the corresponding pixel of the previous decoded reference frame. While concealing a reference frame, the concealed frame is used for display, and is also placed into the reference picture buffer to be used for decoding subsequent pictures. In case of non reference frame concealment, the lost concealed frame is only used for display. An optional de-blocking filtering process can be applied.

2. “motion vector copy” algorithm

In the algorithm, the motion vectors and reference indices of the co-located blocks in the previously decoded reference frame are copied to the lost frame first. The motion vectors

are scaled based on the distance of the reference frame from the concealed frame. Then motion compensation is called to reconstruct the lost frame based on the copied motion information. In “motion vector copy”, the reference frame can be any frame available in the decoder buffer which carries motion information. So even if an IDR frame is lost, as long as it is not the first frame in the bitstream, it can still be concealed with “motion vector copy” algorithm by specifying the reference frame available in the decoded buffer, possibly from the previous GOP. An optional de-blocking filtering process can be applied.

4. EXPERIMENTAL RESULTS

In this section, a comparison is drawn between the “frame copy” algorithm and “motion vector copy” algorithm for concealing lost frames. The JVT common conditions video sequences (YUV 4:2:0) used in our experiments are shown in Table 1.

Sequence	Frame Rate	Number of Frames	Resolution
Foreman	10	300	QCIF
Container	10	300	QCIF
News	10	300	QCIF
Silent	15	300	QCIF
Mobile	30	300	CIF
Tempete	30	260	CIF

Table 1: The JVT common conditions used in Experiment 1.

The sequences shown in Table 1 are coded with Baseline profile, frame mode and IPPP... pattern at QP equal to 24, 28, 32, 36 and 40. 5 reference frames were used and Flexible Macroblock Ordering (FMO) and Arbitrary Slice Ordering (ASO) were disabled. Similar experimental results were obtained by encoding the IpP, IbP pattern of the sequences mentioned in Table 1 where p and b are non reference P and B pictures respectively. The four packet loss patterns with average packet loss rates of 3%, 5%, 10%, and 20% included in ITU-T VCEG Q15-I-16r1 were employed. The error patterns are generated from the experiments on the Internet backbone between one sender and three reflector sites. Details about the generation and file format are available in Q15-I-16r1. In our experimentation the packet loss is determined by the location of the character “0” in the error trace file. A “0” indicates a dropped NAL packet and a non zero character indicates successful packet transmission. The size of all packets is assumed constant and a packet loss is considered equivalent to a frame loss. Bitstreams with dropped pictures were created utilizing the packet loss patterns. The lossy bitstreams were decoded using both the “frame copy” and the “motion vector copy” methods. Table 2 shows the BD-PSNR differences between the two proposed error concealment algorithms for the sequences under different loss conditions. BD-bitrates are not given, because the un-errored bitstream bitrates were used in the computations.

Fig.1 shows the comparison of the two scheme and the no error case for the “foreman” sequence at 3% packet loss rates.

In Fig.2, a subjective comparison is made between the two proposed schemes. Since “motion vector copy” algorithm copies the motion information for each sub block, it is more appropriate than the “frame copy” algorithm for concealing video sequences with large amounts of motion.

Sequence	$\Delta PSNR$ for Packet Loss rate			
	3%	5%	10%	20%
Foreman	1.271	1.421	2.622	2.896
Container	0.037	-0.034	0.323	0.461
News	0.26	0.123	0.787	1.393
Silent	0.253	0.301	0.75	0.933
Mobile	3.054	3.817	4.941	6.418
Tempete	0.866	2.003	3.112	3.617
Average	0.957	1.271	2.089	2.619

Table 2: $\Delta BD - PSNR$ differences between the two proposed error concealment algorithms of the sequences for different packet loss rates.

The “foreman” and “mobile and calender” sequences in Figs. 2, 3 contain significant motion. Hence, the “motion vector copy” algorithm performs substantially better than the “frame copy” algorithm. However, “container” sequence, shown in Fig.4 has very little motion. Hence, a subjective comparison doesn’t reveal much difference.

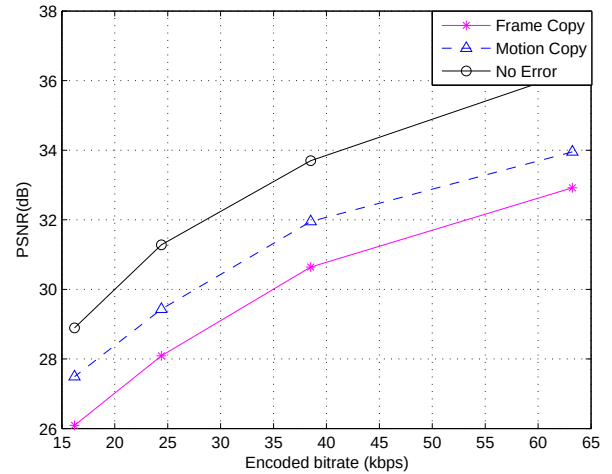


Fig. 1: Comparison of “frame copy”, “motion vector copy” and no error cases with “foreman” sequence for Average packet loss = 3%.

In Fig.5, a frame by frame comparison of the $PSNR$ for the luminance component ($PSNR_Y$) is drawn between the two proposed algorithms. The results are shown for $QP = 24$ of the “foreman” sequence at 20% packet loss rate. The comparison is shown for the first 30 frames in an IPP pattern. The errors occur in frames 2, 3, 6, 7, 15, 19, 20, 23, 28 and 30. The results show substantial improvement while using “motion vector copy” when compared to “frame copy” concealment. The insertion of IDR frames stops error propagation and causes improvement in video quality.

5. CONCLUSIONS

In this paper, an error concealment scheme is proposed for entire frame loss based on H.264/AVC. The proposed technique is capa-



Fig. 2: Frame 79 of the “foreman” sequence. QP=24, Average packet loss rate = 3%. Error Concealment using (a) “frame copy” algorithm, (b) “motion vector copy” algorithm.



Fig. 3: Frame 75 of the “mobile and calender” sequence. QP=24, Average packet loss rate = 20%. Error Concealment using (a) “frame copy” algorithm, (b) “motion vector copy” algorithm.

ble of concealing reference as well as non-reference frame losses. Upon the detection of a missing frame H.264/AVC conceals using either “frame copy” or “motion vector copy” algorithms. While concealment using a “frame copy” method is not strictly necessary for the JM software - the missing frame could have not been coded in the first place and the decoder would normally continue to display the previous picture - it is useful for experiments using fixed frame rate over lossy networks. With the proposed addition to the reference software, a frame-by-frame correspondence is achieved. The “frame copy” algorithm works well in temporal stationary areas but fails at moving areas. Experiments using the common conditions show that the “motion vector copy” algorithm outperforms the “frame copy” algorithm.

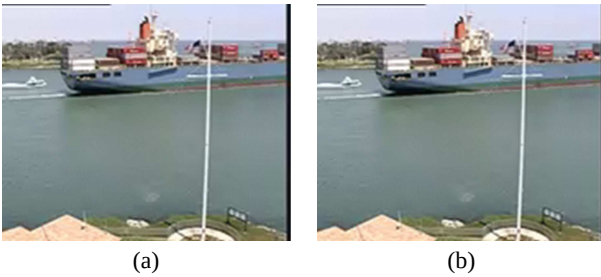


Fig. 4: Frame 50 of the “container” sequence. QP=28, Average packet loss rate = 10%. Error Concealment using (a) “frame copy” algorithm, (b) “motion vector copy” algorithm.

6. REFERENCES

- [1] S. K. Bandyopadhyay, Z. Wu, P. Pandit and J. M. Boyce, “Frame loss error concealment for H.264/AVC,” 73rd MPEG meeting and 16th JVT meeting, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-P072, July 2005.
- [2] Z. Wu, S. K. Bandyopadhyay and J. M. Boyce, “Motion copy decoder error concealment for H.264/AVC,” Tdoc S4-050483, 3GPP TSG-SA4 No.36 Meeting, September 2005.
- [3] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos, “Error resilient video coding techniques,” *IEEE Signal Processing Magazine*, pp. 61–82, July 2000.
- [4] J. Zheng and L. Chau, “A temporal error concealment algorithm for H.264 using lagrange interpolation,” *Proc. of IEEE Int. Symposium on Circuits and Systems*, vol. 2, pp. 133–136, May 2004.
- [5] Y. Xu and Y. Zhou, “H.264 video communication based refined error concealment schemes,” *IEEE Trans. Consumer Electronics*, vol. 50, pp. 1135–1141, November 2004.
- [6] Y. Chen, K. Yu, J. Li, and S. Li, “An error concealment scheme for entire frame loss in video transmission,” *Proc. of the Picture Coding Symposium*, December 2004.
- [7] P. Baccichet, D. Bagni, A. Chimienti, L. Pezzoni, and F. Rovati, “Frame concealment for H.264/AVC decoders,” *IEEE Trans. Consumer Electronics*, vol. 51, pp. 227–233, February 2005.
- [8] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, “Concealment of whole-frame losses for wireless low bit-rate video based on multiframe optical flow estimation,” *IEEE Trans. Multimedia*, vol. 7, pp. 316–329, April 2005.
- [9] T. Weigand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 13, pp. 560–576, July 2003.

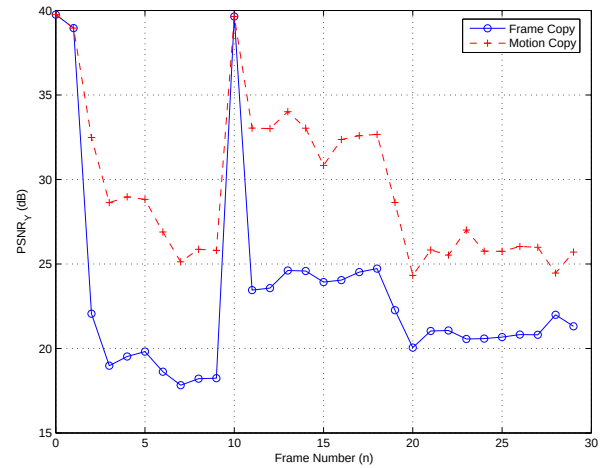


Fig. 5: Comparison of “frame copy”, “motion vector copy” methods to conceal the “foreman” sequence for average packet loss rate of 20%. The Errors occur in frames 2, 3, 6, 7, 15, 19, 20, 23, 28 and 30.