TRANSCODING BASED ROBUST STREAMING OF COMPRESSED VIDEO

Xiaopeng Fan, Oscar C. Au, Mengyao Ma, Ling Hou, Jiantao Zhou, Ngai Man Cheung

Hong Kong University of Science and Technology
Electronic and Computer Engineering
Clear Water Bay, Hong Kong

ABSTRACT

A variety of techniques have been proposed to enhance the error robustness of the video streaming system. However, most of them improves the error resilience during compression rather than after compression. In this paper, we propose a novel transcoding based scheme called lossless inter frame transcoding (LIFT) scheme to improve the error resilience of existing compressed video stream. In the LIFT scheme, inter coded blocks are selectively transcoded into new kind of blocks called 'L-block'. At the decoder, the L-block can be transcoded back to the original P-block when the prediction is available and can also be robustly decoded as I-block when the prediction is unavailable. By offline transcoding and online adjusting the ratio of P-blocks and L-blocks, the proposed streaming server achieves error robustness scalability. Experimental results demonstrate the correctness and effectiveness of the proposed method.

Index Terms— video streaming, error resilience, intra refresh, L-block, adaptive error robustness

1. INTRODUCTION

Due to the rapid growth of the Internet and the increasing demand for video contents, video streaming services over communication network has received tremendous attention from academia and industry. In a typical streaming system, the compressed video streams are transmitted by the server to the client device(s) for decoding and display in real time. To against the packets loss or delay during the transmission, a variety of error resilience techniques have been proposed[1][2][3].

In video streaming system, the server may transmit online compressed video or offline compressed video (e.g. in VOD system). However, most of the previous error resilience techniques mainly focus on improving the error robustness during online compression. Recently, the problem of transmitting offline compressed video in error prone channel is first addressed in [4], in which an alternate macroblock coding (AMC) method is proposed under the assumption of a low delay feedback.

In this paper, we consider the error resilience transmission of offline compressed video, when the server know the packet loss rate of the channel but can not get the feedback acknowledgment as assumed in [4]. Observing that most macroblocks should be coded in inter mode rather than intra mode in offline compressed inter frames, we propose a lossless inter frame transcoding (LIFT) method which can convert an inter coded P-block to a new type of block called L-block to achieve intra-like error robustness. The L-block can be losslessly trancoded back to the exact original P-block when there is no transmission error, and can also be decoded robustly like an I-block when the predicted block is not available.

Further more, we propose a novel offline transcoding and online composing scheme based on the proposed LIFT method. For time-varying channel, this scheme can achieve same adaptive error robustness as online transcoding when the server can not afford online transcoding (e.g. when there are multiple clients of different channel status).

The rest of this paper is organized as follows: Section 2 present the proposed lossless transcoding method and the proposed error robust streaming server. Section 3 gives the experimental results and Section 4 concludes the paper.

2. PROPOSED SCHEME

2.1. Lossless inter frame trancoding (LIFT)

Given a compressed video stream, the proposed LIFT method is to losslessly transcode the P-blocks to achieve intra-like error robustness.

Let $\hat{x}$ denote the reconstructed pixel value in the (to be transcoded) P-block, y denote the inter frame prediction and $\hat{e}$ denote the reconstructed residue respectively. We use $\hat{X}$, Y and $\hat{E}$ to indicate the corresponding signals in transform domain, and use T(.), Q(.) and E(.) to indicate the process of transform, quantization and entropy coding respectively. The proposed LIFT transcoding scheme is to re-encode the compressed video such that it can be recovered losslessly when there is no transmission error, and can also be decoded robustly when the prediction y is not available. As shown in Fig.1, the input of the LIFT transcoder is the quantized residue $Q(E)$ and the prediction y, obtained by decoding the

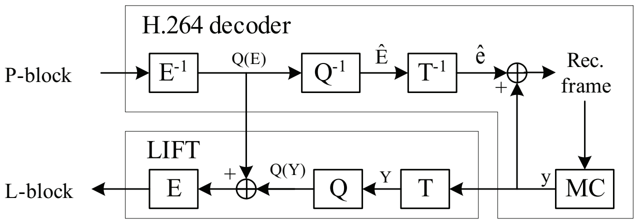

Fig. 1. LIFT transcoder

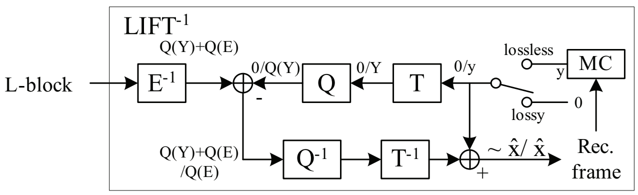

Fig. 2. LIFT decoder

original compressed video stream. The prediction y is transformed by the 4×4 integer transform and then quantized to get quantized prediction $Q(Y)$. Then the sum $Q(E) + Q(Y)$ are entropy coded to replace the original entropy coded residue $Q(E)$. To save bitrate, the DC components of $Q(E)+Q(Y)$ are predicted by neighbor blocks. We keep the motion vectors in the bitstream and compress a binary map to indicate P-block and L-block.

2.2. LIFT decoder

At the LIFT decoder, firstly y and $Q(Y)$ are obtained by motion compensation, transform and quantization. Secondly, $Q(Y) + Q(E)$ are entropy decoded and are used to subtract the $Q(Y)$ to get $Q(E)$. Finally, with the quantized residue $Q(E)$ and the prediction y, the reconstructed signal $\hat{x}$ is recovered losslessly. This decoding process can be expressed as follows:

$$\begin{aligned} & y + T^{-1}(Q^{-1}(Q(Y) + Q(E) - Q(Y))) \\ = \; & y + T^{-1}(Q^{-1}(Q(E))) \\ = \; & y + T^{-1}(\hat{E}) \\ = \; & y + \hat{e} \\ = \; & \hat{x} \end{aligned} \tag{1}$$

In case that the prediction y is not available due to packet loss, we can still decode robustly since $Q(Y) + Q(E)$ can be considered as an approximately scaled version of $\hat{X}$. In this case we directly dequantize $Q(Y) + Q(E)$ and get an approximate reconstruction by inverse transform. In this case, the reconstruction is:

$$\begin{aligned} & T^{-1}(Q^{-1}(Q(Y) + Q(E))) \\ \approx \; & T^{-1}(Y + \hat{E}) \\ = \; & y + \hat{e} \\ = \; & \hat{x} \end{aligned} \tag{2}$$

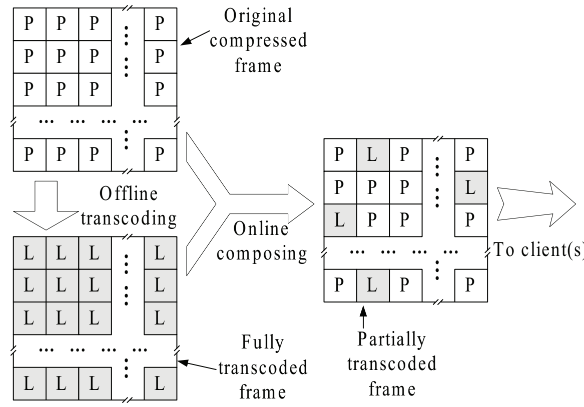

Fig. 3. The operations in the proposed video server

Therefore, the proposed method can recover the original reconstruction in lossless case, and can also restore it approximately in lossy case, i.e. the proposed L-blocks have inter-like reconstruction in lossless case and achieves intra-like robustness in lossy case. Since the magnitude of $Q(Y)+Q(E)$ is larger than $Q(E)$ and is similar to $Q(X)$, the proposed L-blocks have intra-like bitrate.

2.3. Video server with adaptive error robustness

For time-varying channel, offline transcoding can hardly achieves optimal error resilience performance due to the absence of the channel status information, while online transcoding is too complicated especially when there are multiple clients with different channel status. Therefore, we proposed a novel scheme performing offline transcoding and online composing to achieve adaptive error robustness for time-varying channel.

Given a compressed video, the server first offline transcodes all the P-blocks and stores both the P-blocks and the L-blocks in the storage space. When streaming the video, instead of sending all the P-blocks, the server online composes the frames by P-blocks and L-blocks. The L-blocks and the P-blocks can arbitrarily replace each other, since the proposed LIFT is lossless. The server adjusts the ratio of P-blocks and L-blocks according to packet loss rate p. Typically when the pack loss rate increases, more L-blocks and less P-blocks are used to make the video stream more robust.

The problem to optimize the ratio and the locations of L-blocks is quite similar to the problem of intra refresh, since our L-blocks have intra-like bitrate and intra-like robustness

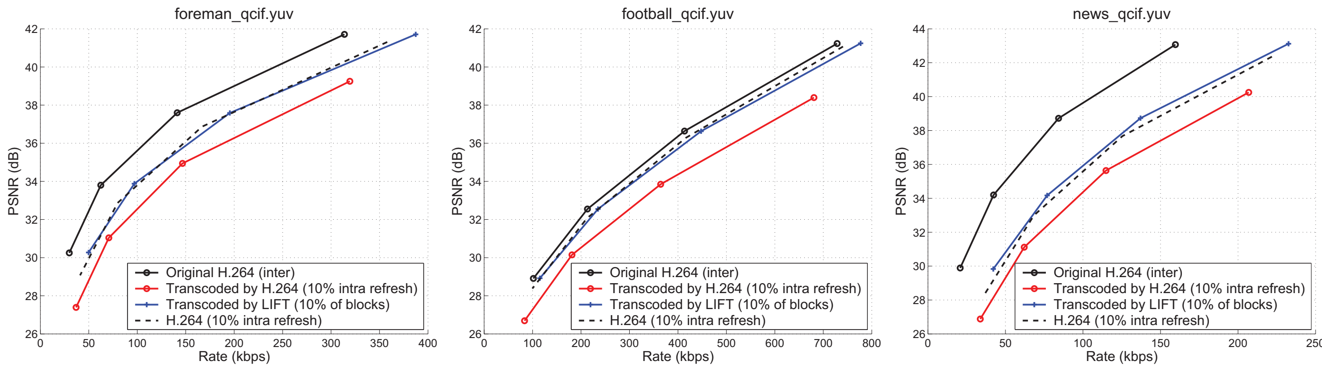

Fig. 4. Rate-Distortion performance

in a lossy situation. According to [5], the optimal intra refresh rate is approximately equal to the packet loss rate p. Thus in our scheme the ratio of L-blocks is p in the composed frame. The position of L-blocks is selected randomly rather than by RDO as in [3], due to the absence of the original video as reference for distortion calculation.

Notice that, to improve the error robustness of a given compressed video, one can also re-encode it to add more I-blocks. However, this recompression introduces transcoding distortion while the proposed LIFT method is lossless. More importantly, the proposed scheme can offline perform transcoding and online adjust the number of L-blocks and P-blocks according to the packet loss rate. However, the recompression method can not online adjust the number of I-blocks if performed offline, and is too complicated if performed online especially when there are multiple clients with different packet loss rate.

3. EXPERIMENTS

In the experiments, we assume all the videos have been encoded by H.264 (JM 13.2, baseline profile) and will be transmitted over a packet erasure channel. The GOP structure is 'IPPP...'. We mainly compare three different methods: the first is to send the video stream directly, the second is to re-encode the video by H.264 with intra refresh (i.e. forcing some blocks to be I-blocks), and the third method is to transcode the video by the proposed LIFT transcoder.

3.1. RD performance

In RD performance test, we encode each video at 4 different bitrate and then transcode the 4 compressed video streams. We transcode 10% of P-blocks into L-blocks by the proposed LIFT transcoder. As benchmark we also re-encode the video by H.264 with 10% intra refresh. To be fair, the location of the L-blocks in our scheme are exactly the same as the I-blocks newly emerged in the recompression scheme.

As shown in Fig.4, the proposed LIFT transcoder has higher RD performance than the H.264 based transcoder, since the proposed LIFT transcoder is a lossless transcoder. Compared with the original H.264 video stream, the proposed transcoded stream has the same PSNR but higher bitrate which is the cost to achieve error robustness.

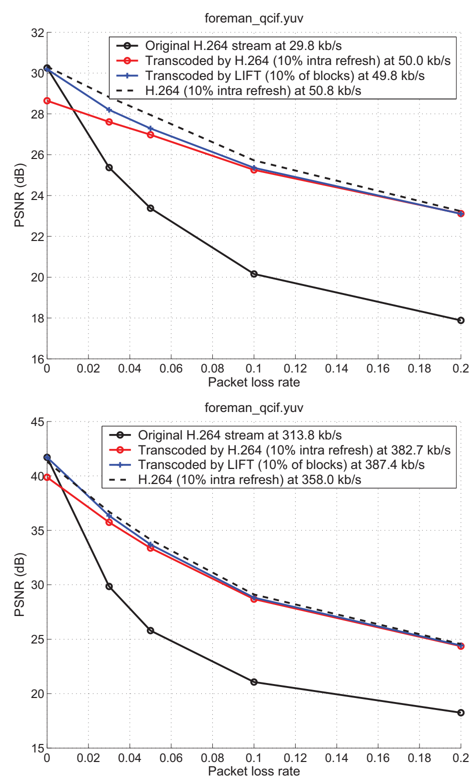

Fig. 5. Error resilience performance (fixed refresh rate)

In this figure we also give another RD curve (the dash line) for reference, which is compressing the original (uncompressed) video with 10% intra refresh. We can see that our method can achieve similar RD performance with it although

Table 1. Adaptive error resilience test (foreman qcif at 15Hz)

Test 1	T1: ($p = 0\%$)		T2: ($p = 10\%$)		T3: ($p = 20\%$)		overall ($\bar{p} = 10\%$)	
	bitrate	PSNR	bitrate	PSNR	bitrate	PSNR	bitrate (kb/s)	PSNR (dB)
original H.264 stream (to be transmitted)	29.84	30.25	29.84	20.16	29.84	17.88	**29.84**	**22.76**
offline recompression (add 10% intra)	50.02	28.64	50.02	25.26	50.02	23.11	**50.02**	**25.67**
proposed adaptive scheme	29.84	30.25	49.82	25.36	69.41	24.77	**49.69**	**26.79**
H.264 stream at 50kb/s (for comparison only)	50.43	32.93	50.43	20.60	50.43	18.07	**50.43**	**23.87**
Test 2	T1: ($p = 0\%$)		T2: ($p = 5\%$)		T3: ($p = 10\%$)		overall ($\bar{p} = 5\%$)	
	bitrate	PSNR	bitrate	PSNR	bitrate	PSNR	bitrate (kb/s)	PSNR (dB)
original H.264 stream (to be transmitted)	29.84	30.25	29.84	23.38	29.84	20.16	**29.84**	**24.60**
offline recompression (add 5% intra)	40.04	28.74	40.04	26.02	40.04	23.70	**40.04**	**26.15**
proposed adaptive scheme	29.84	30.25	40.11	26.36	49.82	25.36	**39.92**	**27.32**
H.264 stream at 40kb/s (for comparison only)	40.90	31.94	40.90	24.13	40.90	20.50	**40.90**	**25.52**

we do not have access to the original video.

3.2. Error resilience performance

In Fig.5, we compare the error resilience performance of the proposed LIFT transcoder and the H.264 transcoder at different packet loss rate 0%, 3%, 5%, 10%, 20%. For each loss rate, we simulate 100 different random loss patterns. The proposed transcoder performs similar to the H.264 transcoder at high loss rate, but can gain 0.6dB at loss rate 3% and 1.8dB in error free case. Again our curve can approaches the dash line (H.264 with 10% intra refresh) which assumes the original uncompressed video is available.

3.3. Adaptive error resilience

We conduct two tests on transmitting compressed video over time-varying channel. In the first test, the average packet loss rate $\bar{p}$ is 10% but p changes during the transmission. The loss rate p is 0% at the beginning, but changes to 10% after 1/3 transmission time and becomes 20% at the last 1/3 transmission time. In the second test, the loss rates are 0%, 5% and 10% respectively in the three time slots and the average loss rate $\bar{p}$ is 5%. In each time slot, the video sequence is transmitted 100 times to get average performance.

We assume the server is not able to afford online transcoding (e.g. due to the huge amount of clients), and thus all the transcoding (if required) need to be performed offline.

The simulation results are given in Table 1. The proposed streaming server performs offline transcoding and online composing as explained in Section 2.3. Since the ratio of L-blocks is equal to the packet loss rate p at each time slot, there are no L-blocks in the first time slot and more L-blocks in the third time slot. However, the H.264 recompression scheme can only transcode the video offline with fixed intra refresh rate (10% in the first test and 5% in the second test), due to the restriction of complexity. Therefore, the H.264 recompression scheme has fixed bitrate during the three time slots. By comparing the results in Table 1 we can see that our scheme saves bitrate at the first time slot (corresponding to low packet loss rate) without increasing the distortion, and puts more bits into the third time slot (corresponding to high packet loss rate) and reduces the distortion significantly. For the whole transmission, the proposed scheme gains more than 1.1dB in average than H.264 recompression at same bitrate.

4. CONCLUSION

In this paper, we propose a lossless inter frame transcoding (LIFT) method to improve error resilience capability of offline compressed video. Compared with H.264 based recompression, we achieve better RD performance and error resilience performance. The proposed error robust server can adapt to the channel condition and can gain more than 1.1dB over H.264 based recompression scheme when the video is transmitted over time-varying channel.

5. REFERENCES

[1] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, 1999.

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

[3] R. Zhang, SL Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packetloss resilience," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 6, pp. 966–976, 2000.

[4] Xiaosong Zhou and C.-C. Jay Kuo, "Robust Streaming of Offline Coded H.264/AVC Video Via Alternative Macroblock Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 4, pp. 425–438, 2008.

[5] G. Cote and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the Internet," *Signal Processing: Image Communication*, vol. 15, no. 1, pp. 25–34, 1999.