Error Resilience in Network-Driven Wyner-Ziv Video Coding

Limin Liu[a], Pau Sabrià[b], Luís Torres[b], and Edward J. Delp[a]

[a] Video and Image Processing Laboratories (VIPER),
School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana
[b] Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona,
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

ABSTRACT

Previously we presented a network-driven Wyner-Ziv video coding method, in which the motion vectors are derived at the decoder and sent back to the encoder through a reliable backward channel. In this paper, we consider the scenario when the backward channel is error resilient. We study the performance of error resilient methods for our codec. A symmetrical Reversible Variable Length Code (RVLC) is used to reduce the bandwidth requirement of the backward channel. A hybrid scheme with selective coding is proposed to improve the coding efficiency when transmission delay occurs. The experimental results show that these error resilient methods can consistently improve the video quality at the decoder.

Keywords: Wyner-Ziv video coding, network-driven motion estimation, error resilience, reversible variable length code (RVLC)

1. INTRODUCTION

The latest video coding standard H.264/AVC [1] adopted many new coding tools to improve video compression performance. This leads to significant complexity increases at both the encoder and the decoder .[2] This poses a challenge for applications such as wireless video surveillance systems, video conferencing over wireless networks and multimedia messaging services on mobile telephone. In these applications, a low complexity encoder is preferred to achieve real time encoding.

Wyner-Ziv video coding, also known as distributed video coding (DVC), was developed based on theory presented in the 1970s. Consider two correlated information sources X and Y, as shown in Figure 1, encoded by two separate encoders A and B, neither has the access to the other source. A rate R is *achievable* if for any $\varepsilon > 0$, there exists an encoder and decoder such that $Pr(\hat{X} \neq X) < \varepsilon$. If joint decoding is allowed, the Slepian-Wolf Theorem[3] proves that the achievable rate region for the system in Figure 1 is $R_X \geq H(X|Y)$, $R_Y \geq H(Y)$, $R_X + R_Y \geq H(X,Y)$. Hence, regardless of its access to side information Y, encoder A can encode X with arbitrarily high fidelity as long as the decoder A has access to Y and the rate is equal to or larger than $H(X|Y)$. Wyner and Ziv extended the result to the lossy compression [4] and proved that the side information at the encoder is also not necessary to achieve rate distortion performance for lossy compression.

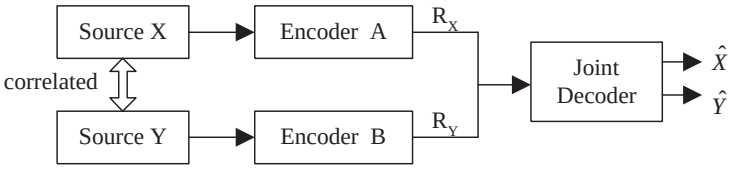


Figure 1. Slepian-Wolf coding

Many current Wyner-Ziv video coding schemes are based on the channel coding techniques.[5–13] In these implementations, INTRA-coded frames are frequently used to ensure low complexity at the encoder and side

This work was supported by the Indiana 21^{st} Century Research and Technology Fund. Address all correspondence to E. J. Delp, E-mail: ace@ecn.purdue.edu)

Visual Communications and Image Processing 2006, edited by John G. Apostolopoulos, Amir Said,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6077, 607717, © 2005 SPIE-IS&T · 0277-786X/05/$15

estimation quality at the decoder. This can lead to a concern with coding performance. Previously we proposed a Wyner-Ziv video coding method[14] based on network-driven motion estimation (NDME)[15] to reduce the dependency of the INTRA modes. Experimental results showed that our network-driven Wyner-Ziv video coding can significantly improve the coding efficiency. In NDME the coding complexity is shifted to the decoder since the motion estimation is performed at the decoder. The motion vectors are computed at the decoder and sent back to the encoder, which requires the existence of a backward channel.

In [14] we assumed that the backward channel is reliable and the motion vectors can be sent back to the encoder without error and delay. This is not always the case in practical channels, especially for the applications involving wireless channels where the channel is generally less reliable. Since the motion vectors sent back to the encoder play a crucial role in network-driven Wyner-Ziv video coding, it is important to make sure the motion vectors are resilient to transmission errors and delays.

Error resilient video coding techniques [16] have been developed to mitigate transmission errors. Conventional motion-compensated prediction (MCP) based video coders, such as H.264/AVC [1] and MPEG-4,[17] include several error resilience methods .[18–20] Since the structure of Wyner-Ziv video coding is different from MCP-based video coding, we need to take it into consideration when we design an error resilience method for Wyner-Ziv video coding. In ,[11] an error resilient method is proposed for Wyner-Ziv video coding by periodically transmitting a small amount of additional information to prevent the indefinite propagation of errors.

In this paper, we propose to use a symmetrical Reversible Variable Length Code (RVLC) to reduce the bandwidth requirement of the backward channel. A hybrid scheme with selective coding is proposed to improve the coding efficiency when transmission errors and delay occur. The rest of this paper is organized as follows. In Section 2 we describe our network-driven Wyner-Ziv video coder. Section 3 we present our error resilient methods for network-driven Wyner-Ziv video coding. Simulation and experimental results are presented in Section 4. Section 5 concludes the paper with remarks.

2. NETWORK-DRIVEN WYNER-ZIV VIDEO CODING

Network-Driven Wyner-Ziv video coding[14] incorporates a channel-coding based Wyner-Ziv scheme with NDME. Basically there are two types of frames: Wyner-Ziv frames (WZ frames) and network-driven frames (N frames) as shown in Figure 2. The Wyner-Ziv encoder includes a uniform quantizer and a turbo encoder .[21] A rate compatible punctured turbo (RCPT) code is used and only a portion of the parity bits are sent to the turbo decoder. At the decoder, the turbo decoder reconstructed the frame with the side information derived from the N frames. If the reconstructed frame does not satisfy the requirement of the quality, the decoder requests more parity bits from the encoder.

A conventional Wyner-Ziv video encoder encodes many frames as INTRA frames to guarantee low complexity encoding and accurate side information at the decoder. It does not make the best use of temporal correlation across the frames, which is a major factor in conventional MCP-based video coding. We replace the I frames with N frames which can be regarded as pseudo P frames. Motion estimation is performed on the previous reconstructed frames at the decoder. The motion vectors for the current frame are predicted from the motion vectors of the previous frames and sent back to the encoder as shown in Figure 2. The high-complexity motion search is done at the decoder. The encoder does the motion compensation and the residual coefficients are transform coded. This video coding method shows a gain in the coding efficiency over conventional Wyner-Ziv video coding. One major contribution of network-driven Wyner-Ziv video coding is to reduce the dependence on the intra-coded frames. By transmitting motion information from the decoder back to the encoder, we are able to increase the use of predictive coded frames without increasing the encoder complexity.

In practice bandlimited channels generally suffer from various types of degradations such as bit/erasure errors and synchronization problems. The compressed video streams are vulnerable to the errors due to the use of predictive coding and entropy coding. In ,[14] the forward channel transmit the parity bits of the WZ frames and the residual coefficients of the N frames. The parity bits generated by the turbo encoder are inherently error resilient to transmission errors. For the residual coefficients, conventional error resilient tools can be used to deal with unreliable networks since the transform coefficients are coded in the same way as in MCP-based video coding. In this paper we address the error resilient problem in the backward channel for the transmission of motion vectors.

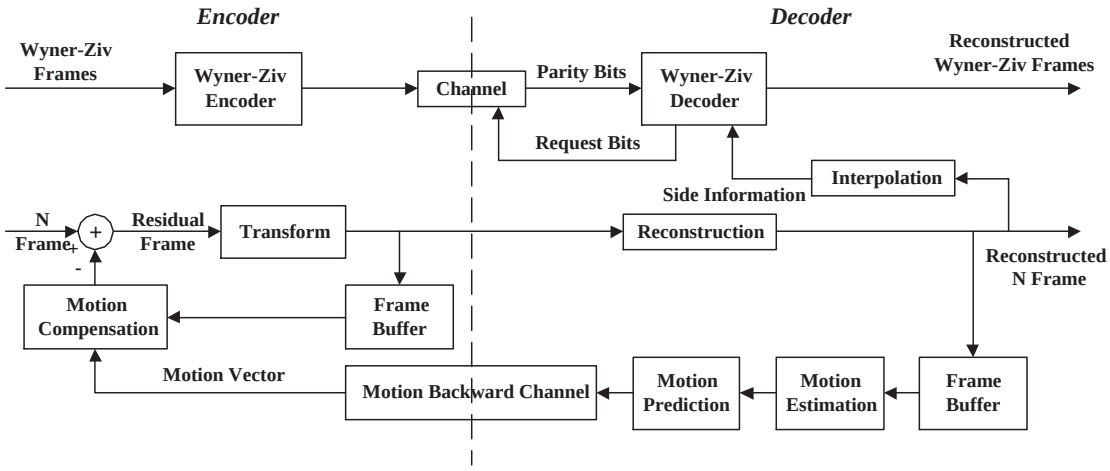


Figure 2. Simplified diagram of the Network-Driven Wyner-Ziv Video Codec

3. ERROR RESILIENCE IN THE BACKWARD CHANNEL

When video data is transmitted over the network, error-free delivery of the data packets are typically unrealistic due to either traffic congestion or impairments of the physical channel. Data recovery tools can be used to recover the lost or noisy data. The errors can also lead to de-synchronization of the encoder and the decoder. In this section we address error control and synchronization on both sides of the channel. Moreover, we also aim to limit the bandwidth of the backward channel used in network-driven Wyner-Ziv video coding. We describe the tools we propose to solve these problems below.

3.1. Bandwidth Limitation

The backward channel in the original Wyner-Ziv scheme is utilized as a communication channel to make requests for more parity bits from the encoder. In the new scheme, the backward channel transmits the motion vectors as well. H.264/AVC standard [22] supports seven block shapes, i.e., 16×16, 16×8, 8×16 and 8×8 in the syntax. If the 8×8 partition is chosen, it can be further divided into 8×4, 4×8 or 4×4 blocks. Although the encoder can obtain more accurate motion vectors with more block shapes, it requires additional bits to transmit the block shape information and the corresponding motion vectors. Furthermore, since the motion vectors predicted at the decoder are not as accurate as the motion vectors derived directly at the encoder, the use of more block shapes does not necessarily improve the coding efficiency as much as it does in MCP-based encoder. Hence here we only use fixed block size.

The data rate of the backward channel is given by:

$$R_{\text{motion}} = 2 \cdot N_{\text{block}} \cdot L_{\text{MV}} \cdot f_{\text{frame}} \tag{1}$$

where R_{motion} is the data rate of the motion vectors sent through the backward channel, N_{block} is the number of the blocks in a frame, L_{MV} is the average length of the codewords representing the motion vectors, and f_{frame} is the frame rate measured in frames/s. The constant 2 exists in (1) due to the motion vectors of both vertical and horizontal directions. The data rate of the backward channel is larger than 100 kbits/s if a QCIF sequence is coded with a block size of 8×8 pixels, half pixel resolution motion vectors, max search range of 16 pixels and a frame rate of 30 frames/s. It accounts for approximately $10\% - 20\%$ of the bandwidth of the forward channel.

One way to reduce R_{motion} is to reduce the number of blocks in a frame N_{block}, which can be achieved by increasing the size of the blocks used in motion estimation and motion compensation. If the size of the blocks are increased from 8×8 to 16×16, the data rate of the motion vectors could be reduced by 75%.

We further lower the average length of the codewords representing the motion vectors L_{MV} through entropy coding by Variable Length Coding (VLC). The basic procedure for the design of the code is described in the following. First, the decoder predicts the motion data by selecting the optimal mode proposed in .[14] Second, the cost of every possible motion vector is estimated for the entire frame, where the cost is derived by computing the frequency of the occurrence. The total number of the possible motion vector symbols is 64 since half pixel resolution search is performed and the search range is 16 pixels. Then the codeword table is set up according to the costs by creating a binary tree of all motion vector symbols. Once the codeword table is built, the motion vector of every block is coded accordingly.

In order to decode the motion data, the encoder needs to have the codeword table. An entire sequence can use a static codeword table. However, experimental results show that updating the codeword table every frame improves the adaptivity and the coding efficiency. To reduce the data rate spent on the codeword table, we exploit an alternative approach by sending the cost table. The approach is based on the assumption that both the encoder and the decoder are capable of building identical binary trees for the codeword tables. To reduce the data rate further, we non-uniformly quantize the costs of the motion vectors to the power of 2 and send the logarithm of the costs. Table 1 describes the design of the cost table. For QCIF sequences, the number of the blocks per frame is 99, which is the maximum original cost. The quantized cost can only reach the maximal number of 64 and thus the logarithm of the quantized cost is 6. To make the process of extracting the cost table simple, we use a fixed-length code to represent the logarithms of the quantized costs. Eight symbols ($0-6$ and null) use 3 bits to code. Through the design of the cost table, we can reduce the data rate for the additional decoding information of the Huffman code. Figure 3 shows the bitstream components of the motion back channel.

Original Cost	Quantized Cost	Logarithm	Code
0	0	null	000
1	1	0	001
2-3	2	1	010
4-7	4	2	011
8-15	8	3	100
16-31	16	4	101
32-63	32	5	110
64-127	64	6	111

Table 1. Design of the cost table

Coded Cost Table	Huffman Coded Motion Vectors

Figure 3. Data stream structure for the backward channel

3.2. Data Recovery by Symmetrical RVLC

Traditional VLC schemes such as the Huffman code, as we described in Section 3.1, have proved vulnerable to channel errors. A single error can cause a the propagation of the error across the entire bit stream. When an error occurs, the Huffman decoder has to discard all the bits until synchronization is re-established. The Reversible Variable Length Code (RVLC) is utilized in the MPEG-4 error resilient tools for the recovery of the residual coefficients.[18] It enables both forward and backward decoding thus it prevents the error propagation. Figure 4 shows the decoding process for the RVLC. The black box shows the data that will be discarded which is much less than the corrupted data between the two resynchronization markers in traditional VLC schemes.

There are two types of RVLC codes, symmetrical RVLC and asymmetrical RVLC .[23] Generally the asymmetrical RVLC is more flexible in codeword assignment and is thus more efficient than the symmetrical RVLC. However, it suffers from increasing the complexity of decoding and error detection. To detect the position of errors, the RVLC decoder needs to compare the output of the forward and backward decoding, which adds extra

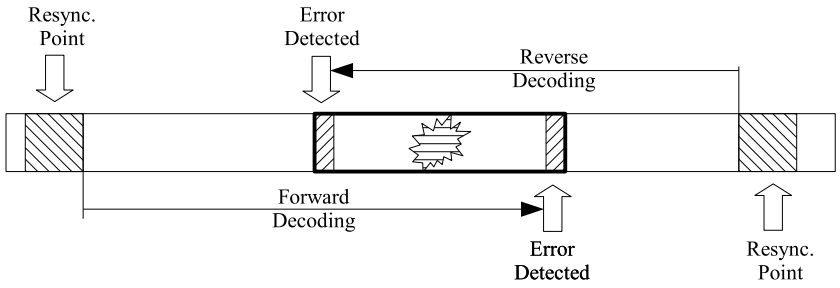


Figure 4. RVLC decoding process

complexity to the encoder. Therefore, we implemented a type of symmetrical RVLC. It provides the capability of straightforward error detection by observing the symmetry of the bitstream. Furthermore it does not need to send two codeword tables as the asymmetrical RVLC requires. We first design a non-reversible Huffman code (Section 3.1). Then every codeword is assigned to a symmetrical counterpart by a top-down scheme.[24] The details of the construction of an efficient symmetrical RVLC are presented in.[23, 24]

3.3. Resynchronization

In standard video compression, the decoder will lose synchronization when burst errors occur. The quality of the reconstructed video will degrade quickly until some remedial actions are taken. The same problem occurs in the backward channel of network-driven Wyner-Ziv video coding. Since the motion information is essential to the encoding of the bitstream, the problem becomes even more severe with transmission delay. If the motion vectors are decoded erroneously or the motion vectors lose synchronization with the corresponding blocks, the motion compensation at the encoder cannot continue.

As we described in Section 3.2, the symmetrical Reversible Variable Length Code (RVLC) is suitable for error detection and resynchronization. However, when a transmission delay occurs, there may be delays across the frames. In a normal coding scenario, the decoder sends the motion vectors of the i-th frame denoted as $\mathrm{MV_i}$. The encoder receives $\mathrm{MV_i}$ and generates the residual frame denoted as $\mathrm{RF_i}$ and sends the bitstream through the forward channel. At the decoder, the decoder reconstructs the frame by the received $\mathrm{RF_i}$ and the stored motion vector $\mathrm{MV_i}$. The situation changes when there is transmission delay at the channel. Assume the encoder receives the motion vectors of the $\mathrm{i}-1$ frame. The encoder generates the residual frame denoted as $\mathrm{RF_i'}$ with the motion vector $\mathrm{MV_{i-1}}$. The decoder reconstructs the frame with the residual data $\mathrm{RF_i'}$ and the motion vector $\mathrm{MV_i}$. Therefore the reconstructed frames at the encoder and the decoder lose identity which propagates to the entire sequences. Figure 5 shows an example of transmission delay. Suppose the reference frames at the encoder and the decoder are the same, due to the difference of the motion data, the motion compensated frames at the encoder and decoder are different. This leads to drift between the reconstructed frames at the encoder and the decoder. This drift propagates and accumulates, and leads to continuous quality degradation.

To show an example of the transmission delay, we tested the *foreman* QCIF sequence. As shown in Figure 6 (a), transmission delay occurs through the sequence. Assume from 10-th frame to 50-th frame, there is one-frame delay in the transmission of the motion vectors. In addition, there is another one- frame delay from the 100-th frame to the 150-th frame. There is a two-frame delay from the 200-th frame to the 225-th frame. We observe a loss of 10 dB for the one-frame delay in Figure 6 (b). When the number of delayed frames increases to two, more loss is observed. Although N frames only account for half of the sequence, the loss persists throughout the sequence since the initial estimate of the WZ frames are worse.

To address the delay problem, a two-stage procedure is proposed. First we detect the delay and pass the location of the delay to the encoder. Second the encoder turns to the error-handling state with delay signaling.

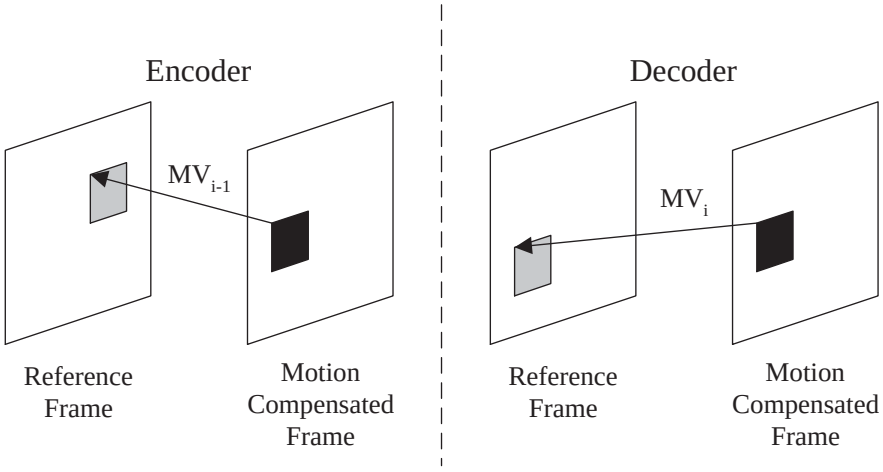
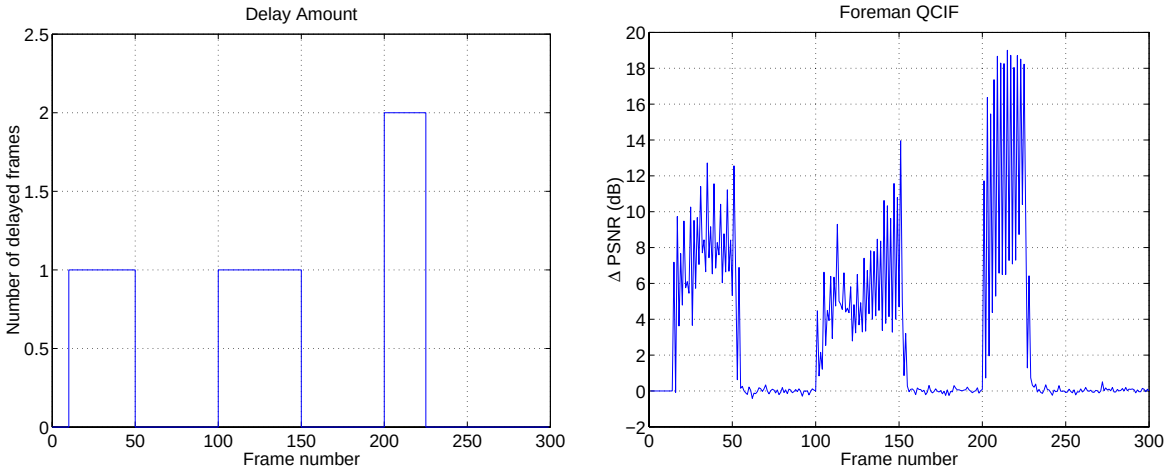


Figure 5. Effect of transmission delay



(a) (b)

Figure 6. (a) Delay pattern over the sequence (b) PSNR loss in performance with transmission delay

3.3.1. Delay Detection

We see sharp drop in PSNR with the occurrence of delay in Figure 6. One simple delay detection method is to compute the PSNR at the encoder, which brings additional computational burden to the encoder. Another approach is to insert resynchronization markers which provides periodic synchronization check.

Similarly, we create a data stream structure where some information is provided for delay detection. As shown in Figure 7 an entry denoting the index of the frame is inserted before sending the motion data. The bandwidth needed to send this extra field is marginal. Thus the field of the frame number functions as a resynchronization marker. The amount of delay can also be extracted from the delay detection.

Frame Number	Coded Cost Table	Huffman Coded Motion Vectors

Figure 7. Data stream structure for the backward channel for delay detection

3.3.2. Delay Management

One way to estimate the motion vectors is to make use of the information from the spatially or temporally neighboring motion data. In the delay situation, the motion data of the spatially neighboring is lost simultaneously. Employing the temporally neighboring motion vectors has demonstrated quality degradation. Since we can detect the delay as described in Section 3.3.1, we can treat the frames with delay in a different way. We denote this form of hybrid network-driven Wyner-Ziv video coder as the selective encoding scheme.

In the hybrid scheme, the delay detector controls the switch between the I frames and N frames. When a frame delay is detected, the encoder ignores the motion data and codes the current frame as an intra frame. Thus error propagation to the following frames is prevented with higher forward data rates. The diagram of the hybrid scheme is shown in Figure 8 where a switch decides the coding methods. The change of the coding scheme is presented in the header of the residual data and the decoder makes the corresponding adjustment.

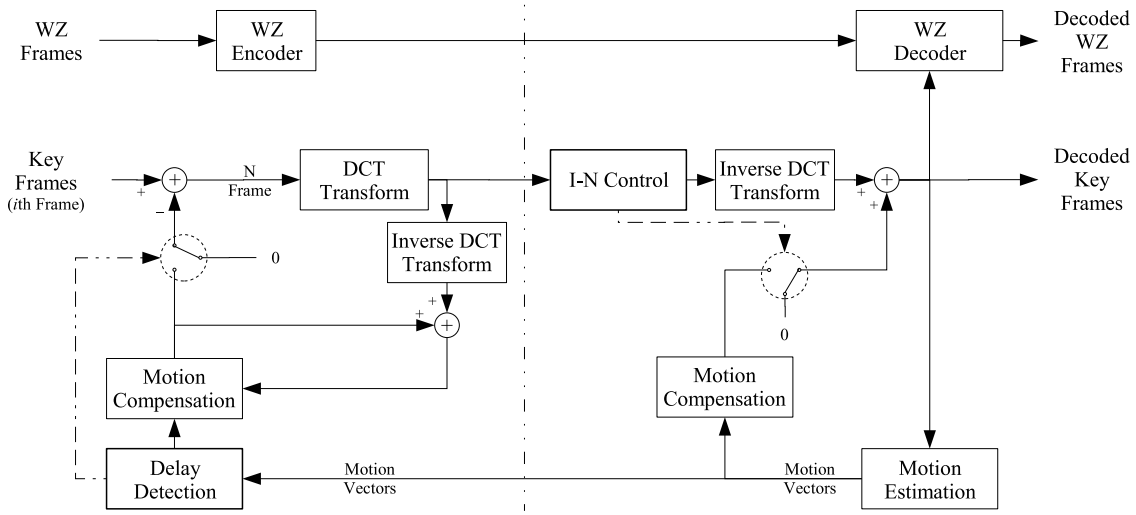


Figure 8. I-N hybrid scheme for the delay management.

4. EXPERIMENTAL RESULTS

The implementation of network-driven Wyner-Ziv video coding was made by modifying the source code of the H.264 reference software JM9.6 .[25] Four standard QCIF video sequences, *foreman*, *coastguard*, *stefan* and *silent*, were used in the experiments. The first and the third frames are intra-coded. The remaining half of the frames

are coded as N frames when there is no transmission delay and the other half frames are coded as WZ frames. Three bit-planes of the pixel values are sent to the turbo encoder, i.e., every pixel $(0-255)$ is represented by its three most significant numbers.

4.1. Symmetrical RVLC

We implemented the Huffman code and the Huffman-based symmetrical RVLC to reduce the bandwidth of the backward channel. In Table 2 we show the average percentages of the coded motion data compared with the data rate using both VLC and RVLC. When we compute the percentages, the data rate of the cost tables are also included. The use of the Variable Length Code results in a reduction of 29.75% in data rate with the *foreman* sequence. A similar rate decrease is shown in the *stefan* sequence. The *coastguard* sequence reduces the data rate by nearly half. The reason is that the movement of the objects are linear and the motion vectors throughout the frame are similar to each other. This results in high probability of some specific motion data and achieves higher compression ratio. More than half the data rate is reduced with the *silent* sequence. This is due to the fact that there is little motion in the *silent* sequence and zero motion vectors account for a large portion of the motion vectors. Coding with symmetrical RVLC costs marginally more in a higher data rate which provides higher error resilient. With the 16×16 block size and the symmetrical RVLC, the backward channel bandwidth is reduced to less than 2% of the forward channel bandwidth.

Sequences	VLC	RVLC
foreman	70.25%	74.59 %
coastguard	53.02 %	54.76 %
stefan	72.34 %	76.60 %
silent	47.41 %	49.60 %

Table 2. Comparison of the backward channel coding options

4.2. Hybrid Scheme with Selective Coding

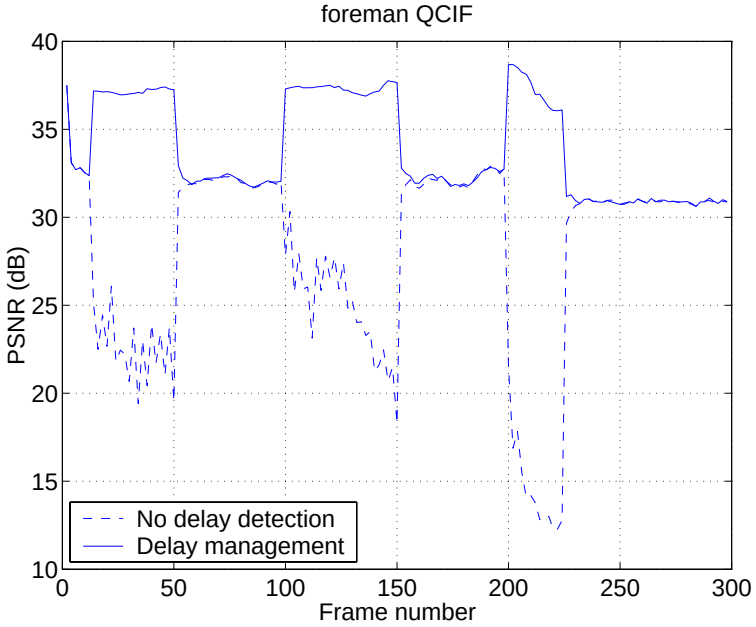


Figure 9. Comparison of performance with and without delay management (Only N frames and I frames)

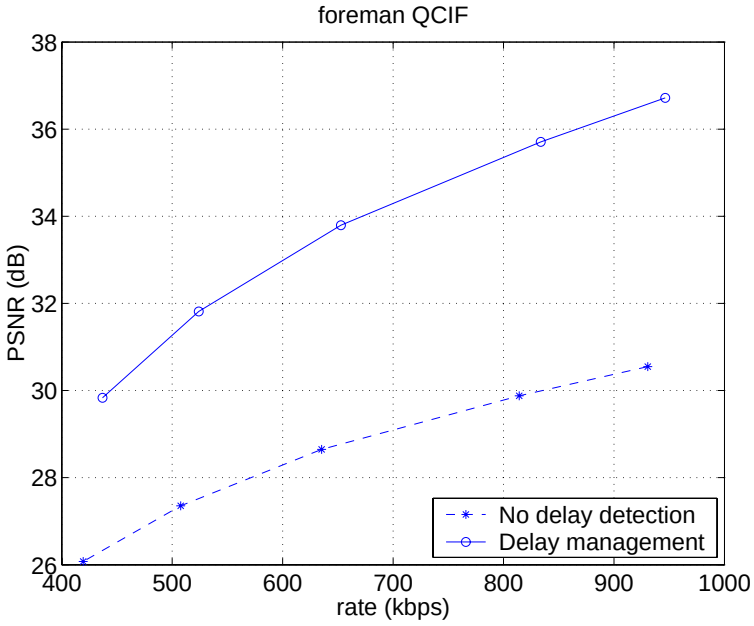


Figure 10. Rate distortion performance without and with delay management (*foreman* QCIF)

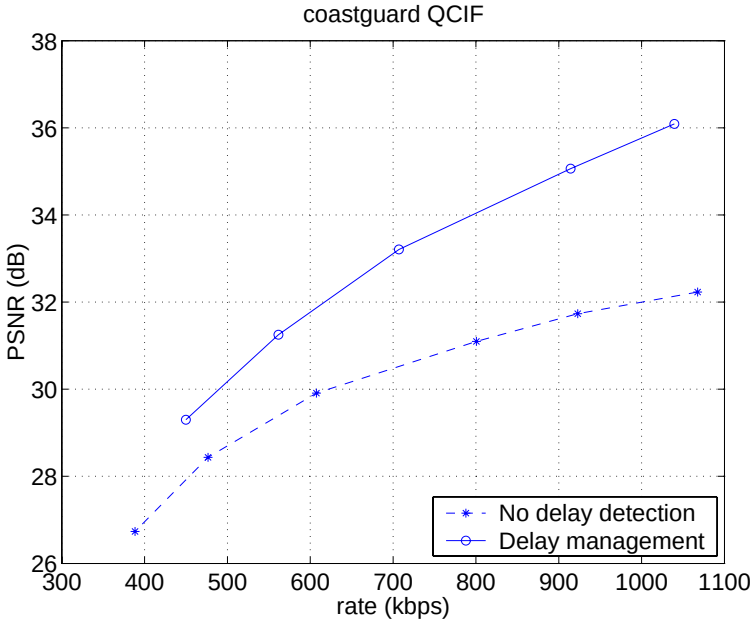


Figure 11. Rate distortion performance without and with delay management (*coastguard* QCIF)

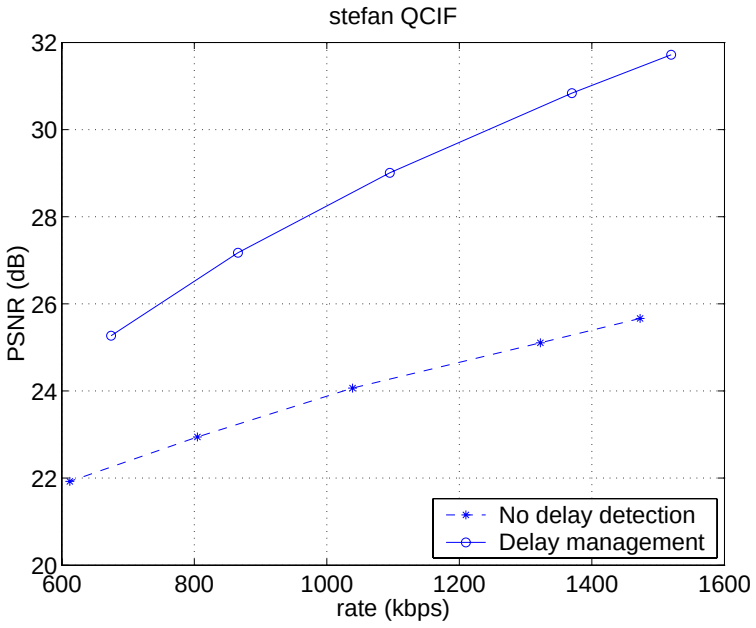


Figure 12. Rate distortion performance without and with delay management (*stefan* QCIF)

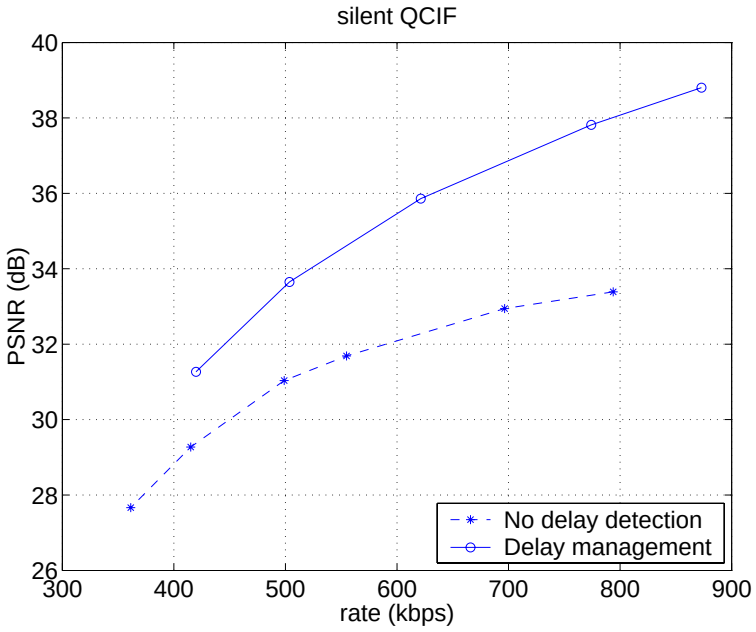


Figure 13. Rate distortion performance without and with delay management (*silent* QCIF)

In the following section, the delay distribution is shown in Figure 6 (a). Approximately 40% of the sequence experiences transmission delay, where some of the frames have two frames delay. The comparisons of the coding performance is shown in Figure 9 for the *foreman* sequence. The dashed line represents the results without delay detection and the solid line represents the result of the hybrid scheme. Only N frames and I frames are presented here. Since N (I) frames and WZ frames are coded alternatively, half of the 300 frames are displayed here. It is obvious that when the delay occurs, the hybrid scheme improves the quality significantly. Meanwhile, the selective coding scheme requires larger data rates since it switches to intra-coded mode. Hence, in the following we show the comparisons of the rate distortion performance.

Figure 10-13 shows the rate distortion comparisons of two sequences with and without Selective Coding. We control the data rates by varying the quantization parameter of the sequences. The solid lines represent the proposed scheme with the switch between intra frames and the N frames. The dashed lines represent the condition when no delay detection is implemented and the sequence is coded by the network-driven Wyner-Ziv scheme even with the existence of the delay. The proposed hybrid scheme with selective coding demonstrates $4-6$ dB gain for the *foreman* sequence. It is noted that the gain is more significant in high data rates. A similar coding gain is shown in the *stefan* sequence. The *coastguard* sequence shows a more moderate $2-4$ dB coding efficiency improvement. This is because in the *coastguard* sequence, consistent motion persist in the sequence and the delay of the motion data still provides an acceptable estimate of the motion information. The *silent* sequence demonstrates an average gain of $2-4$ dB since there is little movement throughout the sequence.

5. CONCLUSIONS

In this paper, we studied the use of error resilient methods in network-driven Wyner-Ziv video coding. We used a symmetrical Reversible Variable Length Code to reduce the bandwidth requirement of the backward channel. An error and delay detection method was proposed. A hybrid scheme with Selective Coding was proposed to improve the coding quality when transmission delay occurs. Experimental results show that the data rate of the backward channel is reduced to less than 2% of the forward channel. The hybrid scheme shows a gain up to 6dB in the coding efficiency for the test sequences.

In the future, we will incorporate a refined side estimator[13, 26] to improve the quality of the reconstructed Wyner-Ziv frames.

REFERENCES

1. "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC)," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, JVT-G050, 2003.
2. J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine,* **vol. 4, no. 1**, pp. 7– 28, first quarter 2004.
3. D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory,* **IT-19, no. 4**, pp. 471–480, July 1973.
4. A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory,* **IT-22, no. 1**, pp. 1–10, January 1976.
5. B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," in *Proceedings of the IEEE*, **vol. 93, no. 1**, pp. 71–83, January 2005.
6. A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proceedings of IEEE Data Compression Conference*, pp. 252–261, (Snowbird, UT), April 2002.
7. A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv coding for video: Applications to compression and error resilience," in *Proceedings of the IEEE Data Compression Conference (DCC)*, pp. 93–102, (Snowbird, UT), March 25-27 2003.
8. J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters,* **vol. 5, no. 10**, pp. 417–419, October 2001.

9. A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters,* **vol. 6, no. 10**, pp. 440 – 442, October 2002.
10. S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory,* **vol. 49, no. 3**, pp. 626 – 643, March 2003.
11. A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: Applications to error resilience," *IEEE Transactions on Multimedia,* **vol. 6, no. 2**, pp. 249 – 258, April 2004.
12. Y. Liu, *Layered Scalable and Low Complexity Video Encoding: New Approaches and Theoretic Analysis.* Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, August 2004.
13. Z. Li, *New methods for motion estimation with applications to low complexity video compression.* Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, August 2005.
14. L. Liu, Y. Liu, and E. J. Delp, "Network-driven Wyner-Ziv video coding using forward prediction," in *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, pp. 641–651, (San Jose, CA), Januaray 2005.
15. W. B. Rabiner and A. P. Chandrakasan, "Network-driven motion estimation for wireless video terminals," *IEEE Transactions on Circuits and Systems for Video Technology,* **vol. 7, no. 4**, pp. 644–653, August 1997.
16. 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, July 2000.
17. "Coding of audio-visual objects, Part-2 Visual, Amendment 4: Streaming video profile." ISO/IEC 14496-2/FPDAM4, July 2000.
18. R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine,* **vol. 36, no. 6**, pp. 112–119, June 1998.
19. Y. Wang and Q. Zhu, "Error control and concealment for video communication: a review," in *Proceedings of the IEEE*, **vol. 86, no. 5**, pp. 974–997, May 1998.
20. T. Stockhammer, M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology,* **vol. 13, no. 7**, pp. 657– 673, July 2003.
21. C. Berrou and A. Glavieux, "Near optimium error correcting coding and decoding: Turbo codes," *IEEE Transactions on Communications,* **vol. 44, no. 10**, pp. 1261 – 1271, October 1996.
22. T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology,* **vol. 13, no. 7**, pp. 560– 576, July 2003.
23. Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Transactions on Communications,* **vol. 43, no. 2/3/4**, pp. 158–162, February/March/April 1995.
24. C.-W. Tsai and J.-L. Wu, "On constructing the Huffman-Code-Based reversible variable-length codes," *IEEE Transactions on Communications,* **vol. 49, no. 3**, pp. 1506–1509, September 2001.
25. http://iphome.hhi.de/suehring/tml/.
26. Z. Li, L. Liu, and E. J. Delp, "Wyner-Ziv video coding: How far away is coding reality?." submitted to the IEEE Transactions on Circuits and Systems for Video Technology.